COMPUTING edge

- Artificial Intelligence
- Computer Vision
- Edge Computing
- Data Trust

APRIL 2024

www.computer.org

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

IEEE COMPUTER SOCIETY IEEE Printed with inks containing soy and/or vegetable oils

STAFF

Editor
Lucy Holden

Periodicals Portfolio Senior Managers
Carrie Clark and Kimberly Sperka

Director, Periodicals and Special Projects
Robin Baldwin

Production & Design Artist
Carmen Flores-Garvey

Periodicals Operations Project Specialist
Christine Shaughnessy

Senior Advertising Coordinator
Debbie Sims

Circulation: *ComputingEdge* (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

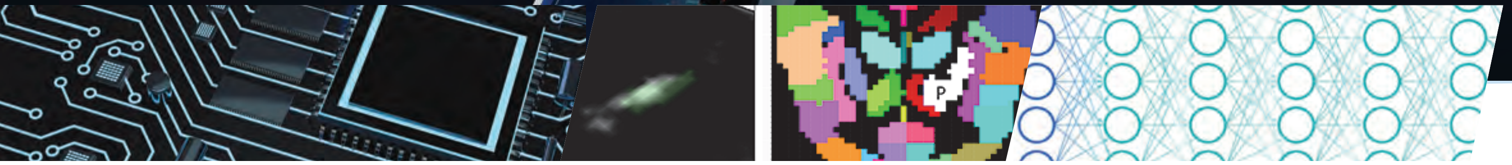
Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

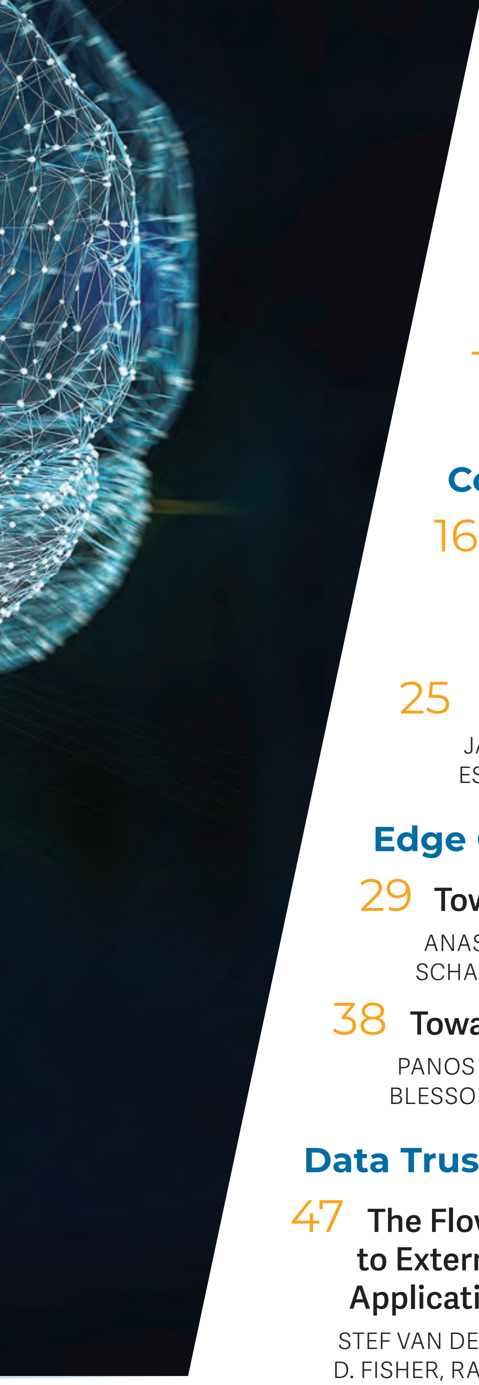
Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *ComputingEdge* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity,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

IEEE Computer Society Magazine Editors in Chief

APRIL 2024 · VOLUME 10 · NUMBER 4

COMPUTING edge





Artificial Intelligence

Computer Vision

Edge Computing

Data Trust

Departments

Magazine Roundup

The IEEE Computer Society's lineup of 12 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

Computer

Self-Encrypting Drive Evolving Toward Multitenant Cloud Computing

The authors of this February 2024 *Computer* article propose multitenant security architectures employing self-encrypting drives (SEDs) for improving security in cloud servers. They introduce emerging security technologies, such as link encryption, trusted virtualization, attestation, and fine-grained encryption, to address questions concerning the new threats and drawbacks when contemporary solutions are naively combined with an SED. They describe how next-generation SEDs can incorporate these technologies and discuss how the multitenant cloud system, properly built with these SEDs, can effectively address the new challenges.

Computing in Science & Engineering

Adopting Software Engineering Concepts in Scientific Research: Insights from Physicists and Mathematicians Turned Consultants

To investigate potential benefits of software engineering concepts (SECs) in scientific projects, a survey was conducted of former physics and mathematics researchers now working as consultants in the software engineering domain. In the survey, as reported in a July/August 2023 article in *Computing in Science and Engineering*, the participants reflected on the usefulness of various SECs for improving repeatability, reproducibility, and correctness of research results. This suggests that research in these fields could benefit from increasing usage of SECs, particularly agile development, continuous integration, and containerization.

IEEE Annals of the History of Computing

Educational Computers in New Zealand Schools: 1977 to 1983

New Zealand, though always a technologically advanced country, experienced some delays in getting computers into schools once available in 1977. In 1981, two New Zealand academics recognized an opportunity to improve access and produced two machines designed for education/schools: the Poly series of computers and the Aamber Pegasus. This October–December 2023 *IEEE Annals of the History of Computing* article examines this piece of New Zealand history and puts it in the context of other countries' approaches to computers in education during this era.

IEEE Computer Graphics and Applications

"Feels Like an Indie Game"—Evaluation of a Virtual Field Trip Prototype on Radioactive Waste Management Research for University Education

This article in the January/February 2024 issue of *IEEE Computer Graphics and Applications* describes the design and evaluation of a virtual field trip on the topic of radioactive waste management research for university education. The authors created an interactive virtual tour through the Mont Terri underground research laboratory by enhancing the virtual experiment information system, designed for domain experts, with background information, illustrations, tasks, tests, and an improved user interface.

IEEE Intelligent Systems

If Our Aim Is to Build Morality Into an Artificial Agent, How Might We Begin to Go About Doing So?

The authors of this November/December 2023 *IEEE Intelligent Systems* article highlight the different aspects that should be considered when building moral agents, including the most relevant moral paradigms and challenges. They also discuss the top-down and bottom-up approaches to design and the role of emotion and sentience in morality, and how governance and policy are becoming ever more critical in AI ethics and in ensuring that the tasks we set for moral agents are attainable and that ethical behavior is achieved.

IEEE Internet Computing

Rethinking Certification for Trustworthy Machine-Learning-Based Applications

Certification in machine learning (ML)-based applications is seen by policy makers, regulators, and industrial stakeholders as the preferred assurance technique to assess nonfunctional properties (e.g., fairness, robustness, and privacy) and improve trustworthiness. In this November/December 2023 article in *IEEE Internet Computing*, the authors analyze the challenges and deficiencies of current certification schemes, discuss open research issues, and propose a first certification scheme for ML-based applications.

IEEE micro

Privacy by Memory Design: Visions and Open Problems

In this January/February 2024 *IEEE Micro* article, the authors propose a first-of-its-kind design regime that realizes differential privacy (DP) in hardware memories. The salient feature of this novel design lies in its transformation of the notorious memory noises at sub-nominal voltages into the desired DP noises, thereby achieving power savings and privacy preservation simultaneously: a "win-win" outcome. They also demonstrate the feasibility of this design regime and outline a potential future research road map.

IEEE MultiMedia

eCubeLand: An Intelligent Multiview Video Data Modeling

The extensive use of surveillance systems, particularly those installed in Internet of Things environments, leads to the continuous harvesting of tremendous amounts of video data, which presents a challenge to process due to variability and unstructured storage. The authors of this October–December 2023 *IEEE MultiMedia* article propose an intelligent modeling framework, offering a convenient representation with indexing for real-world objects and solving complicated computer vision problems, such as anomaly detection and person re-identification.

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

Detecting Mobile Malware Associated With Global Pandemics

This article, in *IEEE Pervasive Computing*'s October–December 2023 issue, proposes the use of app permissions and an extra feature (the total number of permissions) to develop a static detector using machine learning (ML) models to enable the fast-detection of pandemics-related Android malware at installation time. Using a dataset of more than 2000 COVID-19 related apps and by evaluating ML models created using decision trees and Naive Bayes, they show that pandemics-related malware apps were detected with an accuracy above 90% using decision tree models with app permissions and the proposed feature.

IEEE SECURITY & PRIVACY

Augmenting Security and Privacy in the Virtual Realm: An Analysis of Extended Reality Devices

The authors of this *IEEE Security & Privacy* article, in the January/February 2024 issue, present a device-centric analysis of security and privacy attacks and defenses on extended reality (XR) devices. They also explore future research directions and propose design considerations to help ensure the security and privacy of XR devices.

IEEE Software

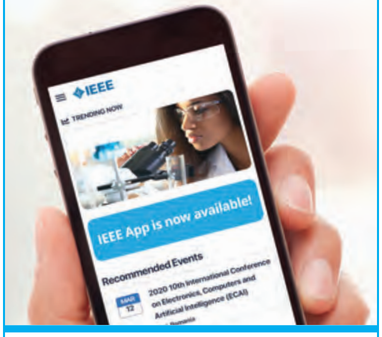
Focusing on What Matters: Explaining Quality Tradeoffs in Software-Intensive Systems Via Dimensionality Reduction

Building and operating software-intensive systems involves exploring decision spaces composed of large numbers of variables and their complex relations. The authors of this January/February 2024 *IEEE Software* article report on using dimensionality reduction techniques that enable decision makers in different domains to focus on crucial elements of the decision space.

IT Professional

An Overall First Responder Tracking and Coordination Framework

For first responders (FRs), self-localization at the scene can be especially difficult in indoor scenarios where signals and navigation systems may be unavailable for reliable positioning. In this November/December 2023 *IT Professional* article, the authors propose a system combining self-localization, communication of the FRs' locations, 3-D building reconstruction or floor plans (if available), and visualization to improve indoor positioning, georeferencing the positions, and finally, visualizing the results in a suitable visualization tool.


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

Editor's Note

The Incredible Potential and Risks of Machine Learning and Artificial Intelligence

The potential uses of artificial intelligence (AI) and machine learning (ML) seem endless. But as progress speeds ahead, developers may overlook the risks of relying too heavily on ML and AI without proper testing and development. This issue of *ComputingEdge* highlights some of the ways ML and AI can aid society, such as through medical discoveries, architectural advancement, edge computing, and data analysis. The articles also grapple with the limitations of depending on ML and AI, particularly when it comes to trust.

To utilize the potential of using AI to advance architecture and other fields, people must be able to trust it. The authors of "Harnessing Artificial Intelligence to Design Healthy, Sustainable, and Equitable Places," from *Computer*, explore how architects can use ML and AI to design buildings and neighborhoods. *IEEE Security & Privacy*'s article "Trustworthy AI Means Public AI [Last Word]," argues that businesses must prioritize security and transparency over profit when developing AI in order to ensure its trustworthiness.

Computer vision applications create exciting opportunities in the human brain and the workplace. In "Visualizing Multimodal Deep Learning for Lesion Prediction," from *IEEE Computer Graphics and Applications*, the authors discuss the potential for using image segmentation to predict lesions caused by strokes. *IT Professional*'s "Implementing Behavioral Biometrics With TRUST," explains why companies should implement behavioral biometrics using the TRUST approach as a way to maintain transparency and respect between workers and management.

IT engineers worldwide are brainstorming strategies for how to make edge computing more efficient and effective. "Toward Building Edge Learning Pipelines," from *IEEE Internet Computing*, envisions a future where data acquisition, advanced ML, and analytics workflows blend together to develop edge learning solutions. In *IEEE Internet Computing*'s "Toward Sustainable Serverless Computing," the authors make the case for using ML to enable more energy efficient serverless computing.

To use ML effectively, organizations must be able to trust it. The authors of "The Flow of Trust: A Visualization Framework to Externalize, Explore, and Explain Trust in ML Applications," from *IEEE Computer Graphics and Applications*, present a framework for helping users build trust in ML through interactive visualizations. In *IEEE Intelligent Systems*' "The Secrets of Data Science Deployments," the authors identify obstacles that are getting in the way of using ML to analyze large data sets and propose possible solutions.

EDITOR: Joseph Williams, Pacific Northwest National Laboratory, joseph.williams@pnnl.gov

This article originally appeared in Computer vol. 55, no. 2, 2022

DEPARTMENT: COMPUTING ARCHITECTURES

Harnessing Artificial Intelligence to Design Healthy, Sustainable, and Equitable Places

Phillip Bernstein, *Yale School of Architecture*
Mark Greaves, *Pacific Northwest National Laboratory*
Steve McConnell, *NBBJ*
Clifford Pearson, *Architectural Record*

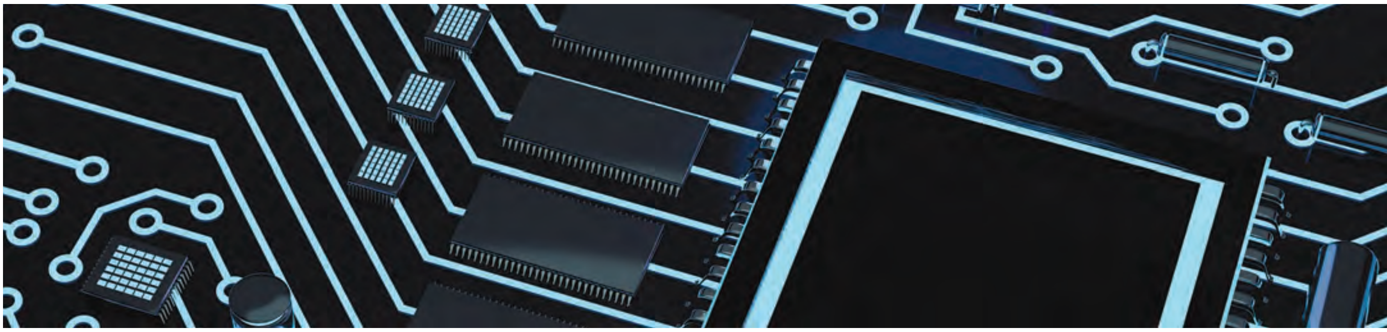
Modern machine learning and artificial intelligence (AI) have revolutionized many disciplines but have only minimally impacted the practice of architecture. We discuss design challenges that architects face, illustrate how AI can meet them, and describe three areas where progress is needed to ignite the AI revolution in architecture.

Our term for the designers of computing systems, *architects*, is derived from the far older profession of designing the structures and places that occupy our world. Today's architectural firms are avid consumers of computing technology, from powerful CAD systems and building models to sophisticated urban simulations. However, while modern machine learning and artificial intelligence (AI) techniques have revolutionized many disciplines, they have only minimally impacted the practice of architecture.

Architects of the built environment need to do more than ever. It is no longer sufficient for buildings and places to meet Vitruvius' rubric: *firmitas, utilitas, et venustas*—strength, utility, and beauty. Now architects must also respond to a broad range of environmental, social, and community concerns. Designing even a single-family home today requires attention to climate change, pollution, the carbon footprint of every material used in construction, fair labor practices throughout the building supply chain, affordability, racial equity, and the development of healthy communities—in addition to all of the usual demands of the client and regulatory agencies. An explosion of information and data on all of these issues now influences every step in the design process and simultaneously threatens to overwhelm the people running that process. How can the architects of our built environment meet the demands of this new class of design goals without losing sight of their less-quantifiable aspiration to create inspiring and captivating buildings?

Designing great buildings has always been a "wicked" problem—one defined by imprecise goals, incomplete knowledge, deeply interconnected subproblems, and the need to continuously make best-guess tradeoffs (https://en.wikipedia.org/wiki/Wicked_problem). Instead of right or wrong answers, wicked problems require us to think in terms of better or worse solutions and rely on professional judgment and experience to point us forward. In a recent op-ed (https://www.archpaper.com/2021/02/op-ed-tackling-bidens-climate-change-challenge-artificial-and-human-intelligence/), we discussed a wicked problem tucked within President Bidens year-one legislative agenda on climate change (https://joebiden.com/climate-plan/)—a call to create the innovative technologies needed to build "zero net energy buildings at zero net cost." Owing to the cross-disciplinary nature of the problem and its resistance to traditional design methods, we argued that architects could leverage AI as a tool to

Digital Object Identifier 10.1109/MC.2021.3134422
Date of current version: 14 February 2022



supplement human intelligence and help find novel solutions.

Today's architects already work routinely with digital tools to design, manage, and construct projects. At the center of this work is a technology known as *building information modeling* (*BIM*), which enables an architect to create a detailed, 3D representation of her design that behaves, digitally, like a real building. That model can then be used to predict the energy performance, daylight usage, and even the cost of the project before it is built as well as generate drawings and images to help explain the project to the client. With BIM as a base, architects can create vast amounts of digital information about their projects with tools that simulate a building and predict how it might operate in reality—everything from the amount of carbon needed to power its lighting system to the number of people who might occupy a particular space at a particular time. The builder can use that same model to price the project, order materials, coordinate labor in the field, and sequence construction. The BIM simulation lives on after construction as a digital tool that can be used by the building owner to operate the property and monitor its performance. BIM has established a common platform for architects, engineers, consultants, and contractors to develop and analyze building projects with robust analyses driven by explicit 3D modeling, well-understood relationships, and conventional types of digital simulation.

Contemporary AI approaches are fundamentally different from traditional BIM-based analyses. While BIM allows designers to model individual structures in significant detail, the machine learning algorithms used by today's AI systems identify patterns and correlations that are implicit in massive pools of data and then use those patterns to make predictions about specific instances. Crucially, the patterns that AI systems discover from large sets of design data can far exceed the power of the handcrafted building data models and simulations that are encompassed by BIM. These patterns capture subtle but authentic regularities across thousands of individual features and millions of examples, surpassing what humans can encode or even effectively explain using traditional models and rules.

Modern facial recognition systems illustrate this well. Rather than having programmers explicitly identify and model all the graphical elements relevant to recognizing faces and then write computer code that recognizes specific faces as unique combinations of these elements, AI systems process millions of images to learn distinguishing facial patterns for themselves. Continuous feedback and retraining allow the AI system to perform with increasing accuracy. Yet, the way that specific machine-learned parameters encode facial features to allow identification of any one face remains a mystery that, so far, is extremely challenging to explain in intuitive terms. This AI process is reminiscent of how humans learn to identify faces. No one explicitly teaches us how to recognize our friends. Our brains independently learn to do this task automatically and reliably based on experience gathered at a very young age, and yet the details of our own ability to recognize faces are a mystery. As with AI, we cannot satisfactorily explain how we do it, even to ourselves.

> *THE BIM SIMULATION LIVES ON AFTER CONSTRUCTION AS A DIGITAL TOOL THAT CAN BE USED BY THE BUILDING OWNER TO OPERATE THE PROPERTY AND MONITOR ITS PERFORMANCE.*

The basic property that characterizes modern AI is that machine learning algorithms are able to extract subtle patterns and correlations from large data sets without requiring an expert to explicitly model all of the detailed relations among the elements of such data sets. For example, AI might help uncover a deep connection between certain design choices and patient health outcomes to create more effective hospitals or exploit complex patterns between space and light to identify designs that would best support a socially vibrant public square.

While modern AI has gained little traction in the building industry to date, it is on track to provide a powerful path to extend and complement the detailed modeling capabilities of BIM and support architects in understanding tradeoffs that are not apparent from traditional digital-model representations of a design. A variety of innovative digital tools available in recent years has amplified architects' capabilities to create, depict, analyze, and communicate their design ideas (https://www.architecturalrecord.com/articles/15409-continuing-education-artificial-intelligence). Many such tools extend the data and power of BIM models, but they are targeted at single designs rather than leveraging a broad array of data compiled from many projects. AI improves on this by identifying patterns implicit in hundreds or even thousands of individual designs to help architects effectively respond to subtle and difficult-to-model design concer–ns. Just as AI has revolutionized how we design our drugs and drive our cars, it can uniquely help architects create inspiring and beautiful buildings that also respond to our desire for equity, justice, health, and community—issues that are at the forefront of the architectural profession today.

> *WHILE AI MAY BE A DISRUPTIVE TECHNOLOGY, IT CANNOT FULLY AUTOMATE THE WORK OF DESIGN, ANY MORE THAN BIM "SOLVED" ARCHITECTURE.*

APPLYING AI TO THE DESIGN OF PLACES

AI can make it possible for architects to resolve vastly more complex design agendas than they are able to today. It will do this by rapidly analyzing huge amounts of data to generate options for design teams to consider and refine, just as Spotify or Netflix do when they recommend music or movies we might like. In architecture, AI can accelerate the design process to find and evaluate options that are likely to satisfy a complex set of design requirements across a variety of programmatic and qualitative domains. Finally, AI can integrate with advanced simulation technology to help architects assess the effectiveness of various design solutions aimed at satisfying the diverse demands of a building project.

Of course, solving complex—even wicked—problems such as "zero net energy buildings at zero net cost" through design involves evaluating the human costs of tradeoffs and judgments, tasks that humans still do far better than algorithms. So while AI may be a disruptive technology, it cannot fully automate the work of design, any more than BIM "solved" architecture. Used properly, AI can give architects mastery of complex agendas by revealing insights based on otherwise impossible-to-recognize patterns and solutions, while preserving the ability of architects to focus on understanding human needs, applying creativity, and developing artistry.

In the near future, architects may employ AI to help with a broad range of challenges. To offer a clearer idea of what this might mean, following are three scenarios involving different project scales and levels of complexity.

BETTER EDUCATION

The first example is an elementary school in a suburban community designed to exemplify best practices in education, sustainability, and racial and economic diversity. In November 2020, the *Public School Review*, a widely used platform that provides free, detailed profiles of American public schools and their surrounding communities, highlighted 10 major challenges facing these institutions—including addressing social and health issues. But this information offers architects and educators only limited help in identifying the cause-and-effect relationships between critical elements in school design and desired outcomes, such as reduced absenteeism, higher test scores, and more parental engagement.

In this example, AI can work as an architect's assistant by leveraging large databases of BIM elementary school models along with regional health and testing data to surface new and promising configurations that best correlate a range of holistic, qualitative objectives.

By broadening the scope and increasing the depth of the architect's understanding of the design problem, such assistance can help create schools that work better for students, faculty, and staff.

BETTER HEALTH OUTCOMES

A second scenario examines a larger, more complex project: a regional hospital with inpatient and outpatient care, a full range of departments (obstetrics, oncology, emergency, cardiac, renal, pulmonary, and so on), and both research and clinical facilities. Not only is this project larger and more expensive than the school, but it must resolve the often-competing needs of its different user groups: patients, visitors, medical professionals, and support staff.

New AI tools can review continually evolving data on patient-stay durations, medical procedure recovery times, and hospital-borne infections to find critical patterns and then apply this knowledge to help architects design the hospital. By revealing the underlying connections between design and outcomes—such as the relationship of building configuration to reduced hospital stays, better utilization of expensive medical equipment, and reduced carbon footprint—AI can establish an evidence-based process for architecture. It can also provide validation for design decisions, allowing architects to explore strategies that might not seem promising at first glance. And AI systems can be continually updated with new data sets and the latest studies.

BETTER COMMUNITIES

The third example involves the master plan for an urban neighborhood that proposes a holistic strategy for commercial mixed-use buildings, residential construction, infrastructure improvements, and public open space. By encompassing not only individual buildings, but also the interactions among them within the larger context of streets, parks, and occupants, the plan must accommodate a degree of complexity that challenges any existing technology. It requires coordination at multiple scales: from the building materials used to solar orientation, from landscaping to transit facilities, from safe bicycle routes to issues of affordability and diversity. By finding correlations across the relevant data, AI can help architects and planners evaluate complex decisions, make myriad tradeoffs, and project how individual pieces will fit together to support a vibrant community.

These data will be provided by today's smart city technologies, which collect vast amounts of information on traffic, pollution, open space usage, crime, energy consumption, and all kinds of other urban functions. If all of this information could be brought together, it could be used to train algorithms to discover patterns and offer insights on complex environmental, social equity, population health, and community governance issues that can't be found using traditional planning methodology. Those same AI tools could periodically update the master plan with new insights as the project moves forward over time, keeping it relevant as the community develops and plans evolve. As the AI systems learn more from these large data sets, they can begin to simulate and predict to help planners and architects shape a healthier and more sustainable future.

Each of these three imagined scenarios offers a window into the realm of possibilities that can leverage AI to correlate complex agendas that span a range of social, health, and environmental issues central to human progress and a healthy planet.

AI can also help architects expand the scope and value of services they provide to clients. Armed with the latest software, firms can engage with clients earlier in the process—providing advice, for example, on real estate decisions and programming. At the other end of the process, AI can help architects use data gathered on a completed building's performance in terms of energy use, water conservation, indoor air quality, and user comfort to manage a property for a client, a service not normally assumed by architects, but one that could be attractive to some.

Ultimately, AI is a tool of empowerment, giving architects the space to do what they do best: develop innovative ideas and new solutions. AI will allow them to focus on the poetics of a project, not just the pragmatics. AI has tremendous potential to advance the practice of design to more reliably create places and buildings that respond to national priorities for equity, justice, health, and community—and leverage the built environment to bolster our values across all strata of society. There are three areas where progress is needed to bring about this future:

1. *The promise of AI to improve architecture depends on the ability of these algorithms to learn from massive assemblies of information about design, construction, and building operation*. Data are the fuel for AI and analysis. Compiling these data, though, is beyond the capability of any single firm or group of firms in the design industry, or even leading professional associations like the American Institute of Architects. There are complex and difficult issues surrounding creation of such a data resource, including ownership, access, privacy, data bias, social equity, data assurance, labeling, governance standards, and protection. We envision a built environment data trust, overseen by a distinguished oversight board, which would aggregate as much building design data as possible. Such a resource will ignite the AI revolution in architecture.
2. *AI can allow designers to create far better performing buildings while reducing their environmental and energy footprints, improving the health of people who use and live near them, and leveraging the built environment to address a broad range of social and economic issues*. These impacts and the processes needed to achieve them will first be explored in architecture schools and leading architecture firms. We envision a set of innovative pilot programs aimed at using AI and information in the built environment data trust to drive AI forward for architectural design. This effort should include public–private partnerships with leading architecture firms, universities, construction companies, real estate developers, and building owners to ensure that technical advances can be quickly applied to real-world projects. Working together, such partnerships can enable the multitrillion-dollar design and construction market to take transformative steps to operate with greater efficiency and create communities that are healthier and more environmentally sustainable.
3. *Each of us recognizes the excitement that can be generated when like-minded people come together.* We look forward to an increasing number of workshops and conferences addressing AI's impact on the built environment. They will involve researchers, technologists, practitioners, developers, financiers, and government officials, all seeking to identify innovative ways of using data and AI to design and build sustainable, resilient, and healthy places for the 21st century.

A powerful tool tends to change its user. AI is a revolutionary technology that can transform the American practice of architecture in deep and positive ways and shape a healthy and sustainable future for the world we live in.

DISCLAIMER

The views expressed in this article are those of the authors and do not necessarily reflect the positions of their employers.

PHILLIP BERNSTEIN is an associate dean and a professor adjunct at the Yale School of Architecture, New Haven, Connecticut, 06510, USA. Contact him at phillip.bernstein@yale.edu.

MARK GREAVES is a senior researcher in artificial intelligence at Pacific Northwest National Laboratory, Seattle, Washington, 98109, USA. Contact him at mark.greaves@pnnl.gov.

STEVE MCCONNELL is an architect and managing partner at the global design firm NBBJ, Seattle, Washington, 98109, USA. Contact him at smcconnell@nbbj.com.

CLIFFORD PEARSON is a contributing editor at *Architectural Record*, New York, New York, 10014, USA, and writes about architecture and urbanism. Contact him at cliff.pearson@gmail.com.

IEEE COMPUTER SOCIETY

www.computer.org

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

OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.

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

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

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

PUBLICATIONS AND ACTIVITIES

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

Periodicals: The Society publishes 12 magazines, 19 journals.

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

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

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

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

Certifications: The society offers three software developer credentials.

COMPUTER SOCIETY OFFICES

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

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

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

IEEE

revised 22 February 2024

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 21, no. 6, 2023

Trustworthy AI Means Public AI

Bruce Schneier, *Harvard University*

Back in 1998, Sergey Brin and Larry Page introduced the Google search engine in an academic paper that questioned the ad-based business model of the time. They wrote: "We believe the issue of advertising causes enough mixed incentives that it is crucial to have a competitive search engine that is transparent and in the academic realm." Although they didn't use the word, their argument was that a search engine that could be paid to return particular URLs is fundamentally less trustworthy. "Advertising income often provides an incentive to provide poor quality search results."

We all know what happened next. Google eventually sold ads: first in dedicated boxes that made it obvious that the links were paid for, and then slowly integrated into the actual search results until paid links were seamlessly integrated into the "real" search results.

It's a story that's been repeated many times. Your Facebook and Instagram feeds are filled with "sponsored posts." An Amazon search returns pages of products whose sellers paid for placement. Buried behind an obscure link, the travel search engine Kayak discloses that "some results have been promoted based on their revenue potential for us."

This is how the Internet works. Companies spy on us as we use their services. Data brokers buy that information from smaller companies, and assemble detailed dossiers on us. They then sell that information back to other companies, who combine it with data they collect in order to manipulate our behavior to serve their interests—at the expense of our own.

And even if they don't manipulate us directly—even if we pay for the products and services—they spy on us. Our televisions spy on us. Our smartphones spy on us, as do our appliances, our cars, and everything else with a plug or battery. Surveillance is the business model of the Internet. And the use of manipulative interfaces is so prevalent that it has its own name: dark patterns.

We use all of these services as if they were our agents. In fact, they are double agents, secretly working for their corporate owners. We trust them, but they are not trustworthy.

We should not expect the current crop of generative AI systems to be any different. And the results will be much worse, for two reasons.

The first is that these AIs will be more relational. We will be conversing with these systems, using natural language. As such, we will naturally ascribe human-like characteristics to them. And we will treat them as trusted assistants and intimate friends.

This relational nature makes it easier for double agents to do their work. Did your chatbot recommend a particular airline or hotel because it's truly the best deal, given your particular set of needs, or because the AI company got a kickback from those providers? When you asked it to explain a political issue, did it bias that explanation towards the company's position? Or in a way that benefits whichever political party gave it the most money?

The second reason to be concerned is that these AIs will be more intimate. One of the promises of generative AI is a personal digital assistant: acting as your advocate with others, and as an intimate butler with you. This requires an intimacy greater than your search engine, email provider, cloud storage system, or smartphone. You're going to want it with you 24 × 7, constantly recording and training on everything you do, so it can most effectively work in your best interest.

And it will help you in many ways. It will notice your moods and know what to suggest. It will anticipate your needs and work to satisfy them. It will be your therapist and life coach.

You will want to trust it. Its interface will make it hard not to trust, because we have evolved to judge

Digital Object Identifier 10.1109/MSEC.2023.3301262
Date of current version: 10 November 2023

humans—and quickly ascribe human agency when we see human-like behaviors. Previous computer interfaces were limited, so it was easy to remember that a search window—or a thermostat's interface—wasn't a person. AI agents will interact with the totality of our existence in ways another person would, easily breaking our prior systems of judgment. And so, again, it will need to be trustworthy.

Today's generative AI systems are not trustworthy. We don't know how they are trained. We don't know their secret instructions. We don't know their biases, either accidental or deliberate. All we know is that they are created, at great expense, by corporations that will use every trick they can think of to make them as profitable as possible.

We are going to need to build these generative AI systems and assistive agents with security in mind, from the ground up. This means security from outside attack, of course, but also from the changing business models of the corporations themselves.

Everyone will need their own secure personal data storage. Processing will require secure enclaves. Communications will need to be encrypted. All of this will need to be decoupled, so that no single provider can compromise security. This isn't hard—it's the sort of stuff security engineers do all the time—but we will continuously need to fight against both corporate and government desires for surveillance. This means no backdoors.

Transparency is essential, but only part of the solution. It's possible to poison a model in a way that even someone looking at the training data won't know. And, more generally, the market incentive for the large tech companies to violate our privacy and influence our behavior is just too great. We are going to need to build open-source public models, not licensed from or otherwise controlled by the large tech companies. The development of this transformative technology must not be steered solely by the private sector's near-term financial interests. Generative AI is just too important, too transformative, too relational, and too intimate. We won't truly trust AI unless it's trustworthy, and that requires both secure technology and secure policies.

EDITOR: Mike Potel, potel@wildcrest.com

DEPARTMENT: APPLICATIONS

This article originally appeared in IEEE Computer Graphics and Applications vol. 41, no. 5, 2021

Visualizing Multimodal Deep Learning for Lesion Prediction

Christina Gillmann and Lucas Peter, *Leipzig University, 04109, Leipzig, Germany*

Carlo Schmidt, *Empolis Information Management GmbH, 67657, Kaiserslautern, Germany*

Dorothee Saur and Gerik Scheuermann, *Leipzig University, 04109, Leipzig, Germany*

A U-Net is a type of convolutional neural network that has been shown to output impressive results in medical imaging segmentation tasks. Still, neural networks in general form a black box that is hard to interpret, especially by noncomputer scientists. This work provides a visual system that allows users to examine U-Nets that were trained to predict brain lesions caused by stroke using multimodal imaging. We provide several visualization views that allow users to load trained U-Nets, run them on different patient data, and examine the results while visually following the computation of the U-Net. With these visualizations, we can provide useful information for our medical collaborators showing how the training database can be improved and which features are best learned by the neural network.

Lesions define areas in organs or tissue damaged through injury or disease. Brain lesions can lead to movement, attention, speech, and language disorders.[1] In the case of stroke, a vessel in the brain becomes occluded and fails to provide the surrounding tissue with an appropriate amount of blood. Strokes are the second leading cause of disability worldwide[2] and a precise and reliable prediction of a lesion, including its location and shape, can help clinicians in making a diagnosis and determining the proper treatment.

Neural networks have become increasingly popular in the medical domain as they have proven to be quite powerful, especially in image segmentation tasks[3] often used for the prediction of stroke lesions.

Lesion prediction can be performed in various ways, ranging from a basic quantification of the outcome (good or bad) to a holistic prediction of tissue damage in the brain. In these disciplines, neural networks have been shown to achieve impressive prediction results.

However, to the end user, neural networks form a black box, meaning that the prediction process cannot be reviewed directly. This is an important issue in medical applications,[4] as clinicians make decisions that can have a massive impact on patients' health. Medical researchers tend to discard novel computational approaches if they do not provide a mechanism that allows them to review and understand how the method works.[5]

Tjoa and Guan[6] highlighted this issue and proposed that a visual interpretation of neural network approaches is a crucial requirement for using these algorithms in applications and reaching the goal of explainable artificial intelligence (XAI) in medicine.[7] Singh *et al.*[8] provided a survey on available visualization approaches in the area of XAI and highlighted that the selection of a proper algorithm is highly dependent on the underlying task.

Although a variety of XAI approaches have been published in recent years, applying XAI to the prediction of brain lesions raises a number of novel questions. Our lesion prediction technique uses multimodal input, meaning that each patient is captured by multiple types of medical images. A neural network is

0272-1716 © 2021 IEEE
Digital Object Identifier 10.1109/MCG.2021.3099881
Date of current version 8 September 2021.

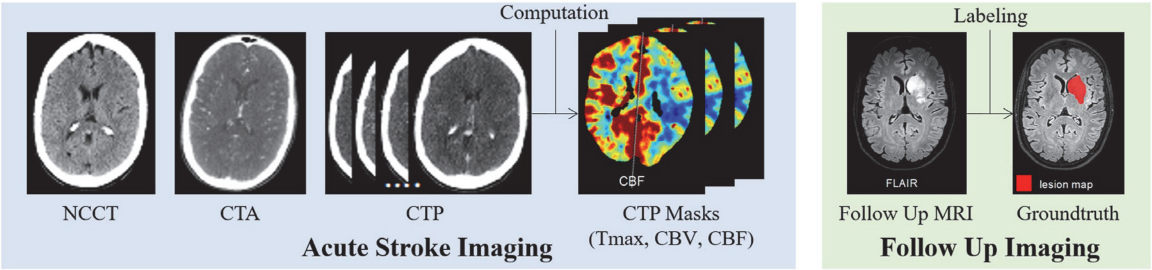


FIGURE 1. Endovascular Stroke Database: for each patient, multiple scans are acquired that can be separated into acute stroke imaging (blue) and follow-up imaging (green). Acute stroke imaging consists of NCCT, CTA, and CTP, where CTP is used to generate CTP Masks (Tmax, CBV, CBF). Follow-up imaging consists of MRI and a semi-automatically created ground truth.

then trained that can output a probabilistic lesion map of a patient's brain. Although there have been visualization approaches that assist in reviewing multimodal medical images,[9] including ones applied to multimodal brain lesion visualization,[10,11] suitable visualization approaches in the context of XAI need to be found.

In this work, we demonstrate an XAI approach that helps clinicians understand and interpret a specific neural network trained to predict brain lesions.

MEDICAL IMAGING MODALITIES

Endovascular Stroke Database

In this project, we worked with datasets acquired at the Department of Neuroradiology, University of Leipzig Medical Center. A standardized pipeline was implemented to create datasets for 117 patients suffering from acute ischemic stroke due to large vessel occlusion. Each patient record contains multiple computed tomography (CT) scans and a magnetic resonance imaging (MRI) scan. These scans can be separated into two groups: *Acute Stroke Imaging* and *Follow-up Imaging*, as shown in Figure 1.

Acute Stroke Imaging

When a patient enters the hospital and all symptoms match a potential stroke, several types of CT scans are acquired using a Brilliance 64-slice or Ingenuity 128-slice CT scanner (Philips Healthcare, Best, The Netherlands) in clinical daily routine. Figure 1 shows the following three types of acute stroke imaging.

Noncontrast CT (NCCT) is a conventional CT scan of a patient's brain, showing the brain of the patient surrounded by the skull. White matter in a brain can be affected by lesions. The slice number depends on the patient's anatomy and slice thickness is between 0.8 and 5 mm. NCCT gives a first overview of the lesion.

CT Angiography (CTA) is the medical term for the radiological imaging of blood vessels through diagnostic imaging procedures. After the application of a contrast medium, one volume from the aortic arch to the vertex of the skull was acquired. CT Angiography allows the identification of blood vessels as can be seen in the lighter color in Figure 1 CTA. As lesions are a result of blocked vessels, this is important information for clinicians.

CT Perfusion (CTP) is a functional radiological examination method used to quantitatively determine cerebral perfusion. During the intravascular injection of a contrast medium, several images of the brain are generated in rapid succession, typically 16 time-steps per volume, capturing the propagation of the contrast medium over time. CT perfusion can indicate areas in the brain that are not properly supplied with oxygen.

Parameter Map(s) (CTP Masks) are used to reduce the complexity of CTP datasets, which can be hard to review as each is a video of volumes. We generate three types of CTP masks summarizing the measured CTP: Cerebral blood volume (CBV) is the volume of blood present at a given moment within the brain, Cerebral Blood Flow (CBF) depicts the flow of blood in the brain measured over time, and *Time to Peak* (Tmax) map shows the moment at which the contrast medium reaches its highest concentration. The calculations are made with software developed by VEOcore (veobrain.com).

Follow-Up Imaging

Although acute stroke imaging contains valuable information about which parts of the brain are currently badly supplied, it does not guarantee that a lesion will be found. Lesions can dissipate as parts of the tissue recover, but can also enlarge in border areas where partially supplied tissue is not able to recover properly. The final outcome is measured some

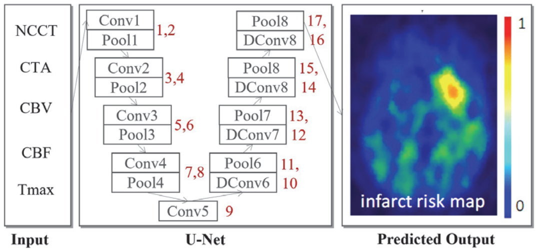


FIGURE 2. Training procedure of U-Net for multimodal stroke imaging. Acute Stroke imaging is used as an input to train a U-Net that outputs a lesion risk map. Layers are enumerated for reference.

days after the stroke, requiring a further imaging procedure.

Follow-Up MRI scans are acquired to definitively diagnose the final lesions.

Lesion Map (LM): Each lesion map is created by a neurologist utilizing a semiautomated segmentation algorithm clusterize tool (github.com/carderne/clusterize) operating on the follow-up MRI imaging containing the final lesion of the patient.

All modalities as well as the ground truth images are resliced and normalized to a standard brain.

LESION PREDICTION USING A U-NET

Using our endovascular stroke database, we can train a neural network to predict lesions. This training process is summarized in Figure 2. The input is formed by an arbitrary subset of acute stroke imaging (excluding CTP). We fuse all input images into one large stack of individual images.

This input is fed into a U-Net, which is a special variant of a convolutional neural network introduced by Ronneberger *et al.*,[12] who showed that U-Nets are well suited to medical imaging segmentation tasks. U-Nets can be separated into two parts. Part one is a composition of multiple convolutional layers (Con) and pooling layers (Pool), which iterate down to a bottleneck at the narrowest part of the neural network. Part two of the U-Net then expands back out using multiple deconvolutional (DCon) and pooling layers into a result that resembles the original image with the addition of a probabilistic lesion risk map, see Figure 2.

It is important to note that we are not simply performing a segmentation task with the utilized U-Net but also aim to predict lesions. The follow-up imaging that is used to generate the ground truth of the lesion is not fed into the U-Net as an input during training.

TABLE 1. Visually evaluated models of U-Net. Each model contains an ID, the included modalities, and the resulting AUC value.

ID	Modalities used	Learning Rate	Data augmentation	Loss function	AUC
0	NCCT CTA CBF CBV Tmax	0.01	rot. / def.	MSE	0.9475
1	NCCT CTA	0.01	rot. / def.	MSE	0.9468
2	NCCT CTA CBF CBV Tmax	0.01	rot. / def.	MSE	0.9463
3	NCCT CTA	0.0001	none	BCE	0.9389
4	NCCT CTA	0.001	none	MSE	0.9344
5	NCCT CTA	0.001	none	MSE	0.9344
6	NCCT CTA	0.01	rot. / def.	MSE	0.9339
7	NCCT CTA	0.0001	none	BCE	0.9309
8	NCCT CTA	0.01	rot. / def.	MSE	0.9268
9	Tmax	0.01	rot. / def.	MSE	0.9039

We conducted our initial experiments using leaky rectified linear units (ReLU) as activation layers, batch norm layers[13] to stabilize the output of activation layers, max pooling layers, stochastic gradient descent as the optimizer, and mean squared error as the error function. We ran the experiments (Table 1) with 1000 epochs and *learning rates* of 0.1, 0.01, 0.001, and 0.0001, where 0.1 and 0.01 learning rates provided the most promising results. This remained true when switching the *loss function* from mean squared error (MSE) to binary cross entropy loss (BCE).

Two kinds of *data augmentation* were also tested: *rotating* the input by 90° in a random direction and elastic *deformation*[14] of the input. Random changes on the input data act as an enlargement of the training dataset for the model, resulting in more accurate

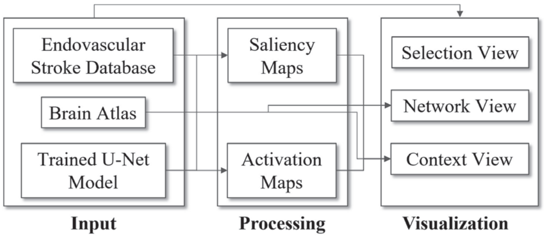


FIGURE 3. Overview of our visualization approach for understanding the learning process of U-Nets used to predict brain lesions.

predictions. However, combining both methods of data augmentation led to slightly worse results, suggesting that more training epochs are still needed to adapt to the virtually enlarged dataset.

In all, we tested 168 configurations of the U-Net in combination with different learning rates and examined more closely ten different models (Table 1) with the highest area under their ROC-curve. The receiver operating characteristics (ROC) curve maps true-positive-rate to false-positive-rate when comparing the ground truth output with the output of the model,[15] see for example Figure 5 upper right. Higher area under this curve (AUC) represents higher sensitivity, a major goal in the training process. AUC-ROC is a performance measure for classification problems as shown in this case.

DEEP LEARNING VISUALIZATION SYSTEM

Our neural network visualization system was implemented using the Python library Dash (dash.plotly.com). An overview of our system can be found in Figure 3.

Input

The input to our system is a composition of the Endovascular Stroke Database, the pretrained U-Net, and a brain atlas. We used the SRI24 atlas[16] to map brain regions to our data. The atlas describes a "standard reference system of normal human brain anatomy." For our purposes, the atlas consists of a volume that has a region label for each voxel and a look-up-table for each region. We resized this volume as needed to match the data from the Endovascular Stroke Database.

The atlas consists of 422 classes using all regions and 96 additional classes clustering these into larger groups. The resulting brain atlas provides an implicit hierarchy, which separates regions into subregions. The single most important level of the hierarchy is the separation of the left and right brain.

Processing

A natural approach to gaining insights into the structure of the examined U-Nets is to visualize the intermediate data of the network. For this purpose, we use activation and saliency maps to analyze the inner computational processes of neural networks and understand the network's computation in image space. Especially for medical researchers, this is an important feature, as it allows them to review the developed visualizations in their usual manner.

Activation Maps

Activation Maps (see Figure 6) describe the activation of the output of each neuron in a given U-Net layer. Combining all respective neurons into a layer and considering the output of a whole layer makes this process more natural for the user. The output of the layer is a set of volumes, which contain an activation value for each voxel.

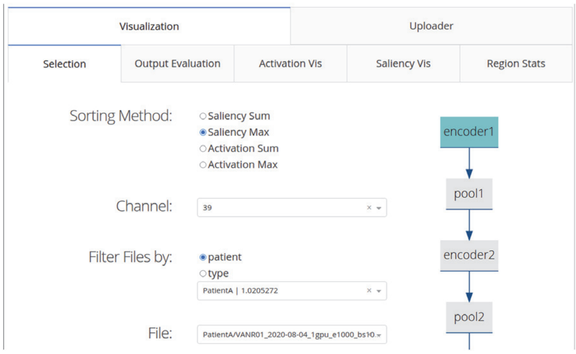
Saliency Maps

The size and dimensions of the imaging volumes themselves can vary substantially. Considering the large number of volumes near the U-Net bottleneck, users will most likely not go through all of them. To help with prioritization, we use Saliency Maps. These look like activation maps, except that a saliency value is given for each voxel indicating which voxels affect the result the most: a positive value indicates support of the classification given by the overall result and a negative value suggests nonsupport. Saliency, as defined by Simonyan *et al.*,[17] is the gradient of the output given by an input. We compute saliency with the AutoGrad system in PyTorch (pytorch.org), which computes the gradient of the function for each node in the computation graph using the chain rule.

Visualization

For reviewing the training process of a U-Net, we provide a visualization system that encodes the different aspects of the network shown in Figure 3. After data processing, the user selects data to be visualized in a Selection View (Figure 4). The Network View (Figure 5) is then used to indicate the importance of different datasets as well as learned features in the network. Finally, for a deeper understanding of the neural network, we provide a Context View (Figure 6) showing the distribution of activation and saliency values for the whole layer. Throughout, we incorporate brain region maps in the slice plots to help the user better visualize spatial information.

Selection View: The closer a layer is to the bottleneck, the larger the number of feature channels there



Sorting Method: ○ Saliency Sum ● Saliency Max ○ Activation Sum ○ Activation Max

Channel: 39

Filter Files by: ● patient ○ type

PatientA | 1.0205272

File: PatientA/VANR01_2020-08-04_1gpu_e1000_bs10

FIGURE 4. The Selection View allows selection and sorting of imaging volumes by attributes such as saliency and activation.

are to view. To facilitate selections within the generated volumes, we implemented a Selection View, which allows selection of volumes according to various attributes. In our case, volumes that have high saliency or activation values are the most interesting. Attributes are measured by summing all values of a volume or by computing the maximum value within a volume. A single scalar value can thus be attributed to every volume. Sorting by this value gives a ranking of all volumes for the selected layer, using either saliency max, saliency sum, activation max, or activation sum. We can also compute the saliency values for the input data and rank all input modalities.

The Selection View provides feedback for the modality that has the highest impact on a given input. A similar approach can be used to compare different patients wherein all modalities are summed so that each patient is assigned one scalar value. The Selection View can be seen in Figure 4.

Network View: To better visualize intermediate data as well as the neural network output, a similar approach is used. Before visualizing, we need to obtain the data from PyTorch. Each layer in the neural network is considered a computation and therefore a node in PyTorch's computation graph. First, a forward pass with the input data is initiated whereby the output of each node is stored and labeled as the activation values of the model. The result of the forward pass is used as the input of a backward pass in which each node applies the gradient of the computation to the given input.

As both of our types of atlas classes are very similar in structure, they can be visualized by the same approach. We use slice plots, where a volume is visualized by a set of color maps. Each color map corresponds to a flat surface given by slicing the volume orthogonally to the *y*-axis, a visualization approach requested by our medical collaborators.

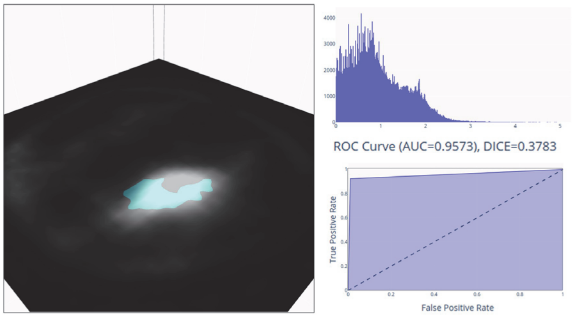


FIGURE 5. The Network View has three components: a slice plot (left), a region plot (upper right), showing the histogram of the selected region, and the AUC (lower right). The slice plot shows the actual output of the network. The cyan regions are classified as a lesion by the ground truth.

For a user-selected y-value, the grayscale color map can then be displayed. Since the output of the neural network is just the activation of the final node in the computation graph, it is visualized in the same way. The only difference is that the output consists of just one volume, whereas the intermediate nodes output multiple volumes.

To give the user a comparison between the selected volume and the other volumes in the layer, a histogram is used (upper right panel of Figure 5). Here, the distribution of all saliency or activation values in the layer is displayed. Since it is not easy to compare the displayed color map with a histogram, we use histogram brushing to make this process easier. After selecting a maximum *max* and a minimum value *min* in the histogram, all the values $v \in [min, max]$ are highlighted in the slice plot (left panel of Figure 5). Considering that the last layer only outputs one volume, the user can opt to highlight the expected output of the neural network to facilitate comparison and draw conclusions about the quality of the prediction. By computing the receiver operating characteristic (ROC) or a Dice similarity coefficient of the expected and the actual output, this quality can also be quantified and visualized (lower right panel of Figure 5).

Context View: To give the user context of the lesion in the brain, we show major brain regions in the slice plots. The user can select a voxel in the plot and all other voxels of that region will also be shown. The highlight color is defined by the color scheme of the SRI24 dataset that we used, as can be seen in Figure 6. For broader context, all the regions in the currently selected slice are displayed in a separate plot. The user can match the selected region in the slice plot by color so that nearby regions can more easily be examined. For both plots, the name of the region is

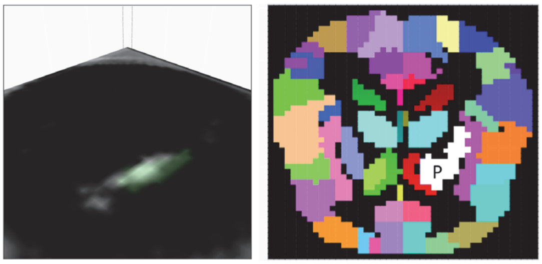


FIGURE 6. The Context View provides an overview of the visualization system. The user can select brain regions of their choosing and these will be highlighted in the slice plot. Here putamen L (P) was selected (right) and the predicted lesion is highlighted in green (left).

displayed when the user hovers over a voxel. Detailed region stats are connected to this visualization for the currently selected volume, e.g., average saliency and activation values for all regions are visualized. This is done using boxplots (Figure 7).

For each layer and region, a boxplot is created that contains all average activation or saliency values for that region. These values can be filtered by model and patient, as shown in Figure 6 for a small set of regions. Here, the putamen L region of the brain was selected (P) and shows in white a significant overlap between the predicted lesion and the selected area of the brain. This area of the brain is known to play a significant role in movement control and would indicate a likely problem for the patient.

ADDRESSING COLLABORATOR QUESTIONS

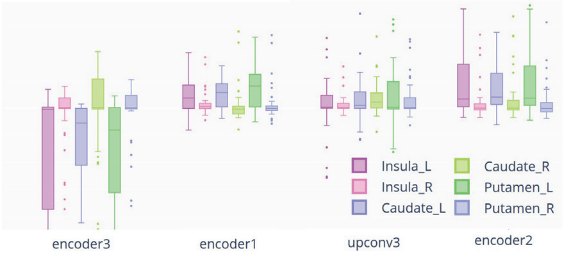


FIGURE 7. Boxplot visualization of brain regions comparing saliency values (*y*-axis) to identify the most interesting regions for a given patient. Colors are determined by the SRI24 dataset.

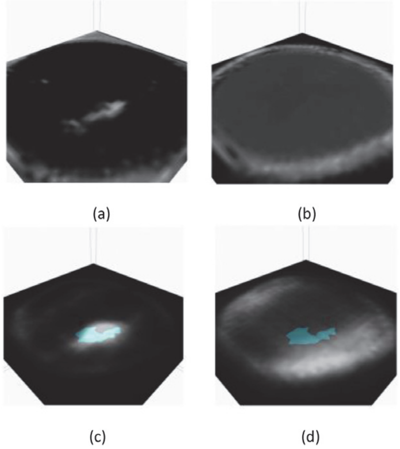


FIGURE 8. Layer 1 activation maps for model ID 0 (a) and model ID 6 (b). Comparison of prediction to ground truth for model ID 0 (c) and model ID 6 (d). Model ID 0 aggregates all imaging modalities and produces the best results.

To test the value of our approach, we used our system to address a number of questions posed by our collaborators, in examining the 10 models shown in Table 1.

Does the database show any flaws?

We utilized our tool to sort the patients according to the AUC of the respective lesion prediction. Based on this sorting, we reviewed the 5 most inaccurate predictions for all models shown in Table 1. We found that the lesion of a patient is predicted less accurately when there is no labeled lesion in the final ground truth. These cases do occur, but the database only contains 9 patients that were treated by a special surgery that was able to prevent damage to their brains. Unfortunately, the number of these patients is very low which inhibits proper training of the U-Net.

Another problem arose for a group of patients that also had an “old” lesion. Here, we determined that the neural network is not able to separate the old and new lesions and tries instead to predict both, which implies that training should incorporate a further label to indicate this separation.

Are all input modalities required?

Figure 8 shows the comparison of models 0 which uses all modalities and model 6 which does not. Figure 8(a) and (b) show the activation values of the first encoder layer in the neural network for model 0 and 6, whereas Figure 8(c) and (d) show the activation values in the output layer of the U-Nets 0 and 6 in white with an overlay of the putamen region (cyan). This comparison indicates that the lesion is detected much faster in model 0, which utilizes all input modalities. We observe that models trained with all modalities start identifying the lesion in earlier layers compared to models that did not, which suggests that using all input modalities leads to more efficient training of the neural network.

Also, spatial information remains intact when all input modalities are used, but models trained solely with CTA and NCCT do not show this property. This reinforces our hypothesis that using all image modalities is better for training.

Do some imaging modalities have more influence on decision making than others?

For the tested models, we examined the activation and saliency maps within different layers of the examined U-Nets. It can be seen that for all models a relatively low activation throughout the CTA and NCCT modalities occurs. This results in good saliency values for these modalities. We can conclude that the parameter maps derived from CTP have a higher influence on the decision-making process of the U-Net. This is also strengthened by the saliency values of the Tmax modality, which has the highest saliency values of all modalities. Thus, the T-max modality is the most important modality in terms of interpretability of the neural network. Still, T-max should be combined with other modalities, as it does not achieve a precise prediction result on its own (see Table 1, Model 9).

In our experiments, the CTP T-Max modality is found to have a greater influence on the outcome of the prediction when used in training, increasing interpretability. CTA and NCCT, when used in conjunction with Tmax, act as a regulating factor and improve the accuracy of the prediction as shown by the ROC-curve.

Which parts of the input images are used by the network to make a decision?

As shown in Figure 8, the neural network first eliminates voxels that are not part of the patient but there is not any obvious separation beyond that. The highly specialized filters in deeper U-Net layers cannot be easily compared, as they adapt to each patient individually. The deeper one gets into the network the more variability is represented in the captured feature channels. Still, we can observe that the network eventually narrows down the area where lesions can be located, starting from the head of the patient, continuing with the two sides of the skull and resulting in the area where the lesion is located.

In general, the network does not “think” about brain regions as defined in the brain atlas. The only observation we can make is that the right area of the brain seems to be considered more strongly than the left. The most important use of the brain map is to indicate which areas of the brain will probably be affected by a lesion.

What features is the model looking for?

Figure 8 shows the activation map of the first layer, in which the network is primarily locating the head of a patient, though it already seems to point out the lesion. When moving toward the bottleneck of the neural network, complex concepts are broken down into smaller ones, but this also means that the number of feature channels is increasing. This makes reviewing the network more challenging, as all feature channels need to be visually inspected to form a hypothesis.

Still, we can say that the closer a layer gets to the bottleneck, the more patient-specific its feature maps become, though locating the head in the first layer is very similar for all patients.

Is there a relation between functional areas in the brain and computed features in the model?

In Figure 7, we analyzed three important areas of the brain, the insula, caudate, and putamen, differentiated by left and right sides. We reviewed the saliency values of each of these classes, as shown in Figure 7. We see that regions which end up containing the lesion (in this case the left hemisphere) result in higher saliency values than their counterpart on the other hemisphere. This behavior is especially pronounced in models that use all the input modalities.

How does the model treat individual patients in relation to the others?

Our visualization approach allows us to rate different patients in the Endovascular Stroke Database according to their importance in the neural network. The sorting according to saliency and activation allows us to review which patients are most important in the neural network prediction process.

Figure 9 shows two feature channels in the third layer of network 6. We can see that the network separates the boundary of the patient’s head into two different parts. Figure 9(a) shows the 53rd feature channel of a patient in layer 3, which indicates that the skull is detected, while Figure 9(b) shows that channel 91 is searching for another part of the skull. Hence, the deeper the computation enters a network, the more clearly the features get separated.

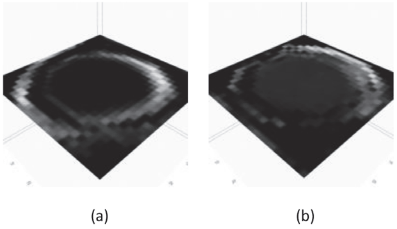


FIGURE 9. Comparison of two feature channels of a patient in layer 3 of model ID 6. The channels show that different parts of the skull are detected by the neural network.

DISCUSSION

Benefits and Drawbacks

Using our visualization system, we are able to retrace the steps of how the chosen U-Net models make decisions. We can identify which datasets and voxels within a dataset contribute to the decision making more than others. Including the brain atlas allows us to determine where in the brain a lesion is located and which parts of the brain are used by the network to make this decision.

However, other steps are still hard to retrace. This especially applies to layers that are close to the bottleneck layer, as the respective feature maps have many channels but only a small number of voxels spatially. Our system does seem to have a preference for the right hemisphere of the brain although the reason for this behavior remains to be explored.

User Feedback

Our medical collaborators were initially reluctant, but after seeing what XAI could do, they were very impressed with the results. However, our initial group of collaborators is relatively small, and a formal user study would not likely result in statistically significant results. Still, our collaborators gave us some very motivating feedback:

- *It is very interesting to see how the neural network becomes more specialized throughout its computation. I was not aware of that.*
- *I am very glad to have a visual indicator that helps me improve the Endovascular Stroke Database.*
- *I would like to encourage you to research the possibility of putting the found features into a mathematical description. This could help to create a better description of strokes in general.*

Future Directions

We believe our visualization approach will be useful for examining neural networks from other application domains, a topic we hope to explore further.

In future work, we also aim to extend our analytic system so that it can visually indicate the uncertainty contained in neural networks, a task we have shown to be of great importance in the medical imaging field.[18]

ACKNOWLEDGMENTS

The authors would like to thank the Department of Neurology of the University Hospital at Leipzig University for providing us with the Endovascular Stroke Database.

REFERENCES

1. R. Adolphs, "Human lesion studies in the 21st century," *Neuron*, vol. 90, no. 6, pp. 1151–1153, 2016.
2. V. L. Feigin, B. Norrving, and G. A. Mensah, "Global burden of stroke," *Circulation Res.*, vol. 120, no. 3, pp. 439–448, 2017.
3. H. Seo *et al.*, "Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications," *Med. Phys.*, vol. 47, pp. 148–1167, May 2020.
4. A. Holzinger *et al.*, "What do we need to build explainable AI systems for the medical domain?," *Clin. Orthopaedics Related Res.*, Dec. 2017.
5. R. G. C. Maack *et al.*, "Towards closing the gap of medical visualization research and clinical daily routine," in *VisGap—The Gap Between Visualization Research and Visualization Software*, C. Gillmann, M. Krone, G. Reina, and T. Wischgoll, Eds., Aire-la-Ville, Switzerland: The Eurographics Assoc., 2020, pp. 25–233.
6. E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2020.3027314.
7. G. J. Katuwal and R. Chen, "Machine learning model interpretability for precision medicine," 2016.
8. A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *J. Imag.*, vol. 6, no. 6, 2020, Art. no. 52.
9. K. Lawonn *et al.*, "A survey on multimodal medical data visualization," *Comput. Graphics Forum*, vol. 37, pp. 1–25, Oct. 2017.
10. K. Schardt *et al.*, "Multi-modal visualization of stroke lesion CT-imaging," *Neurology*, vol. 95, pp. e2954–e2964, Nov. 2020, doi: 10.31219/osf.io/qk39a.

11. C. Gillmann *et al.*, “Uncertainty-aware brain lesion visualization,” in *Proc. Eurograph. Workshop Visual Comput. Biol. Med.*, 2020, pp. 97–101.
12. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, Berlin, Germany: Springer, 2015, pp. 234–241.
13. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, Jul. 2015, vol. 37, pp. 448–456.
14. P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 958–963.
15. P. Gholizadeh, B. Esmaeili, and B. Memarian, “Evaluating the performance of machine learning algorithms on construction accidents: An application of roc curves,” in *Proc. Construction Res. Congr.*, 2018, pp. 8–18.
16. T. Rohlfing *et al.*, “The sri24 multichannel atlas of normal adult human brain structure,” *Hum. Brain Mapping*, vol. 31, no. 5, pp. 798–819, 2010.
17. K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” in *Proc. Workshop Int. Conf. Learn. Representations*, 2014.
18. C. Gillmann *et al.*, “Uncertainty-aware visualization in medical imaging—A survey,” *Comput. Graphics Forum*, vol. 40, no. 3, pp. 665–689, 2021.

CHRISTINA GILLMANN is a Researcher of the Signal and Image Processing Group with the University of Leipzig. She is the corresponding author of this article. Contact her at gillmann@informatik.uni-leipzig.de.

LUCAS PETER is a Student Assistant of the Signal and Image Processing Group with the University of Leipzig. Contact him at lp25sidy@studserv.uni-leipzig.de.

CARLO SCHMIDT is an IT Consultant at Empolis Information Management GmbH. Contact him at carlo.schmidt@empolis.com.

DOROTHEE SAUR is a Head Physician at the Neuroradiology Department, Medical Centre, Leipzig University. Contact her at dorothee.saur@medizin.uni-leipzig.de.

GERIK SCHEUERMANN is the Leading Professor of the Signal and Image Processing Group with the University of Leipzig. Contact him at scheuermann@informatik.uni-leipzig.de.

Contact department editor Mike Potel at potel@wildcrest.com.

EDITOR: Amir Dabirian, amir@fullerton.edu

This article originally appeared in IT Professional vol. 25, no. 1, 2023

COLUMN: IT TRENDS

Implementing Behavioral Biometrics With TRUST

Jayson Killoran, Smith School of Business, Queen's University, Kingston, ON, K7L 3N6, Canada

Yuanyuan (Gina) Cui, Auckland University of Technology, Auckland, 1010, New Zealand

Andrew Park, Gustavson School of Business, University of Victoria, Victoria, BC, V8W 2Y2, Canada

Patrick van Esch, Kennesaw State University, Kennesaw, GA, 30144, USA

Amir Dabirian, California State University, Fullerton, Fullerton, CA, 92834, USA

Jan Kietzmann, Gustavson School of Business, University of Victoria, Victoria, BC, V8W 2Y2, Canada

Companies using behavioral biometrics to monitor employee performance risk creating a dangerous tension in the workplace. Implementing behavioral biometrics with TRUST (transparency, respect, understanding, sharing, and timing) may be the solution.

Behavioral biometrics—a technological evolution where patterns in human movement and activities are identified, captured, and analyzed to optimize organizational processes—provide a new world of opportunities for managers and employees alike.

For some time now, employees have been subjected to physiological biometrics, mostly for security purposes. These include fingerprint or iris scanning as well as facial and voice recognition. However, these authentication processes only produce surface biometrical data that allow managers to track employees' times and locations. They say little about what happens at work.

As technologies for capturing and analyzing behavioral data mature, behavioral biometrics offer managers deeper, richer insights about their employees' conduct. Managers can glean new information based on employees' keystrokes, hand and body movements, heart rates, voice inflections, and even brain activity. Do supermarket cashiers or casino dealers smile enough? Do their smiles appear genuine or fake? Such behavioral attributes can be directly tied to the organizational bottom line.

The tension is clear. Organizations stand to gain a lot of control, whereas their employees fear that revealing such personal and private data will lead to a loss of discretion at work. A recent PwC study revealed that 88% of executives claimed biometric technology makes their business lives better, whereas only 48% of employees agreed.[1] To help close this gap, we suggest that companies implement behavioral biometrics with TRUST: transparency, respect, understanding, sharing, and timing (Figure 1).

TRANSPARENCY

The implementation of behavioral biometrics needs to start with transparency—a rather novel suggestion given recent technology implementation trends. As employees continue to work in hybrid work arrangements, suspicious managers increasingly monitor them using "bossware" or "tattleware."[2]

This monitoring software can scour employees' social media posts, log their keystrokes; take intermittent screenshots; access live video feeds; start webcams to watch their facial expressions; and even measure their cognitive load, stress, and attention levels. This happens both in the home office and at work.

Managers seem to believe that monitoring should happen without explicit disclosure and consent; otherwise employees would evade the technology or alter their behavior to manipulate the data. Consequently, most of these technologies are remotely installed or inconspicuously hidden in other software. If employees discover the existence of these technologies, they justifiably feel that their right to privacy has been compromised and that their superiors fundamentally do not trust them.

1520-9202 © 2023 IEEE
Digital Object Identifier 10.1109/MITP.2023.3236532
Date of current version 10 March 2023.

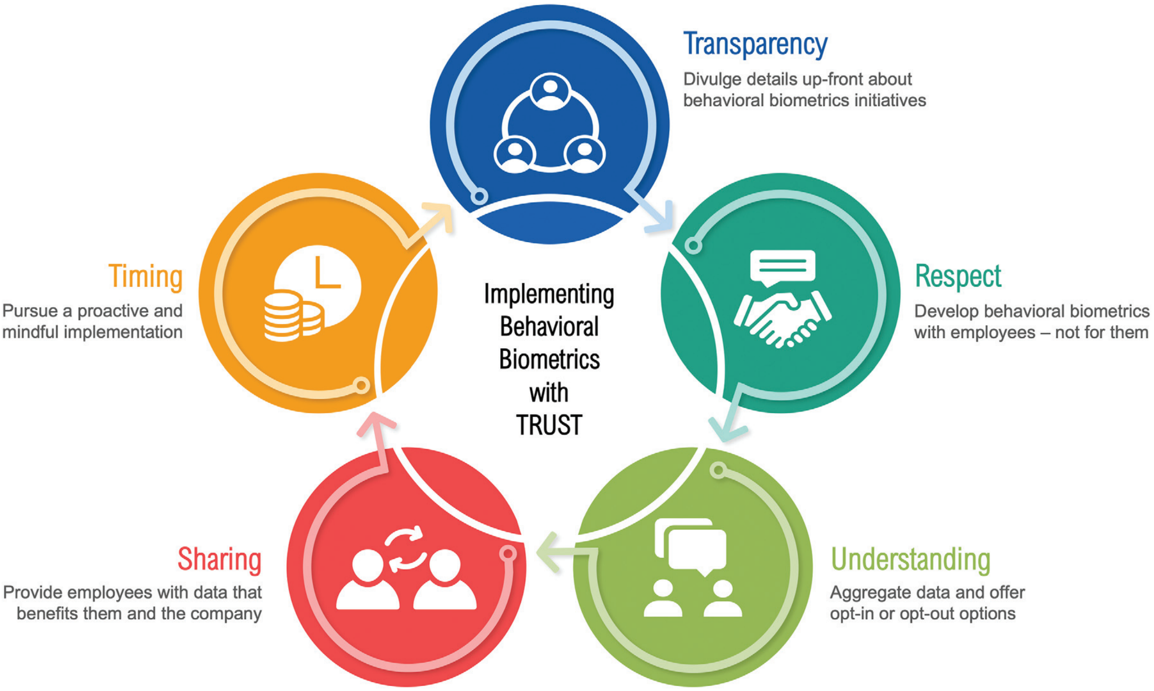

Transparency
Divulge details up-front about behavioral biometrics initiatives
Respect
Develop behavioral biometrics with employees – not for them
Understanding
Aggregate data and offer opt-in or opt-out options
Sharing
Provide employees with data that benefits them and the company
Timing
Pursue a proactive and mindful implementation
Implementing Behavioral Biometrics with TRUST

FIGURE 1. Implementing behavioral biometrics with TRUST.

A fresher approach is needed to combat the long-established us-versus-them history of IT-based distrust between managers and employees. Full transparency around the rationale for and processes of data collection not only reduces distrust but actively builds institutional trust when managers openly talk with employees about

- *why* behavioral biometrics are needed;
- *what* specific behavioral data will be collected;
- *where* the behavioral data will be stored;
- *who* (else) has access to the behavioral data;
- *how* the results will be used to help improve individual and organizational performance; and
- *when* and *how* the data will be destroyed.

BHP Billiton, the second largest mining company in the world, uses "smart caps" to track miners' levels of fatigue and drowsiness. They obtained increased levels of employee buy-in by being up front about the type of data collected, how smart caps detect fatigue, and why this information is important both for the miners' safety and well-being and for organizational success.

RESPECT

The second building block of TRUST is manifested when employers expand transparency through inclusive decision making. When managers empathize with their employees by forming decisions together, they not only demonstrate emotional intelligence but also strong IT leadership skills.

Especially at a time when both employers and employees are concerned that their organizations handle data responsibly, including both parties in decisions related to the implementation of behavioral biometrics builds a culture of trust, inclusion, shared purpose, and responsibility.

Respect is demonstrated when organizations develop behavioral biometrics with their employees, not for them. This can take many forms, from including employees in initial behavioral biometric design activities to involving them in subsequent IT decisions. At the very least, employees will remind managers to go on a data diet—only collecting data when they can demonstrate a clear purpose.

Ricardo Semler, CEO of Semco Partners, sent a strong message to employees by always keeping two board seats open for employees, with equal voting rights. By doing so, Semco empowered employees to have a say in important decisions and kept executives informed of significant employee concerns.

UNDERSTANDING

When companies demand or even coerce employees to agree to be monitored as part of their employment conditions, they do not build trust. Rather, this practice highlights the power imbalance between employers and employees, where one gives orders and the other has no choice but to follow. Despite involving employees in behavioral biometric decisions, there remain justifiable objections to volunteering highly personal data.

Even when included in behavioral biometric decision making, employees might remain skeptical about revealing private data. If these concerns are not taken seriously, the practice can potentially backfire. Ironically, performance monitoring tools can actually diminish performance.[3] When monitoring technologies are implemented poorly, innovativeness, creativity, reflection, and collaboration suffer, and costly instances of employee sickness and burnout increase.

Understanding the dilemma around this power dynamic, we propose two viable recommendations:

1) *Building differential privacy by separating what employees and employers see*: This allows managers to receive useful aggregate behavioral metadata while, at the same time, protecting the anonymity of individual employees.
2) *Offering employees either an opt-in or opt-out option when it comes to data collection without any negative consequences*: Power relationships are balanced when job prospects, bonus payments, professional development opportunities, or promotion are not influenced by an employee's consent to behavioral biometrics.

Humanyze, a Massachusetts Institute of Technology-born people analytics company, provides employees with wearable ID badges to learn who interacts with whom and for how long and measures employees' stress levels based on heart rate and voice inflection. In anticipating strong objections, Humanyze decided to deploy this technology only to employees who opt in. Everyone else received an identical yet nonfunctional badge.

SHARING

With the option to opt in or opt out, why would anyone participate? The TRUST-building answer lies in sharing and co-owning the benefits of behavioral biometrics for both the company and employees alike.

Employees have an intrinsic motivation to increase effectiveness, to understand the potential risks of workplace injuries, and to improve time management skills to avoid burnout. Behavioral biometrics also offer valuable insights into how employees can tweak their performance, especially when compared to organizational benchmarks.

Moreover, employees can also leverage their own behavioral biometric data for extrinsic reasons. Examples abound—employees can use behavioral data from meetings after working hours to negotiate more time off or use voice data to showcase how they managed difficult customer encounters to gain recognition or rewards. This kind of bias-free evidence can also help employees during performance evaluations, when preparing applications for professional development, or when highlighting key skills on their résumés.

Microsoft manages the benefit-sharing process well by giving employees choices and shared access to data. Staff receive confidential reports combining organizational benchmarks with the employees' personal performance and behavioral work patterns. This incentivizes employees to participate while offering them the option to remain anonymous, even from their managers.

TIMING

In most jurisdictions, certainly in the United States, these recommendations are largely optional. Managers who understand the disruptive potential of behavioral biometrics will not wait for the U.S. Department of Labor or similar employment tribunals to mandate standards and compliance-monitoring programs for behavioral biometrics. Instead, leaders know that they are accountable for managing the process responsibly, especially in the absence of regulation. Responsible managers proactively and voluntarily develop a practical framework for behavioral biometrics.

While managers approach the topic with a sense of urgency, they understand that successful IT implementation takes time. Otherwise, short-term gains produced by behavioral insights will likely lead to long-term losses to organizational culture. As a result, leaders resist the temptation to quickly adopt one of numerous tools available now and, instead, focus on openly involving all stakeholders, policies, and practices that are impacted by the new technology.

TRUST MATTERS

Behavioral biometrics are quickly becoming the norm of performance monitoring. By capturing rich behavioral data, these technologies certainly have the potential to offer compelling insights into how organizations truly work. However, just because we can measure and monitor everything does not mean we should. As the appetite for more behavioral data increases, precaution measures—like the TRUST framework—should drive the implementation.

The full potential of behavioral biometrics can only be realized with managers' commitment to transparency, respect for employees, understanding of the importance of informed and optional consent, sharing and co-ownership of the benefits, and timeliness in their approach. By doing so, managers demonstrate good intentions through their concerns for both people and processes. By implementing behavioral biometrics with TRUST, companies can generate data-driven managerial value, build a strong brand, attract and empower high-caliber employees, and develop a far-reaching reputation as being a great place to work!

REFERENCES

1. C. Duarte, D. Staley, and B. Sethi. "Our status with tech at work: It's complicated." PwC. Accessed: Jan. 20, 2023. [Online]. Available: https://www.pwc.com/us/en/services/consulting/library/images/PwC_CIS-Tech-at-Work.pdf
2. J. Crispin, "Employers are spying on Americans at home with 'tattleware'. It's time to track them instead," *Guardian*, Sep. 16, 2021. [Online]. Available: https://www.theguardian.com/commentisfree/2021/sep/16/tattleware-employers-spying-working-home
3. D. L. Tomczak, L. A. Lanzo, and H. Aguinis, "Evidence-based recommendations for employee performance monitoring," *Bus. Horiz.*, vol. 61, no. 2, pp. 251–259, Mar./Apr. 2018, doi: 10.1016/j.bushor.2017.11.006.

JAYSON KILLORAN researches management information systems with the Smith School of Business at Queen's University, Kingston, ON, Canada. Contact him at j.killoran@queensu.ca.

YUANYUAN (GINA) CUI researches marketing with the Department of Marketing at Auckland University of Technology, Auckland, New Zealand. Contact her at yuanyuan.cui@aut.ac.nz.

ANDREW PARK is an assistant professor of information systems at the University of Victoria, Victoria, BC, Canada. Contact him at apark1@uvic.ca.

PATRICK VAN ESCH is an assistant professor of marketing and professional sales at Kennesaw State University, Kennesaw, Georgia, USA. Contact him at pvanesch@kennesaw.edu.

AMIR DABIRIAN is the vice president for the division of information technology and a faculty member with the Department of Marketing, California State University, Fullerton, CA, USA. Contact him at adabirian@fullerton.edu.

JAN KIETZMANN researches management information systems and innovation at the University of Victoria, Victoria, BC, Canada. Contact him at jkietzma@uvic.ca.

EDITOR: Schahram Dustdar, dustdar@dsg.tuwien.ac.at

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

This article originally appeared in Internet Computing vol. 27, no. 1, 2023

Toward Building Edge Learning Pipelines

Anastasios Gounaris and Anna-Valentini Michailidou, *Aristotle University of Thessaloniki, 541 24, Thessaloniki, Greece*

Schahram Dustdar, *Technical University of Vienna, 1040, Vienna, Austria*

From a bird's eye point of view, large-scale data analytics workflows, e.g., those executed in popular tools, such as Apache Spark and Flink, are typically represented by directed acyclic graphs. Also, they are in a large scale in two dimensions: first, they are capable of processing big data (e.g., both in terms of volume and velocity) mainly through employing massive parallelism, and second, they can run over (powerful) distributed infrastructures. This article focuses on edge computing and its confluence with big data analytics workflows, which nowadays place special emphasis on deep learning and data quality.

Early examples of large-scale data analytics workflows were primarily MapReduce programs,[1–3] which, however, could only handle nonstreaming data over fixed data centers; streaming data analytics was initially evolving rather independently,[4] but nowadays, massive parallelism for processing data streams is the norm.[5] There were also several efforts that have tried to extend database query plan optimization technology to account for arbitrary user-defined functions, so that such plans can correspond to generic analytics workflows extending traditional ETL data pipelines.[6–9] Overall, it has been shown that database technology can offer a lot to large-scale data analytics workflows from its several decades of experience in terms of declarativeness and principled (cost based) optimization.[6,7,10,11]

More recently, research emphasis regarding executing big data analytics tasks is placed on the following topics.

- High-level system details, such as 1) derivation of the exact requirements from the software engineering point of view and the exact architecture to be adopted,[12] or 2) the data models underlying big data analytics.[13]
- Low-level execution engine details, e.g., with regards to aspects, such as state management.[14] This is also related to tuning and optimized resource usage in complex systems for big data analytics, such as Hadoop and Spark[15,16] along with service level agreement (SLA) management when such systems are deployed in the cloud.[17]
- In addition to the advances abovementioned, several efforts advocate taking a more holistic view in modern data analytics, i.e., address all components and steps involved in real applications, from storage to user interfaces and DevOps, and from data preparation to (iterative) model building and validation. Also, in practice, complete ecosystems need to be developed around processing engines for big data analytics.[18–20]

In all the aspects mentioned thus far, mature industry-level solutions exist and are adopted by both researchers and practitioners. Nevertheless, these aspects are not adequate for supporting large-scale data analytics workflows to their full extent, as defined at the beginning of this report. This is because the current solutions cannot support deployment over arbitrarily resource-constrained distributed computing infrastructures. Modern data analytics engines are decoupled from fixed data centers and are moved to cloud solutions, but their deployment remains largely centralized. In other words, there is a lack of mature support of deployment of data-intensive analytics workflow jobs over widely heterogeneous multiowner geo-distributed edge, fog and/or cloud resources. However, common IoT and edge computing settings are characterized by all these three factors, namely, 1) heterogeneity in several dimensions including resource characteristics, availability,

1089-7801 © 2023 IEEE
Digital Object Identifier 10.1109/MIC.2022.3171643
Date of current version 3 February 2023.

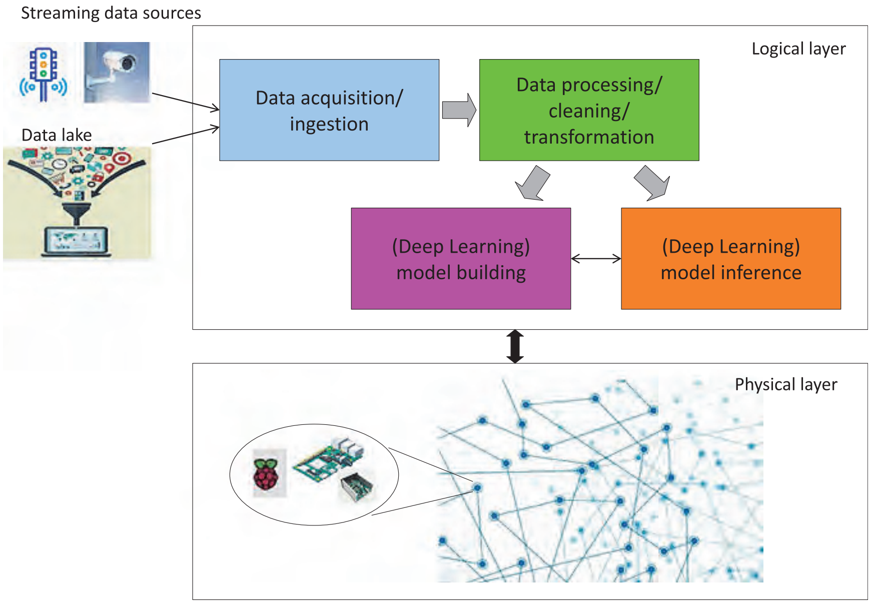

Streaming data sources
Data lake
Logical layer
Data acquisition/ ingestion
Data processing/ cleaning/ transformation
(Deep Learning) model building
(Deep Learning) model inference
Physical layer

FIGURE 1. Example of an edge data acquisition, processing, and learning pipeline.

permissions and connection speeds, 2) geographical distribution, and 3) multi-ownership. Such settings are thus not adequately covered.

Our motivating remark: Future technical advances in data analytics pipelines should target to fill this gap, i.e., to account for heterogeneity, geographical distribution, and multiownership in large-scale data analytics workflows. To this end, there are several initiatives to adapt data-center-oriented solutions, such as Hadoop and Spark, to heterogeneous geo-distributed settings.[21,22] But all these fall short in dealing with primary concerns that edge and fog computing realms entail in a holistic manner. In Bansal *et al.*'s work,[23] which discusses the confluence between IoT and Big Data, several challenges are identified with regards to aspects, such as volume, velocity, variety, veracity, value, variability, visualization, validity, vulnerability, volatility, venue, vocabulary, and vagueness. Even when treating single aspects, such as venue, in isolation, the corresponding research is rather in its infancy, and many aspects are typically not considered. For example, for venue, commonly employed schedulers, resource managers, and orchestrators, such as Kubernetes, YARN, and MESOS, cannot place tasks at arbitrary geo-distributed places in a judicious manner. But this is just one part of the complete picture. It is important to acknowledge that, especially in an edge computing setting, there is a growing and demanding need for 1) treating data quality aspects as a first-class citizen and 2) move complex deep learning model training and inference to the edge,[24] which adds significant complexity to the analysis pipelines.

VISION FOR NEXT-GENERATION EDGE-ENABLED BIG-DATA ANALYTICS WORKFLOWS

Next generation edge-enabled big data analytics workflow solutions should not only address the current limitations but also go beyond them. We envisage a solution that would not only allow to run every analytics task everywhere but can also detect the appropriate data sources to feed the analysis tasks in an automated or at least semiautomated manner. By *everything*, we cover, for example, intelligent deep learning model training and inference. By *everywhere*, we cover cases, where a federation of low-end edge/fog devices forms the computation infrastructure to execute the workflows. For this vision to be realized, it is important to blend data lake technologies with edge learning so that local model training can benefit from all the relevant data required rather than locally produced ones solely.

Imagine a smart-city scenario, where advanced deep learning model construction, and inference are deployed on edge devices, e.g., to reduce latency.[25] In the rest of this article, edge and fog devices will be used interchangeably for simplicity. Such a scenario includes always-on surveillance coupled with the ingestion of data streams from third parties, e.g., to acquire meteorological conditions data. Similarly, in many application domains benefiting from edge learning, such as smart health and agriculture, it is common to join multiple data sources.[26] Retail is another field that can benefit from edge analytics. In this application domain, the real-time big data are collected through various methods, including video cameras, basket analysis, POS terminals, and customer memberships.[27–29] Moreover, by combining these data with data from data lakes, for example, social media posts, demographic features, and customer information,[30] retailers can forecast product demand, predict customer purchases, provide personalized advertisements, discover trends, and grow their overall profit, all these based on edge learning techniques.

Edge learning implies several additional features of the corresponding workflow: 1) model building needs to be parallelized across several different computing nodes in an efficient heterogeneity-aware manner,[31–33] 2) data need to be shared only partially and after ensuring that any privacy concerns are addressed,[34] and 3) intensive data-quality actions, such as outlier removal need to take place to avoid data poisoning.[35]

The workflow, depicted as a DAG, comprises four main groups of tasks corresponding to data acquisition, data processing, model building, and model inference, respectively (see Figure 1). Each of these groups is something broader than, for instance, a single stage in Spark. Tasks may interact in complex manners and they may also involve human interaction.

Our vision includes the following three pillars.

1) Allow end users to define complex workflows in a mostly declarative manner, and these workflows to run on top of any (edge/fog) computational infrastructure judiciously in a massively parallel manner. This entails optimal resource usage and task allocation taking into account a wide range of data quality and optimization criteria, well beyond 2 or 3 typically employed in modern multiobjective scheduling/task allocation solutions.
2) Encapsulate integration with data lakes technology, possibly involving novel human-in-the-loop architectures to semiautomatically, detect the appropriate data sources that feed the remainder of the complex workflow analysis pipelines.
3) Account for edge learning scenarios, which impose strict constraints on which data can be shared and may require the existence of mutually trusted central cloud nodes that become responsible for specific parts of the model construction.

Building the aforementioned pillars should cover the big-data aspects in the Bansal *et al.*'s[23] work, also termed the 13 V's, as summarized in Table 1. In the rightmost column, we mention the challenges involved, which range from dealing with novel data and task placement problems to integrating data quality detection and improvement solutions, and appropriate source detection in data lakes.

Edge learning improvements: Our vision can be deemed as a call for extension to the state of the art in edge learning,[24] which currently focuses on building ML models over edge devices in a collaborative manner, while considering data, computational, communication, privacy, security, and incentive-related challenges. As reported in the Deng *et al.*'s[36] work, not only data analytics on the edge, but even the more restrictive scenario of AI on the edge employing a limited set of optimization criteria is a topic that requires much deeper investigation. In any case, the extensions are very important in that

1) they do not separate data acquisition and processing pipelines with the model training/inference ones;
2) they account for the full spectrum of big data aspects; and
3) they call for novel workflow management, i.e., expression, execution, and scheduling techniques.

These extensions are further analyzed in the following.

Toward Next-Generation Edge Learning: A Closer Look at the Three Axes

First, integrating data lake technology with database engines has already been identified as a key research direction[37]; what we advocate is such an integration to also cover the edge learning workflows that we aim to run over geo-distributed edge nodes. There are three main problems that are encountered:

(1) detection of the most appropriate sources
(2) optimized sharing of data across all nodes that run model construction tasks and may benefit from such sources
(3) including humans in the loop.

Why this is challenging? If a single computational node becomes responsible for source detection, this node may easily become a bottleneck. However, if multiple nodes undertake this task, it is unclear how to split the corresponding workload and synchronize their searching process. Finally, having the human-in-the-loop leads to the development of a whole new family of techniques.

Second, covering the full spectrum of Big Data aspects is strongly connected to meeting the 13 V's requirements abovementioned. In addition to the presentation in Table 1, which explains how all big data aspects are important in our vision, data quality issues

TABLE 1. Issues and challenges in multifaceted coverage of big data aspects in our vision.

Aspect	Description of impact on the solutions	Challenge
Volume	Relates to maximizing throughput, leveraging massive parallelism, moving filtering operations as close to the data sources as possible, reusing data, minimizing data transfers, and so on.	Data cannot be arbitrarily shared, which renders existing techniques inefficient or even inapplicable.
Variety	Relates to considering all kinds of resource heterogeneity involved.	The variety covers both computational and networking infrastructure and the local datasets available on each edge device, the combination of which is not currently considered.
Velocity	Emphasizes on minimizing latency, heavily relies on massive parallelism on top of heterogeneous resources, and poses restrictions on where model inference can run.	Incurs tradeoffs when deep learning models are large and need to be split across multiple nodes.
Veracity	Calls for detecting the most appropriate and trustworthy data sources on the fly.	Calls for the development of novel data lake-aware edge learning solutions that emphasize on both the training and the data acquisition process.
Value	Relates to including intelligent analytics and machine learning (ML) steps in the workflows apart from simpler data management tasks.	Such intelligent analytics may require synchronizations, which are difficult to be attained in a heterogeneous setting.
Variability	Relates to the capability of the solution to adapt to environmental changes, i.e., any task/data placement solutions may need to be adaptive.	Tasks are typically stateful.
Visualization	Relates to the fact that human-in-the-loop is a key distinctive feature (as also in the Industry 5.0 vision).	Impacts on metrics, such as latency, in a non-straightforward manner.
Validity	Envisaged as including data quality checks and enforcement steps as first-class citizens (in addition to data management and ML operations).	Data quality can be quantified in several manners and is not typically considered using execution plan optimization.
Vulnerability	Calls for addressing privacy and security requirements, an issue of paramount importance in edge learning.	Involves tricky tradeoffs with performance and placement flexibility.
Volatility	Calls for continuously refining analysis results as more data are produced, that is, the corresponding analytics workflows should run continuously to both refine and apply trained models.	Calls for novel techniques to reduce operations and data transmissions when no changes from previous values and/or results are detected/predicted.
Venue	Relates to the judicious placement of tasks to resources.	Need to account for resource heterogeneity and geographical distribution.
Vocabulary	Relates to the development of higher level (declarative) abstractions to describe tasks, resources, constraints, objectives, and so on.	No standardized approach exists to date for the relevant aspects.
Vagueness	Complements validity and veracity.	Same as validity and veracity abovementioned.

should be further emphasized. Data quality aspects are defined in multiple manners. For instance, in Deequ,[a] Apache Griffin,[b] and Great Expectations,[c] simple data checking operations are defined and implemented. However, data quality aspects can be described more broadly, e.g., through ISO-25012[d] with a view to covering the veracity, validity, and vagueness big data dimensions. In this standard, there are several relevant aspects of data quality. For example, completeness relates not only to the desire the input data to have non-NULL values but also all the corresponding data for model training to be available. Also, precision monitors IoT streams for unjustified data fluctuations, which are attributed to sensor malfunction; assessing precision in this sense entails the insertion of a lightweight statistics module in the complete analysis pipeline. As a third example, credibility relates to the accuracy of an ML model and is affected by the presence of a human in the loop. Data quality aspects are also directly relevant to optimization objectives, e.g., timeliness relates to the pipeline performance and its capability to perform model refinement and inference with low latency.

More specifically, examining the 15 data quality characteristics of the ISO-25012, we can extract eight of them, as presented in Table 2, which can be directly mapped to optimization objectives and encapsulation of data quality-oriented tasks in the pipeline. The other seven data quality characteristics are also relevant, but they cannot be easily quantified in our context, e.g., accessibility, understandability, and portability. The quantitative metrics in the table complement performance metrics, such as throughput, latency, power, resource, and network utilization, which are well understood.[4] Also, it is still important to consider quality of service (QoS), which may be deemed as quantifying accuracy after load shedding or can be application dependent.

Third, workflow management should be geared toward more declarativity, well beyond merely employing and calling complex ML libraries through user-friendly scripts, as is the main status to date.[10] The extended set of optimization criteria and constraints raise the need for a convenient manner to express them; similarly, the user feedback needs to be in a format amenable to immediate processing and enactment of corresponding actions, e.g., with regards to source selection. Thus, there is an interplay of expression and execution. Execution is also affected by the data and task placement decisions that also need to be controlled, at least partially, in a declarative manner. This implies changes in the underlying resource managers, negotiators, and schedulers.

TABLE 2. Data quality characteristics, as defined in ISO-25012 and the corresponding envisaged optimization metrics and tasks in data analytics pipelines.

Characteristic	Quantitative metric	Corresponding task
Accuracy	Degree to which values of ingested data deviate from their reference values.	Measure the accuracy; choose data sources based on their accuracy values.
Completeness	Number of data features extracted from external data sources employed in model building.	Seek for relevant and combinable data sources.
Consistency	Degree to which values for the same features from different sources are aligned.	Measure the consistency; consider consistency when choosing data sources.
Credibility	Degree to which data and models built is believable by users.	Receive human feedback on the credibility of external data-lake-based sources and ML models.
Currentness	The time difference between data generation and data processing.	Assess the currentness (also referred to as timeliness).
Compliance	Degree to which fields such as timestamps follow standards.	Task to assess compliance; choose data sources based on their compliance values.
Precision	Degree to which sensing mechanisms produce precise measurements.	Assess the measurements fluctuations due to sensing mechanism imprecision.
Availability	Degree to which external data sources are available.	Profiling of the availability of external sources.

[a] [Online]. Available: https://github.com/awslabs/deequ
[b] [Online]. Available: https://griffin.apache.org/
[c] [Online]. Available: https://github.com/great-expectations/great_expectations
[d] [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25012

TECHNICAL NOTES REGARDING THE RESEARCH ISSUES INVOLVED

The abovementioned discussion entails and touches upon several topics. In the following, we further elaborate on four of them.

Data source selection: This is the most challenging part when building data lake-aware edge learning pipelines. State-of-the-art solutions leverage LSH, while also focusing on feature engineering, consideration of both schema- and instance-level data, and advanced data transformations to reason about the relatedness of data sources.[38,39] The ultimate goal is to yield a list of combinable data sources. When the data sources are mapped to a relational schema, this goal is equivalent to detect joinable tables. Apart from the related data quality objectives, there are several performance-related optimizations that need to be taken into account given the high computation complexity of these tasks; such optimizations all target aggressive pruning of the search space. Combining these objectives, with data quality-related ones and human involvement, gives rise to optimization problems radically different than those encountered when considering task allocation. However, still, a promising approach is to aim to cast the whole problem as an integer linear programming one in a manner that no scalability problems are encountered, and enhance the initial solution with nature-inspired techniques,[40] something already shown to work very well in demanding geo-distributed, heterogeneous scenarios.[22,41]

Workflow expression: Declarative statement of service-level objectives (SLO) is not something new and is extensively employed in guiding elasticity in large-scale heterogeneous environments, e.g., Pusztai *et al.*'s[42] work. Such initiatives may serve as a basis to build more complete solutions that consider the full range of criteria and constraints involved, and also account for actions other than elasticity. More specifically, the elasticity actions need to be extended to allow not only the scaling and migration of tasks, but also the incorporation of additional data quality-oriented tasks and the reconfiguration of running tasks on the fly. This entails the development of novel schedulers and extensions to current state-of-the-art resource managers. It also implies more advanced optimization modules, which are discussed separately in the following.

Workflow optimization: Analyzing data closer to the edge devices, rather than in a central cloud, offers low latency, security, and scalability in many scenarios, such as smart cities. Several challenges arise when developing an edge computing-oriented analytics optimizer. Edge devices are highly heterogeneous in terms of resources, such as memory and computational capacity, and initial works that model such heterogeneity exist, e.g., Hiessl *et al.*'s[43] work. Furthermore, different cellular networks, as well as the emerging 5G technology, induce an additional challenge when combined.[44] Edge devices may also comprise smartphones and other mobile devices. This mobility needs to be taken into account when seeking an optimal service placement.[45] Overall, the corresponding service placement needs to be dynamic and adaptive to network and resource changes in real-time.[46] Finally, when dealing with sharing data across multiple edge devices, privacy constraints and restrictions may arise.[47] Thus, it becomes clear that multiple aspects should be considered when optimizing edge computing-oriented analytics. There is also a need to optimize multiple objectives at the same time or find a beneficial tradeoff between them. For example, response time, latency, energy consumption, and data transfer need to be minimized while resource utilization and QoS need to be maximized. Moreover, developing dynamic pricing models for service providers poses an additional challenge.[44] Independently optimizing workflows may be sufficient for scenarios with a small number of users; however, in edge computing applications, multiple users submit queries at the same time. Optimizing these queries simultaneously is complicated but would lead to more efficient resource utilization as techniques, such as service caching and resource sharing could be utilized.

Overall, the biggest challenges in the optimization of the workflows that we envisage stem from the combination of a much broader set of constraints and additional data quality-oriented objectives. Furthermore, the optimizations are not merely limited to judicious multiobjective task and data placement, configuration of parallelism degree, choice of the operator implementation, and so on. They should also cover modifications of the logical DAG execution plan, e.g., through inserting new data acquisition- and quality-specific operators. Also, modifying the type of tasks in the DAG based on the placement choices needs to be considered. For example, inference using complex deep networks could be allocated to either a set of edge nodes running different layers sequentially or to a single node, and thus, the workflow DAG is modified accordingly to reflect such decisions.

Workflow frameworks: In addition, the combination of large-scale data analytics frameworks, such as Apache Spark, Flink, and Storm, with edge learning frameworks, such as TensorFlow Federated[e] and Fate,[f] needs to be investigated in depth. Reinventing

[e] [Online]. Available: https://www.tensorflow.org/federated
[f] [Online]. Available: https://fate.fedai.org/

the wheel should be avoided, but it is unclear how this can be achieved in practice.

SUMMARY

Blending data acquisition, advanced ML, and analytics workflows to be executed over arbitrary heterogeneous, and geo-distributed computational resources both envisages and aspires to develop next-generation big data analytics and edge learning solutions. Current technologies need to be significantly extended in terms of the big data aspects directly considered, which in turn yields an updated list of optimization criteria, SLOs, and constraints. Data lake technologies, human intervention, and data quality guarantees become far more prevalent, while the underlying workflow execution engines need to be equipped with more advanced optimizers. Nevertheless, significant research efforts have already been conducted in several isolated aspects of the complete vision described hereby. Therefore, the technical roadmap is twofold: to both extend and judiciously combine existing solutions rather than starting from scratch, which is inefficient and unnecessary. To this end, we have identified the main research issues, and we sketched the current state of the art on top of which we advocate to build.

ACKNOWLEDGMENTS

The work of Anastasios Gounaris and Anna-Valentini Michailidou were supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.), Greece, through the First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment Grant under Project 1052.

REFERENCES

1. C. Doulkeridis and K. Nørvåg, "A survey of large-scale analytical query processing in MapReduce," *VLDB J.*, vol. 23, pp. 355–380, Dec. 2013, doi: 10.1007/s00778-013-0319-9.
2. S. Babu and H. Herodotou, "Massively parallel databases and MapReduce systems," *Found. Trends Databases*, vol. 5, pp. 1–104, 2013, doi: 10.1561/1900000036.
3. F. Li, B. C. Ooi, M. T. Özsu, and S. Wu, "Distributed data management using MapReduce," *ACM Comput. Surv.*, vol. 46, 2014, Art. no. 31, doi: 10.1145/2503009.
4. M. Hirzel, R. Soulé, S. Schneider, B. Gedik, and R. Grimm, "A catalog of stream processing optimizations," *ACM Comput. Surv.*, vol. 46, Mar. 2014, Art. no. 46, doi: 10.1145/2528412.
5. M. Fragkoulis, P. Carbone, V. Kalavri, and A. Katsifodimos, "A survey on the evolution of stream processing systems," 2020, *arXiv:2008.00842*, doi: 10.48550/arXiv.2008.00842.
6. A. Rheinländer, U. Leser, and G. Graefe, "Optimization of complex dataflows with user-defined functions," *ACM Comput. Surv.*, vol. 50, May 2018, Art. no. 38, doi: 10.1145/3078752.
7. G. Kougka, A. Gounaris, and A. Simitsis, "The many faces of data-centric workflow optimization: A survey," *Int. J. Data Sci. Analytics*, vol. 6, pp. 81–107, 2018, doi: 10.1007/s41060-018-0107-0.
8. S. M. F. Ali and R. Wrembel, "From conceptual design to performance optimization of ETL workflows: Current state of research and open problems," *VLDB J.*, vol. 26, pp. 777–801, Sep. 2017, doi: 10.1007/s00778-017-0477-2.
9. P. Jovanovic, O. Romero, and A. Abelló, "A unified view of data-intensive flows in business intelligence systems: A survey," in *Proc. Trans. Large-Scale Data-Knowl.-Centered Syst.*, 2016, pp. 66–107, doi: 10.1007/978-3-662-54037-4_3.
10. N. Makrynioti and V. Vassalos, "Declarative data analytics: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2392–2411, Jun. 2021, doi: 10.1109/TKDE.2019.2958084.
11. A. Modi *et al.*, "New query optimization techniques in the spark engine of azure synapse," *Proc. VLDB Endowment*, vol. 15, pp. 936–948, 2021, doi: 10.14778/3503585.3503601.
12. A. Davoudian and M. Liu, "Big data systems: A software engineering perspective," *ACM Comput. Surv.*, vol. 53, pp. 110:1–110:39, 2020, doi: 10.1145/3408314.
13. H. V. Olivera, G. RuiZhe, R. C. Huacarpuma, A. P. B. da Silva, A. M. Mariano, and M. Holanda, "Data modeling and NoSQL databases - A Systematic mapping review," *ACM Comput. Surv.*, vol. 54, pp. 116:1–116:26, 2021, doi: 10.1145/3457608.
14. Q.-C. To, J. Soto, and V. Markl, "A survey of state management in big data processing systems," *VLDB J.*, vol. 27, pp. 847–872, 2018, doi: 10.1007/s00778-018-0514-9.
15. H. Herodotou, Y. Chen, and J. Lu, "A survey on automatic parameter tuning for big data processing systems," *ACM Comput. Surv.*, vol. 53, 2020, Art. no. 43, doi: 10.1145/3381027.
16. I. A. T. Hashem *et al.*, "MapReduce scheduling algorithms: A review," *J. Supercomput.*, vol. 76, pp. 4915–4945, 2020, doi: 10.1007/s11227-018-2719-5.
17. X. Zeng *et al.*, "SLA management for big data analytical applications in clouds: A taxonomy study," *ACM Comput. Surv.*, vol. 53, pp. 46:1–46:40, 2020, doi: 10.1145/3383464.

18. S. Khalifa *et al.*, "The six pillars for building big data analytics ecosystems," *ACM Comput. Surv.*, vol. 49, 2016, Art. no. 33, doi: 10.1145/2963143.
19. C. C. Aggarwal *et al.*, "How can AI automate End-to-End data science?," 2019, *arXiv:1910.14436*, doi: 10.48550/arXiv.1910.14436.
20. S. Tang, B. He, C. Yu, Y. Li, and K. Li, "A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 71–91, Jan. 2022, doi: 10.1109/TKDE.2020.2975652.
21. S. Dolev, P. Florissi, E. Gudes, S. Sharma, and I. Singer, "A survey on geographically distributed big-data processing using MapReduce," *IEEE Trans. Big Data*, vol. 5, no. 1, pp. 60–80, Mar. 2019, doi: 10.1109/TBDATA.2017.2723473.
22. A.-V. Michailidou, A. Gounaris, M. Symeonides, and D. Trihinas, "EQUALITY: Quality-aware intensive analytics on the edge," *Inf. Syst.*, vol. 105, 2022, Art. no. 101953, doi: 10.1016/j.is.2021.101953.
23. M. Bansal, I. Chana, and S. Clarke, "A survey on IoT big data: Current status, 13 V's challenges, and future directions," *ACM Comput. Surv.*, vol. 53, 2021, Art. no. 131, doi: 10.1145/3419634.
24. J. Zhang *et al.*, "Edge learning: The enabling technology for distributed big data analytics in the edge," *ACM Comput. Surv.*, vol. 54, pp. 151:1–151:36, 2022, doi: 10.1145/3464419.
25. C.-J. Wu *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *Proc. 25th IEEE Int. Symp. High Perform. Comput. Archit., HPCA*, 2019, pp. 331–344, doi: 10.1109/HPCA.2019.00048.
26. P. Raith and S. Dustdar, "Edge intelligence as a service," in *Proc. IEEE Int. Conf. Serv. Comput.*, 2021, pp. 252–262, doi: 10.1109/SCC53864.2021.00038.
27. H. B. Pasandi and T. Nadeem, "CONVINCE: Collaborative cross-camera video analytics at the edge," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2020, pp. 1–5, doi: 10.1109/PerComWorkshops48775.2020.9156251.
28. A. W. Senior *et al.*, "Video analytics for retail," in *Proc. 4th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2007, pp. 423–428, doi: 10.1109/AVSS.2007.4425348.
29. A. Griva, C. Bardaki, K. Pramatari, and D. Papakyriakopoulos, "Retail business analytics: Customer visit segmentation using market basket data," *Expert Syst. Appl.*, vol. 100, pp. 1–16, Feb. 2018, doi: 10.1016/j.eswa.2018.01.029.
30. K. B. Subramanya and A. Somani, "Enhanced feature mining and classifier models to predict customer churn for an E-retailer," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng. - Confluence*, 2017, pp. 531–536, doi: 10.1109/CONFLUENCE.2017.7943208.
31. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1–10. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html
32. H.-J. Jeong, H.-J. Lee, C. H. Shin, and S.-M. Moon, "IONN: Incremental offloading of neural network computations from mobile devices to edge servers," in *Proc. ACM Symp. Cloud Comput.*, 2018, pp. 401–411, doi: 10.1145/3267809.3267828.
33. J. Cipar *et al.*, "Solving the straggler problem with bounded staleness," in *Proc. 14th Workshop Hot Topics Oper. Syst.*, 2013, Art. no. 22, doi: 10.5555/2490483.2490505.
34. J. Zhao, "Distributed deep learning under differential privacy with the teacher-student paradigm," in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 404–407.
35. J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3520–3532, doi: 10.5555/3294996.3295110.
36. S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.
37. D. Abadi *et al.*, "The Seattle report on database research," *SIGMOD Rec.*, vol. 48, pp. 44–53, 2019, doi: 10.1145/3385658.3385668.
38. A. Bogatu, A. A. A. Fernandes, N. W. Paton, and N. Konstantinou, "Dataset discovery in data lakes," in *Proc. 36th IEEE Int. Conf. Data Eng.*, 2020, pp. 709–720, doi: 10.1109/ICDE48307.2020.00067.
39. Y. Dong, K. Takeoka, C. Xiao, and M. Oyamada, "Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach," in *Proc. 37th IEEE Int. Conf. Data Eng.*, 2021, pp. 456–467, doi: 10.1109/ICDE51399.2021.00046.
40. J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*, 1st ed. 2011. [Online]. Available: https://github.com/clever-algorithms/CleverAlgorithms
41. M. Nardelli, V. Cardellini, V. Grassi, and F. L. Presti, "Efficient operator placement for distributed data stream processing applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1753–1767, Aug. 2019, doi: 10.1109/TPDS.2019.2896115.
42. T. W. Pusztai *et al.*, "SLO script: A novel language for implementing complex cloud-native elasticity-driven SLOs," in *Proc. IEEE Int. Conf. Web Serv.*, 2021, pp. 21–31, doi: 10.1109/ICWS53863.2021.00017.

43. T. Hiessl, V. Karagiannis, C. Hochreiner, S. Schulte, and M. Nardelli, "Optimal placement of stream processing operators in the fog," in *Proc. 3rd IEEE Int. Conf. Fog Edge Comput.*, 2019, pp. 1–10, doi: 10.1109/CFEC.2019.8733147.
44. W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, 2019, doi: 10.1016/j.future.2019.02.050.
45. N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.
46. Y. Teranishi, T. Kimata, H. Yamanaka, E. Kawai, and H. Harai, "Dynamic data flow processing in edge computing environments," in *Proc. 41st IEEE Annu. Comput. Softw. Appl. Conf.*, 2017, pp. 935–944, doi: 10.1109/COMPSAC.2017.113.
47. W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018, doi: 10.1109/ACCESS.2017.2778504.

ANASTASIOS GOUNARIS is an associate professor at the Department of Informatics, Aristotle University of Thessaloniki, 541 24, Greece. His main research interests include large-scale data management, massive parallelism, workflow and business process optimization, big data analytics, and data mining. Gounaris received his Ph.D. degree from the University of Manchester, Manchester, U.K. Contact him at http://datalab.csd.auth.gr/~gounaris/.

ANNA-VALENTINI MICHAILIDOU is a Ph.D. student with the Department of Informatics, Aristotle University of Thessaloniki, 541 24, Greece. Her research interests include distributed data analytics, dataflow and workflow optimization, data-quality, and edge computing. Michailidou received her B.Sc. degree in informatics from the Aristotle University of Thessaloniki. Contact her at http://annavalen.webpages.auth.gr/.

SCHAHRAM DUSTDAR is full professor of computer science heading the Research Division of Distributed Systems, TU Wien, 1040, Austria. He is an IEEE fellow. Contact him at dustdar@dsg.tuwien.ac.at.

EDITOR: Schahram Dustdar, dustdar@dsg.tuwien.ac.at

This article originally appeared in
IEEE Internet Computing
vol. 25, no. 6, 2021

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

Toward Sustainable Serverless Computing

Panos Patros, University of Waikato, 3240 Hamilton, Aotearoa New Zealand
Josef Spillner, Zurich University of Applied Sciences, 8400 Winterthur, Switzerland
Alessandro V. Papadopoulos, Mälardalen University, 722 20 Västerås, Sweden
Blesson Varghese, Queen's University Belfast, BT7 1NN Belfast, U.K.
Omer Rana, Cardiff University, CF10 3AT Cardiff, U.K.
Schahram Dustdar, TU Wien, 1040 Vienna, Austria

Although serverless computing generally involves executing short-lived "functions," the increasing migration to this computing paradigm requires careful consideration of energy and power requirements. Serverless computing is also viewed as an economically-driven computational approach, often influenced by the cost of computation, as users are charged for per-subsecond use of computational resources rather than the coarse-grained charging that is common with virtual machines and containers. To ensure that the startup times of serverless functions do not discourage their use, resource providers need to keep these functions hot, often by passing in synthetic data. We describe the real power consumption characteristics of serverless, based on execution traces reported in the literature, and describe potential strategies (some adopted from existing VM and container-based approaches) that can be used to reduce the energy overheads of serverless execution. Our analysis is, purposefully, biased toward the use of machine learning workloads because: 1) workloads are increasingly being used widely across different applications; and 2) functions that implement machine learning algorithms can range in complexity from long-running (deep learning) versus short-running (inference only), enabling us to consider serverless across a variety of possible execution behaviors. The general findings are easily translatable to other domains.

People and organizations are increasingly coming to terms with the urgent need to reverse the deleterious effects of climate change. The 2015 International Paris Agreement on Climate Change[a] mandated a temperature rise well below 2 °C—ideally capped at 1.5 °C. The UN proposed 17 Sustainable Development Goals (SDGs), such as "SDG7: Affordable and Clean Energy," "SDG9: Industry, Innovation, and Infrastructure," and "SDG13: Climate Action."[b] As our society's needs for computational power—and as such energy—increase, the software and computer engineering industries also need to decisively respond by adopting and encouraging sustainable operational paradigms. Serverless computing, as a new cloud computing paradigm, must also be made sustainable. As many predict serverless to be the next evolution of cloud systems,[1] ensuring that power and energy efficiency of such systems is adequately managed remains a crucial challenge.

[a] https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement

[b] https://sdgs.un.org/goals

1089-7801 © 2021 IEEE
Digital Object Identifier 10.1109/MIC.2021.3093105
Date of current version 6 December 2021.

Serverless computing expands on state-of-the-art cloud computing by further abstracting away software operations (ops) and parts of the hardware–software stack. One could consider functions, the execution unit of serverless computing, as "lightweight" containers, invoked with a set of inputs and expected to produce a set of outputs, when triggered. A key value proposition for serverless computing is its cost model, based on dynamic memory and CPU usage (connected directly to function invocations and as such, resource utilization and thus power/energy). This is unlike the more traditional cloud computing approaches, which charge based on the reservation of computing resources.

Data centers, and as such cloud and serverless computing, do have a significant impact on the world's total energy and power requirements. Estimates range from 200 to 500 TWh, which corresponds to 1–2.5% of the world's total energy usage. Additionally, this number is likely to increase as the demand for cloud computing increases: The estimated number of machines in data centers increased from 11 M in 2006 to 18 M in 2020. However, estimates are that only around 50% of this terawatt-hour energy consumption is used for actual computation; the other half is used on idling servers.[2] Serverless computing has a key role to play in this; this 50% waste in idling could in theory be completely reclaimed by this novel paradigm. Leading cloud providers have acknowledged the need to introduce real consumption pricing, something that becomes feasible with serverless architectures despite measurable overheads due to decomposition.[3] Serverless also provides a strong value proposition to users, who can pay for short time frames (less than a second), compared to reserving resources for an hour or more.

Apart from computation and memory, another energy-intensive computing task is networking. A 2015 metastudy estimated the Internet transmission energy to be 0.06 kWh/GB.[4] The problem is exacerbated in the era of Internet-of-Things (IoT) devices explosive growth: CISCO predicted 5ZB of IoT-related data to be transmitted in 2022.[5] This amount of IoT traffic will require 60 TWh of energy in 2022, essentially on par with 12–30% of data centers' energy needs.

A solution is to move small functions near the data instead of moving zettabytes of data to the data center; however, this adds significant extra development and operational burdens. Serverless computing, unlocks an easy migration toward easy-to-manage edge computing. Crucially, experimental evaluation on an IoT-driven video-analytics application suggests a 50% reduction in emissions is achievable if edge servers are used and data transmission to the data center is used sparingly.[6]

Therefore, to assure sustainable development for the Information Technology sector, there is an urgent call to establish energy- and power-aware design and operational strategies for the novel paradigm of serverless computing. We posit that the call for sustainable serverless computing can be split into three directions.

1. Sustainability techniques need to be designed and developed at the serverless platform level, such as power capping, scheduling, consolidation, and switching off policies. Crucially, to provide more room for maneuvering to the serverless Platform operator, serverless end-users need to be given incentives to minimize noncritical (deadline constrained) requirements, which can result in provisioning for sustainable service level agreements (SLAs).
2. The efficacy of serverless sustainability is closely coupled to workload patterns. The data center as a whole should avoid peak power consumption on its grid, as this leads to the use of emissions-heavy fossil-fuel-driven backup generators. As the world increasingly relies on artificial intelligence (AI) and machine learning (ML), the workload patterns generated by such smart systems must be studied and should (potentially) make use of relaxed SLAs, for instance during the training phase.
3. Connecting the above two topics, how can we, indeed, know how successful a serverless sustainability technique is? Sustainability oriented serverless benchmarks are needed to assess the quality of the proposed techniques, and these benchmarks need to be designed with realistic contemporary workloads, such as AI/ML, in mind. Crucially, as computation needs to move closer to the data, monolithic AI applications need to be replaced with function-oriented microservice architectures such that they can fit on low-powered edge devices and serverless operators can leverage the various aforementioned sustainability techniques.

Figure 1 illustrates data sources that *feed* data streams into serverless functions. Functions are frequently invoked with stream chunks as input, receiving data across different types of communication channels. A single data source, e.g., in-built environments, Industry 4.0, and electric mobility, may utilize different types of communication infrastructure.

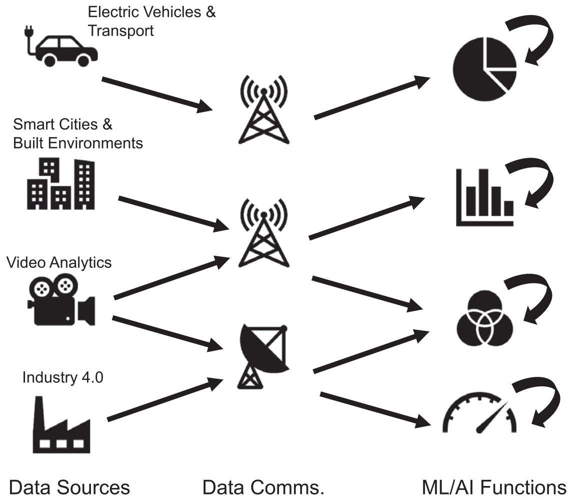


FIGURE 1. Serverless functions—responding to incoming data streams.

DESIGNING SUSTAINABLE SERVERLESS PLATFORMS

Various approaches can be used to limit the power consumption of serverless functions, ensuring more efficient use of energy of the associated infrastructure on which these functions are hosted. These techniques, which can be invoked transparently (from an end-user perspective) must be implemented by the serverless platform and can include: power capping of serverless deployments, use of scheduling strategies to make more effective use of the physical resources on which serverless functions are hosted, and mechanisms to minimize cold start times that can have significant power consumption requirements. Each of these approaches is described in this section, along with their benefit (and limitations).

Power Capping: This approach relies on limiting the power consumed by functions hosted within a specific container environment. Power capping techniques such as dynamic voltage and frequency scaling (DVFS) and running average power limit (RAPL) are hardware-based approaches that reduce CPU frequency and voltage to lower processor power consumption. However, this degrades the entire system performance and consequently the deployed application. Power cap violations are undesirable and need to be effectively managed, as the power benefit can be counterproductive—leading to applications running for longer time periods, which at times is the worst possible outcome from a sustainability perspective as it prevents shutting down under-utilized machines. More specifically, power cap violations occur when the total power consumed by a server exceeds a threshold defined by the server administrators.

Two power capping techniques are particularly relevant in the context of container-based functions: 1) DockerCap for Docker containers can make use of system power consumption obtained from a hardware power meter and RAPL. The CPU quota of all containers at different scheduling priority is reduced, thereby affecting the performance of all containers; and 2) DEEP-mon power monitoring can be used for Docker containers on the Kubernetes platform. This technique relies on RAPL and DVFS to manage power cap limits. It is demonstrated that RAPL affects the run-time performance of all containers on a server. RAPL enforces a power cap on the processor and DRAM by reducing the CPU frequency and thus degrading the overall system performance.

In the context of language-runtime-based functions, such as those supported by serverless platforms such as funcX, which runs on Python,[c] more fine-grain power capping can take place. Such algorithms could target specific subcomponents that might not need to run at full speed, such as resource-intensive dynamic memory management, aka., garbage collection.[7] Alternatively, the language runtime might be able to better characterize the resource requirements of its functions, enabling improved execution density via adaptive resource sharing among multitenant functions.[8]

The performance and execution behavior of a function is influenced by the power consumed by each function. Longer running functions can be terminated, for instance, if their power consumption exceeds the prespecified cap.

Network Power Saving: QUIC employs some of the basic mechanisms of TCP and TLS while keeping UDP as its underlying transport layer protocol. QUIC is, therefore, a combination of transport and security protocols by performing tasks including encryption, packet reordering, and retransmission. QUIC can be considered a user space, UDP-based (stream-oriented) protocol developed by Google—published by IETF in May 2021 as RFC9000. It is estimated that approx. 7% of Internet traffic employs QUIC. This protocol offers all the functionalities required to be considered a connection-oriented transport protocol, overcoming numerous problems faced by other connection-oriented protocols, such as TCP and SCTP. Specifically, the addressed problems are reducing the

[c] https://funcx.org/

connection setup overhead, supporting multiplexing, removing the head-offline blocking, supporting connection migration, and eliminating TCP half-open connections.

QUIC executes a cryptographic handshake that reduces connection establishment overhead by employing known server credentials learned from past connections. In addition, QUIC reduces transport layer overhead by multiplexing several connections into a single-connection pipeline. Furthermore, as QUIC uses UDP, it does not maintain connection status information in the transport layer. This protocol also eradicates the head-of-line blocking delays by applying a lightweight data-structure abstraction called *streams*. Due to its lightweight nature and support for data encryption, it is viewed as an important enabler for serverless functions. Using reduced overheads, the power consumption of QUIC is also reduced compared to other equivalent network protocols used for serverless deployment. The QUIC protocol can be also be used to preserve energy resources, especially between sleep and awake states that are often used by IoT devices. Maintaining a TCP connection requires use of keep-alive packets, which can consume energy and bandwidth. Understanding how this can undertake more efficiently is also an important approach to reduce energy use.[9]

Hotspot and Coldspot Migration: A common approach to reducing power consumption is the dynamic consolidation of virtual machines and containers on a smaller number of physical machines (PMs). This is based on the observation that PMs run at 10–50% of their maximum CPU usage and the majority of PMs are idle while still consuming about 70% of their peak power. This process involves migrating workload to enhance resource usage and minimize the use of machines that are underutilized within a data center—often turning these PMs OFF so that they do not consume power. Migration is expected to be transparent and beneficial when a physical server is highly overloaded (creating a hotspot) or underloaded (creating a coldspot). However, consolidation policies reduce energy consumption significantly but live VM migration results in increased violations of SLAs.

Many of these techniques, however, suffer from issues of instability and fluctuation—as migration of workload is often based on an instantaneous (or time-window-based) workload analysis. Only recently, time-series (machine-learning-based) forecasting techniques that take account of multiple criteria for estimating workloads are being used. Understanding where *cold spots* are likely to happen is as important as identifying the location of over utilized resources within a data center. A key challenge that differentiates this challenge within a serverless environment is the overhead of migrating workloads compared to: 1) the function execution time; and 2) the migration time and associated startup time of the function at the new location. Both of these aspects limit the benefit of migration for short-running functions—compared to longer running VMs or containers.

Power-OFF Strategies: As mentioned, traditional approaches in data center consolidation have focused on migrating long-running virtual machine instances to eventually power down idle hosts. More recently, these approaches have been suggested for reapplication in cloud-to-fog continuums.[10] In our view, such continuums will emerge everywhere due to the proliferation of sensing, and it would be short-sighted to assume conventional virtual machines as an execution technology. Instead, with a serverless computing approach, there are several advantages to simplify management and increase efficacy. First, short-running code can be left alone, and hosts can be switched OFF or suspended when none or even few instances remain. This greatly increases flexibility to decide when a switch-OFF shall occur. Second, the inherent event-driven nature of function invocation allows coupling with dynamic resumption such as Wake-on-LAN, in particular with fast-resuming and fast-booting technologies such as Coreboot[11] in conjunction with delay-tolerant function invocations. This way, hardware sensors along with virtualized fog nodes can be connected to as if they were permanently running, and yet they can power OFF in between. This programming simplicity resonates with the serverless computing mindset that infrastructural concerns are abstracted and largely hidden from application engineers.

Wake-on-LAN concepts have already reached beyond LANs and are commonly used in Internet-wide device management, including with custom protocols such as Apple's Bonjour Sleep Proxy (Multicast DNS, RFC 6762). For messaging-based triggers, protocol wrapping will allow a device to be booted or resumed before answering a request. For time-based triggers, an external time source needs to be added. Figure 2 shows the sequence of events, including eventual suspend and resume actions by the device or virtualized runtimes, based on rules or machine-learned patterns.

According to our early work experiments on event-driven power switching of a FaaS platform triggered by occasional IoT events, this approach added on average 0.95s execution time per request, within the delay tolerance to most batch jobs. In return, the platform could be suspended for 73% of the time, leading

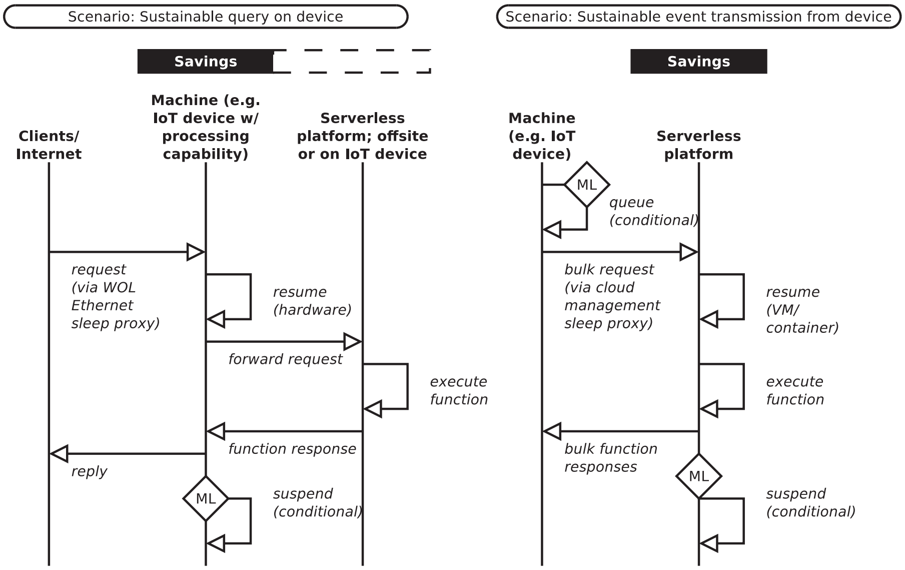

Scenario: Sustainable query on device
Savings
Clients/ Internet
Machine (e.g. IoT device w/ processing capability)
Serverless platform; offsite or on IoT device
request (via WOL Ethernet sleep proxy)
resume (hardware)
forward request
execute function
function response
reply
ML
suspend (conditional)
Scenario: Sustainable event transmission from device
Savings
Machine (e.g. IoT device)
Serverless platform
ML
queue (conditional)
bulk request (via cloud management sleep proxy)
resume (VM/ container)
execute function
bulk function responses
ML
suspend (conditional)

FIGURE 2. Sustainability approaches for Internet-wide control of device states and virtualized runtime lifecycle based on serverless event processing.

to great savings in power consumption. Figure 3 summarizes the suspend/resume pattern over a window of 6 min. The research challenge is, then, to learn and predict messaging patterns to optimize the suspend/resume or switch-OFF/switch-ON actions.

EFFECT OF WORKLOAD PATTERNS

We consider machine learning workloads consisting of deep neural networks (DNN), which comprise a sequence of layers and is a general term that covers all neural networks with multiple hidden layers (that is multiple layers between the input and output layers). Connectivity between the layers and propagation of an error function differentiates the different types of DNN architectures. Overall, a DNN may include: 1) fully connected layers, where each node in the network is connected to nodes at layer+1 and layer-1. A DNN may also include nodes that are not fully connected, or where connections may skip layers; 2) convolution layers convolve the input to produce feature maps of inputs to learn features. This is generally undertaken by identifying X-by-Y windows that are moved over a stride of Z. A convolution filter is chosen to identify key properties observed within the input dataset; 3) pooling layers apply a predefined function (maximum or average) to downsample the input; 4) an activation layer applies nonlinear functions and the most commonly used is the rectified linear unit (ReLu); and 5) a Softmax layer is generally used for classification to generate a probability distribution over the possible classes. The complexity of the DNN model is dependent on the number of nodes, the interconnectivity structure, and the choice of hyperparameters (such as X, Y, Z for convolution layers and learning rate).

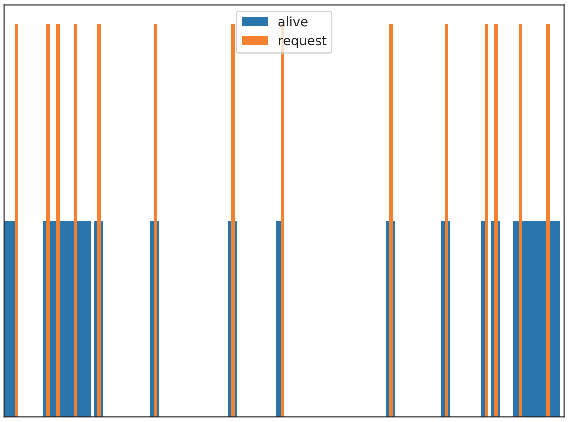


FIGURE 3. Event-driven suspend/resume patterns leading to power consumption savings.

Two different ML deployment scenarios can be considered: 1) workloads that are distributed and 2) workloads that are *miniaturization*. Traditionally, a DNN is trained at a data center and, then, deployed as a monolithic application on other resources where they need to be trained. More recently, it has been demonstrated that DNNs can be partitioned and deployed across different tiers of resources spanning the cloud, edge, and user devices to preserve privacy and achieve performance and energy efficiency in distributed systems.[12,13,14] In this scenario, the layers of a partitioned DNN can be mapped onto serverless functions that are invoked on-demand for inference on both resource-abundant (data center) and resource-constrained (edge servers or user devices) tiers. Such an approach can meet the power cap requirements on different resources. Since training is typically a long-running task, traditional mechanisms such as containers or VMs may be suited for deployment.

In the second scenario, machine learning workloads that need to fit on resource-constrained resources that are located outside conventional data centers closer to where data are generated are considered. The energy consumed by both the networking and compute infrastructure can be reduced. During inference, the energy consumption of the networking infrastructure is naturally conserved if limited data are transferred to geographically distant servers and can be processed at the edge of the network (up to a 50% reduction in the carbon footprint when processing data at the edge has been demonstrated[15]). In addition, there are opportunities to reduce energy consumption on a compute resource.

Consider the example of a real-time video analytics application, such as identifying objects on different frames of a video stream. A different DNN model from a portfolio of models can be employed for each frame to maximize the accuracy of detection.[16] This is achieved by leveraging the metacharacteristics of each video frame, such as the size of the object and the speed of movement of the object. Certain DNN models are more accurate when detecting fast moving objects but may have higher power requirements. Conversely, low-power models may deliver sufficient accuracy for slow-moving objects. Since the models contained in the portfolio have different power requirements, serverless computing can leverage the accuracy and power tradeoff to deliver a transprecision-based approach that maximizes accuracy.

QUALITY ASSESSMENT OF SERVERLESS SUSTAINABILITY

While the sustainability aspect of serverless computing has gained a lot of attention, the same cannot be said about approaches and methodologies for the quality assessment of serverless sustainability. Kistowski *et al.* define a benchmark as a "Standard tool for the competitive evaluation and comparison of competing systems or components according to specific characteristics, such as performance, dependability, or security."[17] The SPEC Cloud IaaS 2018 benchmark,[d] for example, focuses on four key benchmark metrics: 1) replicated application instances, 2) performance score, 3) relative scalability, and 4) mean instance provisioning time, none of which includes sustainable-related metrics. This is the typical focus of most of benchmarks in cloud computing,[18] and the ones developed for serverless computing.[19,20]

Including sustainability in benchmarks for serverless computing is challenging, yet extremely important, especially when considering the fast growth of AI applications deployed in the cloud. In a study conducted by Strubell *et al.*,[21] it has been found that *training* a single deep learning model can generate up to 284,000 kg of CO_2 emissions. This corresponds to the total lifetime carbon footprint of approximately five cars. But this is not a one-off cost, concluding with the training of the ML algorithm—that could be potentially mitigated using transfer learning techniques. Amazon estimates that 90% of production ML infrastructure costs are for *inference*, not training.[22] NVIDIA estimated that 80%–90% of the energy cost of neural networks deployed in data centers lie in inference processing.[e]

In addition, a benchmark should not just focus on the raw numbers of energy consumption, but rather on where the energy comes from. If an AI model were trained using electricity generated primarily from renewables, its carbon footprint would be correspondingly lower. For instance, Google Cloud Platform's power mix is more heavily weighted toward renewables than the AWS Platform (56% versus 17%, according to Strubell *et al.*[21]). Lacoste *et al.*[23] developed an ML CO_2 calculator[f] that practitioners can use to estimate the carbon footprints of their deployment based on the following factors: 1) hardware type, 2) hours used, 3) cloud provider, and 4) geographical region. The last factor can have a significant impact on carbon emission, as different locations may have different access to greener energy sources.

Most of these efforts focus on long-lived applications that may not fully exploit the potential of serverless computing. Researchers and practitioners can try to focus on how to optimize their deployments and executions of ML applications. However, more fundamental long-term solutions are needed to automate and optimize the sustainability of ML applications. This could be achieved through the main features of serverless computing and the development of suitable

[d] https://www.spec.org/cloud_iaas2018/
[e] https://www.forbes.com/sites/moorinsights/2019/05/09/google-cloud-doubles-down-on-nvidia-gpus-for-inference/
[f] https://mlco2.github.io/impact/

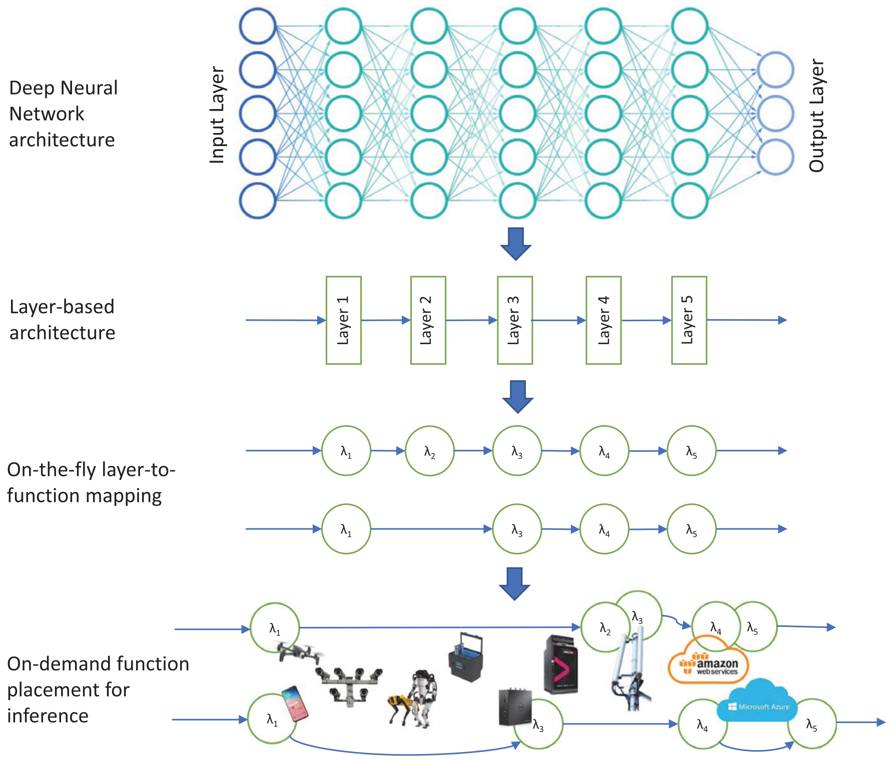


FIGURE 4. Serverless approach for the sustainable execution of deep AI inference.

management techniques and cost models that can promote greener computation.

In Figure 4, we display our proposed approach for enabling serverless computing for AI-intensive workloads. As major energy requirements for AI workloads are due to inference (i.e., during usage rather than training—where training is only needed occasionally or once), we also focus our attention on the actual operation of deep learning algorithms versus their training. A key observation is that DNNs do not need to run as monolithic structures; instead, we propose that each layer of a DNN be segmented into a function suitable for fine-grain deployment and scheduling—which can achieve improved computational density both on cloud and on edge servers. As such, consider for simplicity a multilayer DNN.[24] Each layer will have outputs $u \in R^m$, weights $W \in R^{m \times n}$, biases $b \in R^m$, activation function g, and connected with inputs $u \in R^n$ will execute the following function:

$$\lambda^{\text{DNN}} := y = g(W \cdot u + b).$$

Thus, a DNN with depth k can be recursively split into independent functions that can be *stacked* as follows, considering that λ_0^{DNN} describes the raw data inputs to whole DNN:

$$\lambda_k^{\text{DNN}} := \gamma = g(W \cdot \lambda_{k-1}^{\text{DNN}} + b).$$

Consequently, from a serverless perspective, the instructions required to be transmitted to execute one of these λ^{DNN} functions reduces to the weights, biases, and type of activation function. From a cluster management perspective, the serverless provider can now make more informed decisions on where and when each λ^{DNN} instance should run, taking into consideration data locality to minimize network energy, renewable, and off-peak power availability to reduce the stress on the grid, depending on their sparsity, place them on machines with the appropriate level of hardware parallelism (e.g., CPU versus GPU), as well as existing utilization of computing resources to maximize utilization of power-on resources while keeping as many machines as possible powered OFF. Additionally, smart reusable algorithms could be created that easily combine existing structures to efficiently deploy novel DNN architectures by essentially leveraging this microservice-oriented design of DNNs.

Furthermore, on the actual device that executes one of these λ^{DNN} functions, an autonomic manager operated by the serverless platform can enable runtime-specific energy-aware optimizations. For example, consider multitenancy of DNN lambdas, i.e., the secure sharing of equal λ^{DNN} functions used by multiple tenants—the statelessness of these functions allows for this optimization—which can save energy costs by reducing unneeded context switches or thrashing the hardware caches. Additionally, if users are willing to sacrifice a bit of accuracy for improved energy efficiency, even slightly different λ^{DNN} could be shared, for example, by using the average weights of the multitenant users or one user accepting to use λ^{DNN} of another.

CONCLUSIONS

The serverless computing paradigm enables abstracting away hardware resource management and resource operations, which transfers the burden of energy innovation to the serverless platform provider. With an urgent call for worldwide sustainable development, serverless platforms must also be designed to be energy- and power-aware.

We highlight the need for sustainable serverless computing, which we posit can be achieved via: 1) serverless platform design and infrastructure, 2) improved characterization of novel IoT- and AI-driven workloads, which are bound to dominate the world's computing needs, paired with smarter decision-making at the application-design level, and 3) automated methodologies that assess the sustainability efficacy of such power and energy-aware methods.

Finally, people, developers and end-users must also contribute to this effort of sustainable serverless computing! For instance, a user might need to consider turning ON the "eco-mode" for their functions, relaxing the requirements just enough so that the serverless provider has enough time to schedule them during an off-peak time or can keep that extra server in reserve turned OFF. "Human brains can do amazing things with little power consumption. The bigger question is how can we build such machines."[25]

ACKNOWLEDGMENTS

Note: This contribution is based on discussions initiated at a Daghstuhl event on "Serverless Computing" (16–21 May 2021).

Details of the event can be found at: *https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=21201*

REFERENCES

1. J. Schleier-Smith *et al.*, "What serverless computing is and should become: The next phase of cloud computing," *Commun. ACM*, vol. 64, no. 5, pp. 76–84, 2021.
2. D. Mytton, "How much energy do data centers use?" Hypertext document, 2020. [Online]. Available: https://davidmytton.blog/howmuch-energy-do-data-centers-use/
3. A. Poth, N. Schubert, and A. Riel, "Sustainability efficiency challenges of modern IT architectures—A quality model for serverless energy footprint," in *Systems, Software and Services Process Improvement*, M. Yilmaz, J. Niemann, P. Clarke, and R. Messnarz, Eds. Cham, Switzerland: Springer, 2020, pp. 289–301.
4. J. Aslan, K. Mayers, J. G. Koomey, and C. France, "Electricity intensity of internet data transmission: Untangling the estimates," *J. Ind. Ecol.*, vol. 22, no 4, pp. 785–798, 2018.
5. Cisco, Cisco annual internet report (2018–2023), San Jose, CA, USA, White paper, 2020.
6. B. Ramprasad, A. da Silva Veith, M. Gabel, and E. de Lara, "Sustainable computing on the edge: A system dynamics perspective," in *Proc. 22nd Int. Workshop Mobile Comput. Syst. Appl.*, 2021, pp. 64–70.
7. P. Patros, K. B Kent, and M. Dawson, "Mitigating garbage collection interference on containerized clouds," in *Proc. IEEE 12th Int. Conf. Self-Adaptive Self-Organizing Syst.*, 2018, pp. 168–173.
8. V. Podolskiy, M. Mayo, A. Koay, M. Gerndt, and P. Patros, "Maintaining SLOs of cloud-native applications via selfadaptive resource sharing," in *Proc. IEEE 13th Int. Conf. Self-Adaptive Self-Organizing Syst.*, 2019, pp. 72–81.
9. P. Kumar and B. Dezfouli, "Implementation and analysis of QUIC for MQTT," *Computer Netw.*, vol. 150, pp. 28–45, 2019.
10. O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.
11. J. Sun, M. Jones, S. Reinauer, and V. Zimmer, "Building coreboot with Intel FSP," in *Embedded Firmware Solutions*. Berkeley, CA, USA: Apress, 2015, pp. 55–95.
12. L. Lockhart, P. Harvey, P. Imai, P. Willis, and B. Varghese, "Scission: Performance-driven and context-aware cloud-edge distribution of deep neural networks," in *Proc. IEEE/ACM 13th Int. Conf. Utility Cloud Comput.*, 2020, pp. 257–268.
13. H. Ahn, M. Lee, C.-Ho Hong, and B. Varghese, "Scissionlite: Accelerating distributed deep neural networks using transfer layer," 2021, *arXiv:2105.02019*.
14. Y. Kang *et al.*, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. 22nd Int. Conf. Architectural Support Program. Lang. Oper. Syst.*, 2017, pp. 615–629.

15. B. Ramprasad, A. da Silva Veith, M. Gabel, and E. de Lara, "Sustainable computing on the edge: A system dynamics perspective," in *Proc. 22nd Int. Workshop Mobile Comput. Syst. Appl.*, 2021, pp. 64–70.
16. J. K. Lee, B. Varghese, R. Woods, and H. Vandierendonck, "TOD: Transprecise object detection to maximise real-time accuracy on the edge," in *Proc. 5th IEEE Int. Conf. Fog Edge Comput.*, 2021.
17. J. von Kistowski, J. A. Arnold, K. Huppler, K.-D. Lange, J. L. Henning, and P. Cao, "How to build a benchmark," in *Proc. 6th ACM/SPEC Int. Conf. Perform. Eng.*, 2015, pp. 333–336.
18. A. V. Papadopoulos *et al.*, "Methodological principles for reproducible performance evaluation in cloud computing," *IEEE Trans. Softw. Eng.*, to be published.
19. E. van Eyk, J. Scheuner, S. Eismann, C. L. Abad, and A. Iosup, "Microbenchmarks: The SPEC-RG vision for a comprehensive serverless benchmark," in *Proc. Companion ACM/SPEC Int. Conf. Perform. Eng.*, 2020, pp. 26–31.
20. M. Copik, G. Kwasniewski, M. Besta, M. Podstawski, and T. Hoefler, "SeBS: A serverless benchmark suite for function-as-a-service computing," 2020, *arXiv:2012.14132*.
21. E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 9, pp. 13693–13696, Apr. 2020.
22. A. Jassy, Amazon AWS ReInvent keynote, 2018. [Online]. Available: https://www.youtube.com/watch?v=ZOIkOnW640A
23. A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, Quantifying the carbon emissions of machine learning," 2019, *arXiv:1910.09700*.
24. A. Burkov, *The Hundred-Page Machine Learning Book*, volume 1. Québec, QC, Canada: Andriy Burkov, 2019.
25. K. Hao, "Training a single AI model can emit as much carbon as five cars in their lifetimes," MIT Technol. Rev., 2019.

PANOS PATROS is currently a Senior Lecturer with the Department of Software Engineering, University of Waikato, Aotearoa, New Zealand. He leads the Cloud and Adaptive Systems (Ohu Rangahau Kapua Aunoa) ORKA Lab and is a member of the Ahuora Smart Energy Systems Centre. He received the Ph.D. degree in computer science from the University of New Brunswick, Fredericton, NB, Canada, on multitenancy, performance, and modeling of cloud systems. Contact him at panos.patros@waikato.ac.nz.

JOSEF SPILLNER is currently an Associate Professor with the Zurich University of Applied Sciences, Winterthur, Switzerland, currently investigating distributed application computing paradigms and nation-scale system designs. He is a Senior Member of IEEE. Contact him at josef.spillner@zhaw.ch.

ALESSANDRO V. PAPADOPOULOS is currently an Associate Professor in Computer Science with Mälardalen University, Västerås, Sweden. His research interests include real-time and embedded systems, distributed systems, self-adaptive systems, and control theory. He received the Ph.D. degree in information technology (systems and control) from Politecnico di Milano, Milan, Italy. He is a Senior Member of IEEE. Contact him at alessandro.papadopoulos@mdh.se.

BLESSON VARGHESE is currently an Associate Professor in Computer Science with Queen's University Belfast, Belfast, U.K. His current research interests include distributed systems and machine learning spanning across the cloud and edge. He received the Ph.D. degree in computer science from the University of Reading, Reading, U.K. Contact him at b.varghese@qub.ac.uk.

OMER RANA is currently a Professor of Performance Engineering with Cardiff University, Cardiff, U.K., with research interests in distributed systems (cloud/edge and IoT) and data analytics (machine learning). He received the Ph.D. degree in neural computing and parallel architectures from Imperial College, London, U.K. Contact him at ranaof@cardiff.ac.uk.

SCHAHRAM DUSTDAR is currently a Full Professor of Computer Science (Informatics) with a focus on Internet Technologies heading the Distributed Systems Group, TU Wien, Vienna, Austria. He is a member of the Academia Europaea: The Academy of Europe. He is a Fellow of the IEEE. He is the corresponding author of this article. Contact him at dustdar@dsg.tuwien.ac.at.

EDITOR: Theresa-Marie Rhyne, theresamarierhyne@gmail.com

This article originally appeared in IEEE Computer Graphics and Applications vol. 43, no. 2, 2023

DEPARTMENT: VISUALIZATION VIEWPOINTS

The Flow of Trust: A Visualization Framework to Externalize, Explore, and Explain Trust in ML Applications

Stef van den Elzen, *Eindhoven University of Technology, 5612, Eindhoven, The Netherlands*

Gennady Andrienko and Natalia Andrienko, *Fraunhofer Institute IAIS, 57357, Sankt Augustin, Germany*

Brian D. Fisher, *Simon Fraser University, Burnaby, BC, V5A 1S6, Canada*

Rafael M. Martins, *Linnaeus University, 352 52, Vaxjo, Sweden*

Jaakko Peltonen, *Tampere University, 33100, Tampere, Finland*

Alexandru C. Telea, *Utrecht University, 3584, Utrecht, The Netherlands*

Michel Verleysen, *University of Louvain, 1348, Ottignies-Louvain-la-Neuve, Belgium*

We present a conceptual framework for the development of visual interactive techniques to formalize and externalize trust in machine learning (ML) workflows. Currently, trust in ML applications is an implicit process that takes place in the user's mind. As such, there is no method of feedback or communication of trust that can be acted upon. Our framework will be instrumental in developing interactive visualization approaches that will help users to efficiently and effectively build and communicate trust in ways that fit each of the ML process stages. We formulate several research questions and directions that include: 1) a typology/taxonomy of trust objects, trust issues, and possible reasons for (mis) trust; 2) formalisms to represent trust in machine-readable form; 3) means by which users can express their state of trust by interacting with a computer system (e.g., text, drawing, marking); 4) ways in which a system can facilitate users' expression and communication of the state of trust; and 5) creation of visual interactive techniques for representation and exploration of trust over all stages of an ML pipeline.

The last two decades have been marked by the explosion of *data* sources ranging over virtually all application types, such as multimedia collections (images, text, sound, videos), data tables from databases having increasing diversity and size, and measurements from the physical world, such as GPS and trajectory data. As the size, diversity, and complexity of the data increased, so did the awareness that higher level *information* can be extracted from these sources. A particularly successful manner to infer such information from raw data is proposed by machine learning (ML). ML applications construct *models* of the phenomena from which data are acquired and aim to generate predictions related to these phenomena in the presence of new, unseen, data. ML applications covering classification and prediction are increasingly present in diverse contexts of decision support and task automation by generating outputs relevant to a human user in the given context.

0272-1716 © 2023 IEEE
Digital Object Identifier 10.1109/MCG.2023.3237286
Date of current version 23 March 2023.

As ML models become increasingly powerful, so does their engineering and inherent complexity. As such, an increasingly important research direction targets explainable AI (XAI) (i.e., the creation of methods and tools that shed light on the functioning of such models to their various users). However, while such techniques help users to understand how a model is structured and works, they currently do not directly cover building *trust* in the model (and/or the process leading to it). We consider XAI and trust to be loosely related but independent topics. Providing explanations may help to increase trust, but not necessarily: even if a system provides a perfect explanation of how its model works, the user may still not trust the system, due to, e.g., wrong model decisions. The reverse also holds: although XAI might show a model's flaws, users might still have high trust in the system, due to, e.g., faith in the authority or organization behind it, or because they simply lack (domain) knowledge to understand the explanation. As such, in current systems trust is typically represented implicitly, lacking, e.g., explicit interaction and support feedback mechanisms. In this article, we argue that trust (or the lack thereof) in ML applications is an aspect as important as—if not more important than—understanding the operation of such applications.

Currently, visual analytics (VA) and ML applications lack an interface for expressing trust and/or distrust. What is missing from current interfaces is both 1) ways for the user to express and explain (dis)trust, and 2) ways to capture and manage such (dis)trust in an explicit manner such that it can directly affect the visual interactive ML process. We believe that in complex systems, *expressing trust* (beyond a superficial overall level of trust) requires exploratory, interactive visualization support to discover the areas of trust and distrust along with their reasons.

As a first step, to create awareness, and to work toward treating trust as a first-class citizen in designing and reasoning about VA applications that use ML, we introduce a conceptual framework that captures the flow of trust. This framework lays a foundation for externalization, exploration, and explanation of trust using interactive visualization techniques during development of ML and VA applications and helps with post hoc analysis of existing systems. The framework guides researchers and tool creators in making trust explicit by considering different trust elements: 1) content—what needs to be captured and explicitly represented; 2) target form of the content; 3) communication media (e.g., text, drawing, marking); 4) facilitation (e.g., prompting, templates); and 5) visualization techniques. Our contributions are:

- a conceptual framework that enhances the ML pipeline with a model that captures the flow of trust, and,
- guides the construction of VA solutions that support and explicitly manage trust development;
- the application of our framework to examples of current ML models extended with interactive visualization support for evolution of trust;
- identification and discussion of research directions concerning trust.

MOTIVATING EXAMPLE

To corroborate the need for a framework for externalizing, exploring, and explaining trust and to illustrate the presentation of the framework, we introduce a real-world example. It involves our experiences gained during the creation and usability testing of an optimization model for flight scheduling.

Domain Problem

The airspace (particularly, in Europe) is divided into compartments, called sectors, within which the traffic is supervised by air traffic controllers. The sectors have limited capacities defined as the maximal safely manageable number of flights that can cross a sector in one hour. Flights are conducted according to plans. Initial flight plans are prepared by airlines intending to conduct the flights. It often happens that the demand for a sector (i.e., the number of flights that need to cross it within an hour) exceeds the sector capacity and thus creates a so-called hotspot. For safety reasons, it is necessary to eliminate the hotspots by modifying parts of the flight plans. The most common modification is delaying a flight. It is sometimes possible to modify flight routes so that overloaded sectors are avoided while the route lengths do not increase significantly. The task of an optimization model is to create a daily flight schedule such that no hotspots will emerge. The input data consist of a set of initial flight plans; the output is a set of final flight plans.[2]

Solution Development

The model for solving the problem was built using historical data D for a large region of Europe and a time span of one year. For each day, there were sets of initial and final flight plans. A flight plan in D has the form of a trajectory consisting of geographic positions (waypoints) and time stamps. This format was not suitable for model development. The model developers (MD) defined a set of features (i.e., numeric attributes) derivable from the original data and suitable for model building and thus transformed D to D'. Later on, it turned out that the derived features were not easily understandable to the domain users (DU). Also, the selection of these particular features was not properly justified.

MD built the model M by means of a reinforcement learning algorithm. The flights were modeled as agents taking decisions to delay for X minutes. Later on, this approach to modeling was questioned as the behavior of the resulting model did not match users' way of reasoning. Assuming that reinforcement learning was the right method to create a model, a better idea might be to model sectors as agents.

The built model M (a neural network) was not inherently explainable; therefore, MD created a surrogate model M' to explain the behavior of M. M' was a combination of decision trees with a depth up to 35 levels. The amount of information was far beyond the human capability to comprehend it. Although visualization developers (VD) invented some tricks to present M' in a simplified and aggregated form, it was not enough for a good understanding of the model behavior.

The execution of M is an iterative process of modifying an original flight schedule. Each step results in a version of the flight schedule that differs from the previous one in terms of flight delays and sector loads. VD created a visualization that presented an overview of the process with summarized changes from step to step and allowed us to explore the details and compare different versions of the schedule. The visualization showed how hotspots were resolved at the cost of flight delays. At the overall level, the delays appeared to be justified; still, DU were not convinced that the delays were not longer than necessary, and there was no good way to check this. At the detailed level, DU questioned the choice of the flights to be delayed. Although XAI methods were used, and the explanations could be explored, trust in the model was still low.

The output of M was viewed and explored by means of a visualization showing the final flight schedule and enabling its comparison with the original schedule. M' was used for providing explanations for modifications of a particular user selected flight plan. The explanations were presented with decision rules. DU found them unsatisfactory: excessively long, hard to understand due to complicated nonintuitive features, and failing to explain the choice of the flights to be delayed. DU concluded that they are not convinced that the model operates properly and thus cannot adopt such a model for use in practice.

This project provided a number of lessons concerning possible trust issues along the process of model development and use. In brief, the MD put too high trust in the chosen modeling method and in the capability of a surrogate model to explain the logic of the trained model. DU, in turn, did not trust the model as a whole due to a lack of understanding of its behavior, and they did not trust the proposed solutions due to a lack of evidence of the solutions being optimal.

RELATED WORK

The importance of users' trust in ML and the ways in which visualizations affect it have been discussed and summarized in a few survey papers in recent years. For instance, Endert et al.[13] identified enhancing trust and interpretability as one of the open challenges and opportunities for ML and VA. According to the authors, analysts can build mental models of how ML models work via interactive visualization, which will increase trust. This happens in two different levels of cognition: a *qualitative* level, where the most important goal is to communicate information about the model in the most intuitive way, such as using classical visualization methods; and a *quantitative* level, to provide sound evidence to confirm the insights obtained in the previous level.

Sperrle et al.[14] provided a systematic analysis of how evaluations are carried out in human-centered machine learning papers, with trust as one of the important focuses of the survey. They identify trust issues in relation to the interaction between the performance and the presentation: even VA systems with the highest usability must consider the performance of their underlying ML models in order to remain useful, while, on the other hand, well-performing ML models might not be used to their full potential if users do not trust them. Trustworthiness is considered an important dimension of analysis of both model properties ("A model can be considered trustworthy when users believe it is correct") and the explanations themselves ("The ability for the explanation to be believed in or accepted by the user as an honest representation or correct description"). The authors indicate, however, that only a small percentage of the analyzed papers actually evaluate such characteristics: 10% for trustworthiness as a model property and 6% for trustworthiness in explanations.

Probably the most related work to ours is the survey by Chatzimparmpas et al.[15] where a comprehensive mapping of the currently available literature on using visualization to enhance trust in ML models is provided. The authors discuss which visualization techniques are used, how effective they are, and the domain areas they are applied to, including a conceptual discussion of what trust means in ML and what challenges are still open. However, the issue of explicitly expressing and/or managing trust within the VA

TABLE 1. Proposed trust framework key requirements.

Key requirement	Detailed explanation ("The framework should…")
Tasks	Support trust expression, explanation, development, and communication.
Coverage	Apply to all steps of the ML pipeline (model design, training, execution, and result usage).
Generality	Support any type of ML application (e.g., classification, regression) and technique (e.g., feature engineering, deep learning, supervised/unsupervised learning).
Versatility	Address a broad class of users (e.g., scientists, ML professionals, nonspecialist users).

pipeline itself is not discussed in any of these surveys or their analyzed papers. While most of the related works mention the increase of trust in ML as one of their most important goals, they do not discuss how to directly achieve (or manage) that in a concrete manner. We intend, in this article, to bridge this gap by proposing and discussing the design decisions behind a concrete framework where trust is a first-class citizen within the VA workflow itself.

TRUST AS FIRST-CLASS CITIZEN

Based on the motivating example and related work, we argue that trust should be considered a "first-class citizen" throughout the entire process of constructing and using ML applications, much like data provenance has become a first-class citizen in visualization pipelines.[3] For this, we propose a conceptual framework to represent, express, explore, communicate, and develop trust. Table 1 lists the key requirements this framework aims to comply with, based on the authors' own experience in building VA and ML applications. This list is not exhaustive but shows the requirements we believe are minimally needed.

To build this framework, we start bottom-up by first considering the traditional ML process. Figure 1 (bottom) depicts this as a data flow pipeline (data $=$ sharp-corner boxes, operations $=$ rounded-corner boxes). It starts by (1) acquiring *training data* T for the intended ML application. Using T, (2) ML professionals build and train an ML model M for the problem at hand. The model is next evaluated by its intended customers (3). Eventually, these decide to deploy and execute it (4) on post-training data D (also called "unseen" data in ML). Finally, the model produces a set of results $O = M(D)$, which are then used for the application at hand (5). By externalizing trust, we connect the different user roles (for more details, see the "Flow of Trust" section). Note that we do not explicitly show model monitoring and retraining, as we consider these reiterations of (parts of) the pipeline.

Each step of the ML pipeline can be characterized by five key elements (Figure 1 markers (i)–(v), shown only for pipeline step 1 to limit drawing clutter): *Users* consider their object of interest in the ML pipeline (i). This can be either a tangible object (training data T, trained model M, or model output O) or a process (model building, model execution). To assess the object, they next change its various *parameters* (ii) and observe their *effect*, i.e., how the object responds to parameter changes (iii). Based on this, they reach a *trust* conclusion (iv), which they next detail and document by providing *feedback* (v). These elements are described in the following (with additional examples in Table 2).

User roles: A role models the types of *activities* performed by a user involved in a given pipeline step. These can be taken by different, or the same, persons, depending on the application context, much like roles in the classical software engineering pipeline.[12] For instance, in a production setting, scientists or field researchers collect the training data (1); ML engineers construct the ML model (2) which is then deployed by IT professionals (3) and used in applications by the general public (4–5). In contrast, in a research or prototyping setting, all roles are often assumed by the same person.

Parameters: These describe how users *interact* with their object of interest (purple arrows marked P in Figure 1). For instance, training data can be resampled or transformed in various ways (1); training tunes various model hyperparameters (2); a trained model is deployed on platforms having different computing power provisions (3–4); and the model's outputs are shown to the end user via various parameterized visualizations (5).

Effect: This captures how the object under study *reacts* to changes of its parameters P and is shown in Figure 1 by the orange arrows marked E. Effects can range from simple numerical results (e.g., accuracy scores during training) to complex visualizations that depict the changing activations of units in a neural network during inference. Note that E also includes XAI techniques appropriate at each pipeline step. Exploring E allows users to form a mental model of the studied object and ultimately *explain* its behavior.

Trust: As users iteratively repeat the change-parameters-explore-results loop (ii)–(iii) outlined previously, they build an increasingly clearer trust (or lack thereof, with all in-between nuances possible) of the objects under study. The actual trust *conclusion*

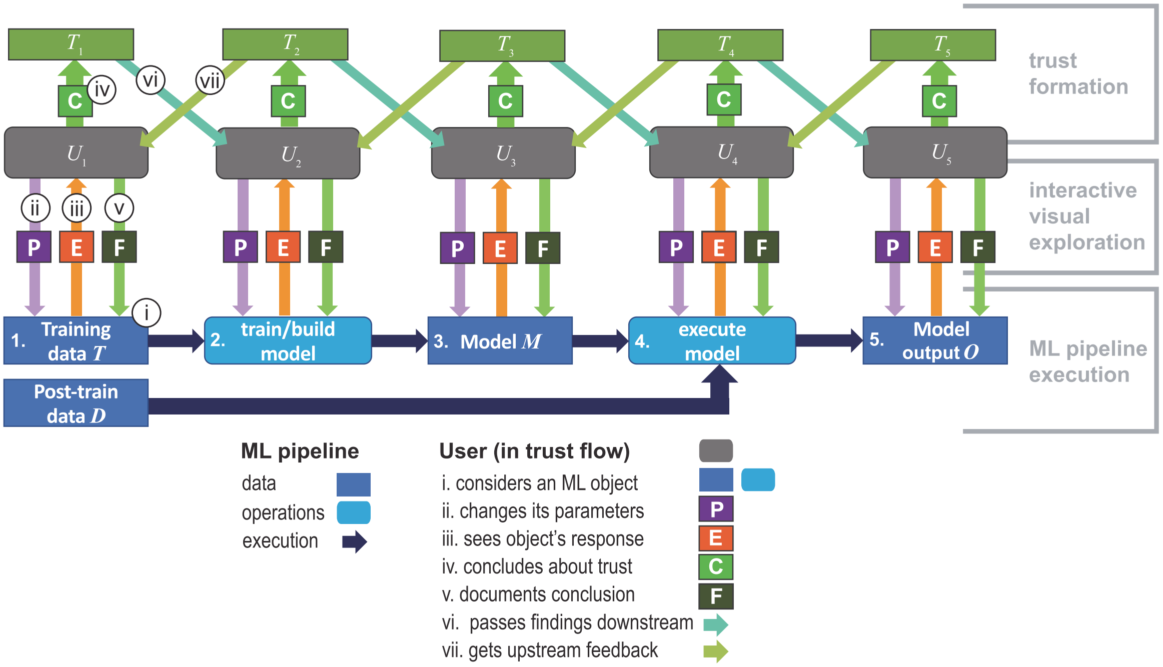


FIGURE 1. Trust modeling and flow throughout the construction and use of ML applications (see the "Trust as First-Class Citizen" section).

formed by users is shown by the green boxes T_i in Figure 1 top. These conclusions can be simplistically represented by values on a binary (yes/no) or on an ordinal (low to high) scale, but the trust state may be more complex and nuanced (e.g., not equal for different components or aspects of the object). Importantly, this trust forms up in the *mind* of the users (arrows marked C in Figure 1). As different user roles exist, it follows that *trust* has different meanings for the various pipeline steps (T_i, $1 \leq i \leq 5$, Figure 1 top). For example, a model engineer will trust a *model* M if it shows a good training convergence and it scores highly during ML testing scenarios; these aspects are not relevant for end users who will trust the *output* O of an ML pipeline if O is in line with their common expectations of what the pipeline should do.

Feedback: As explained, trust forms in the mind of a user. Modeling trust as basic ordinal values (see the previous section) offers a simple way to communicate a user's conclusion trustwise, but does not further explain *why* the user has reached that conclusion. This is important since both when trust T_i is high or low one needs to understand the reasons to react accordingly. Also, users may have unequal trust to different parts or aspects of the object of interest. We propose to solve this by making the abovementioned aspect explicit: A so-called *feedback* mechanism, denoted by the green arrows marked F in Figure 1, enables users to annotate their object of interest to explain what they (mis)trust and why. For example, end users can mark specific outputs O of a pipeline as untrustworthy (e.g., too many delayed flights); model engineers can mark aspects of a training process as suspicious (e.g., poor convergence curves or nonmonotonic changes of performance indicators); and data scientists can mark samples of a training set as potentially incorrectly acquired or labeled.

Flow of Trust

We have described so far how individual user roles arrive at achieving their own views of trust and how they can externalize these. In practice, this per-step formed trust next *travels* along the ML pipeline to connect user roles. We model this in Figure 1 by the diagonal arrows at the top. Arrows marked (vi) indicate trust *provisions* given by earlier pipeline steps to later ones (e.g., an ML engineer providing statistical metrics of model performance and representation of the distribution of model errors to justify their trust in the model engineering they performed). Simply put, trust "flows forward" in the pipeline to convince subsequent users that the objects they are provided with are trustworthy enough. Trust also propagates backward: arrows marked (vii) indicate trust *requirements* set by later pipeline steps to earlier ones (e.g., an end user telling his smart-driving car provider

TABLE 2. Examples of user roles, exploration parameters, explanation of ML behavior, trust aspects, and trust feedback mechanisms for the five steps of a generic ML pipeline.

Training data T	**User role**	**Collects and curates training data from a given application area.**
	Parameters	Affect the data representation (e.g., sampling and reconstruction parameters).
	Effect	Shows data properties (outliers, clusters) and potential problems (errors, missing values, duplicates).
	Trust	Data are sufficient, of good quality, and capture well the modeled phenomenon.
	Feedback	User determines unfit training data (e.g., missing, wrong, or duplicate values or poorly samples the intended distribution).
Model building	User role	ML practitioner involved in architecting, coding, training, and testing the model M.
	Parameters	Feature selection and engineering; problem decomposition; hyperparameters tuned during model engineering.
	Effect	Shows M's behavior in data and parameter spaces during training.
	Trust	Model works well for all applicable data and parameters and its sensitivity to data/parameters is understood.
	Feedback	Indicates that some of M's decisions (e.g., for specific samples) do not look correct and need improvement.
Model M	User role	ML practitioner; model evaluator (domain expert or certification body) determining model suitability for adoption.
	Parameters	Users explore model behavior by, e.g., applying it to different inputs, which act as parameters changed by the user.
	Effect	Model specific methods versus model agnostic methods. Depends on whether M is inherently interpretable or not.[9]
	Trust	Model is sound—works correctly, is efficient, well explained, and suitable for its intended usage.
	Feedback	Some model blocks are not needed or too complex; M is (not) understandable/(not) applicable to user's context.
Model execution	User role	Domain expert/integrator building an end-to-end solution using a given model.
	Parameters	Control the model's execution (e.g., memory and processor time available for a run).
	Effect	How the model modifies the solution during its execution process.
	Trust	Solution improves as the model runs; process converges fast enough; model avoids local minima.
	Feedback	The solution is evolving (in)appropriately.
Model output O	User role	End user of the ML pipeline (scientist, domain expert, ML engineer, nonspecialist).
	Parameters	Control how the outputs are shown (e.g., which text-based or visualization method is used).
	Effect	Bring insight how the model produces the output; XAI methods (LIME, SHAP, counterfactuals, local surrogate models).
	Trust	Based on domain knowledge, the output of M is plausible and in line with the users' mental model(s).
	Feedback	Selection of data items that comply to the users' mental model or not (continuous scale).

that they do not trust the car's behavior in certain conditions). Upon receiving such signals, users of earlier steps need to adapt their objects.

The flow of trust occurs by first passing the key conclusions *between* user roles (a U_i trusts object i this much, i.e., to level T_i). Next, additional information on *why* the respective trust level was reached can be passed along to justify the conclusion. Such information can also include details, such as particular components or aspects of the object, or conditions this level of trust refers to. The communication of trust takes the form of passing the annotated objects (obtained via the feedback F) that motivate the respective trust conclusion. Also, note that trust typically flows over multiple layers and multiple times during the lifetime of an ML pipeline, e.g., from the final users back to the scientists preparing the training data T. This is similar to the lifetime of software systems: the forward execution of the pipeline (and forward trust flow) is analogous to forward software

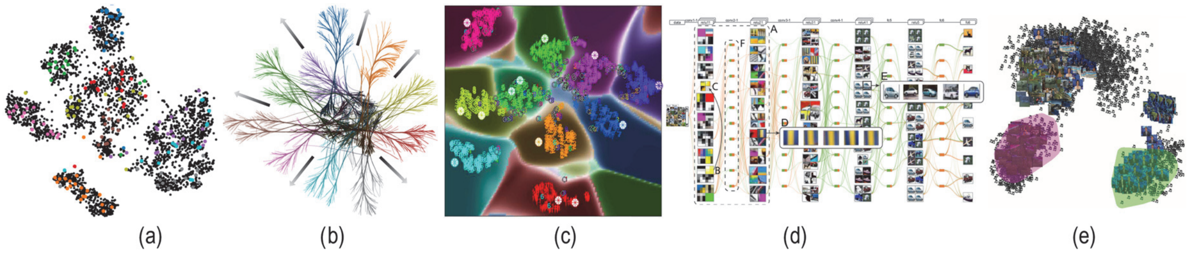


FIGURE 2. Examples of using interactive visualization in the trust modeling and flow. (a) Training data. (b) Train/build model. (c) Model. (d) Execute model. (e) Model output.

engineering from requirements gathering until the first deployed version. The backward trust flow is analogous to the collection and processing of change requests during software maintenance.[11]

Role of Interactive Visualization

Visualization plays a crucial role in our framework. First, it enables the *exploration* of the ML objects of interest by varying parameters P and observing effects E, since these objects are large, abstract, and complex. Second, interactive mechanisms allow users to select parts of these objects and annotate them to express their trust conclusions, thus to create the *feedback* F. Third, visualization enables an explicit representation of the trust (e.g., to track over time).

Tens of such visualization mechanisms exist—for a recent survey, see Garcia et al.'s work.[4] Figure 2 shows five such examples, one per pipeline stage. We selected techniques using dimensionality reduction (DR) as an underlining mechanism for ease of presentation and to demonstrate the model- and visualization-agnostic pipeline.

Training data: DR is the tool of choice in unsupervised learning to display large collections of high-dimensional samples to observe how these group (or not) into multiple clusters. In semisupervised learning, labeled samples are colored by their labels [see Figure 2(a)], enabling users to determine where in the training data to next perform annotations to enrich otherwise poorly labeled training sets,[5] and, thereby, improve their trust in such training sets.

Train/build model: DR can be used to visualize the evolving activations in the last hidden layer of a deep model (latent space). Figure 2(b) shows such evolutions as class-colored trails in a projection space, which increasingly diverge as training progresses. The visual separation of trails allows users to gauge their trust in the training and also spot outlier samples for which training did not perform well.[6]

Model: Classifiers can be assessed beyond typical aggregate metrics, such as accuracy by plotting so-called decision boundary maps [see Figure 2(c)]. These enrich a classical scatterplot-like DR projection of the input data space by coloring every pixel of the projected space to show the label (and its confidence) inferred by the model at that location.[7] Bright areas indicate regions of low confidence where the classifier is to be less trusted.

Model execution: To understand how large deep models process unseen input data, one can use DR to cluster their neuron activations and next depict the most salient input-data patterns that these respond to.[8] Figure 2(d) shows such patterns overlaid atop a clustered network architecture, which helps users gain trust by understanding how such black-box models actually operate.

Model output: Similar to the first stage, DR can be used to depict the output of a model (e.g., inferred classes) along with the input data [see Figure 2(e)]. This enables users to, e.g., mark in which regions of the data space (i.e., for which kinds of inputs) they trust the model or not.[10]

INTENDED USE OF THE TRUST FRAMEWORK

The purposes of this conceptual framework are to define a new research area in VA and to guide future research in this area. It is generally believed that VA can potentially help users to develop trust in ML models and, more generally, in various kinds of computational artifacts. However, the supposed help is currently limited to providing tools for interactive exploration of the artifacts (e.g., with XAI techniques). Our framework states that trust formation depends not only on the information users can gain by exploring an object but also on the flow of trust along the pipeline of the object construction and use. Referring to Figure 1, previous research has been focused on supporting the operations ii (parameters) and iii (effects). Our framework shows the need to support also passing findings downstream (vi) and receiving upstream feedback (vii). The key challenge that needs to be solved for developing this kind of support is to enable and facilitate explicit expression of trust. In terms of Figure 1, the task is to enable the operation v (expressing trust feedback) so that its results can be passed through the links vi and vii.

The framework shows that the meaning and structure of trust may not be the same for the different kinds of objects along the pipeline. Consequently, it is necessary to consider the specifics of each kind of object for understanding what contributes to the formation of trust in it. Table 2 includes our initial ideas concerning the possible meanings of the trust. This understanding, in turn, enables researchers to think how the essential ingredients of trust can be expressed explicitly. In other words, for a given kind of object, researchers will define, first, a conceptual model of the object-specific trust and, second, a suitable language to represent the trust. On this basis, researchers should work on developing interactive visual interfaces to facilitate externalization of the trust by the user (using the conceptual model to guide the user) and representation of the externalized trust by means of the language.

Solving the problem of trust externalization enables further research on supporting the trust flow along the pipeline. Typically, uncertainty also plays a role here. Appropriately representing uncertainty and its propagation along the pipeline is important information for users to make conclusions about the degree of trust. However, like explanations in XAI, representation of uncertainty and evaluation of its impact on trust building is an established research topic.[16] In our framework, we assume that users receive all relevant information, including uncertainties, for making trust decisions. Our focus is trust expression and communication.

The key question is how to support users with different roles to use trust feedback from the previous and next steps of the pipeline in fulfilling their roles. A related question is how to capture the evolution of the trust of each user resulting from the trust flow. We would like to emphasize that the purpose of this framework is to define research directions and pose research questions but not yet to give answers to these questions. Let us reconsider our motivating example from the air traffic domain to ponder how the trust issues could be addressed according to the proposed framework with a post hoc analysis.

In our motivating use case, the MD played the roles U_1, U_2, and U_3. The roles U_4 and U_5 belonged to the DU helped by VD. Based on our framework, MD would be expected to pass their trust in the model they built further along the ML pipeline (i.e., to VD and DU). MD would need to provide explicit trust feedback showing the reason for their trust (i.e., they would need to present evidence that the model operates appropriately). This would motivate them to explore the model carefully in order to create annotated visualizations for the following users. Thus, to verify and express their trust in the model, MD could apply it to test cases and visualize the characteristics of model performance across the cases: how the counts of delayed flights, unresolved hotspots (if any), and the average and maximal delay duration depend on the original number of the hotspots and the number of involved flights. This would demonstrate to DU that the model performance is good.

In reality, MD were not used to doing visual explorations. Therefore, their trust was communicated implicitly without being supported by evidence. DU with the help of VD explored the model behavior and its solutions and found a number of reasons for mistrust, as described earlier. They provided their feedback orally and in written form. Since there was no convenient way for DU to complement their feedback with annotated illustrations, the comments were rather general and insufficiently informative for MD to understand and address the problems. If DU were enabled to interactively explore the visualization received from MD, in particular, consider details of selected test cases, they could mark the flights deemed to be excessively delayed and ask MD to provide justifications. In response, MD might visually demonstrate to DU how a decrease in the delays of the marked flights would lead to the appearance of unresolved hotspots. We believe that *explicit expression* and *appropriate representation* of the trust feedback would allow MD to better adapt the model to the needs of DU and also increase the DU's level of trust by communicating well-substantiated trust of MD forward along the ML pipeline.

Another example of the intended use of the framework (in a different domain) is sketched next. Assume an image classification model is built to predict item production faults. The end user, who is responsible for picking out the faulty products from the assembly line, uses a VA system to identify faulty products. Imagine the following scenario. 1) The VA system reports a fault in the production. However, after inspection, it turns out that the product contains no faults and the user concludes that the ML model produced a misclassification. 2) After multiple misclassifications, the trust in the model decreases. The user expresses trust through direct manipulation of the trust object in the VA system (e.g., a slider ranging from *no trust* to *full trust*). 3) Next, after some iteration, the trust drops below a predefined threshold. As a result, the misclassified items are annotated and passed downstream to the model stage where the responsible user role (the MD) is notified. 4) The developer (visually) tests the generalization of the involved class, and unfortunately, the model does not generalize well for this class. Now, the trust in *the data* is lowered. The user passes the data distrust to the previous stage, along with (a visualization of) the data items of interest. 5) The responsible user role for the training data stage then inspects if the involved class labels are correct. This user concludes the labels are correct and expresses a high trust that is passed forward, along with the findings, to the MD again. 6) The MD can now trust that the data labeling is correct and starts improving the model by adding more instances of the problematic class to the training data.

This example is kept simple to demonstrate the main concepts; in reality the objects, models, and interactions are more complex.

DISCUSSION

Explicitly modeling, interacting with, and visualizing trust in ML applications generates new questions and open areas for research. From the conceptual framework we derive and discuss the following five research directions for future work.

1) *Trust objects:* taxonomy of trust objects, trust issues, and possible reasons for (mis)trust.
2) *Formalisms:* to represent trust in machine-readable form.
3) *Expression:* ways for users to express their state of trust by interacting with a computer system.
4) *Flow of trust:* ways to explore and develop trust over all stages of an ML pipeline using visual interactive techniques.
5) *Guidance:* ways to facilitate users' expression and communication of the state of trust using visual interactive techniques.

Trust objects: In this article, we identified and focused on the five trust objects of a traditional (classification/regression) ML pipeline: data, model development, model, model execution, and model output (see Figure 1, blue boxes). We believe our framework covers all main elements of the traditional ML pipeline at a high level of abstraction. The framework can be refined and applied to a broad range of ML model classes (classification, regression, optimization) as well as different methods of model building where trust objects are also likely involved (e.g., reinforcement learning, active learning, and self-supervised learning). As a first step toward development of applications with explicit trust, all trust objects should be identified and categorized using taxonomy. For each trust object in this taxonomy, different trust challenges play a role (e.g., for the data object, trust in the data gathering/collection and subsequent labeling of the data plays a role); for the model output, trust in the model as well as (subsets of) the output is formed by the user. For a system that fully supports trust as intended with the conceptual framework (within and between each pipeline step), an identification and understanding of reasons for trust, or the lack thereof is needed.

Formalisms: Currently, trust is not expressed explicitly, but rather it implicitly forms in the mind of the user. As argued in this article, we believe trust should be expressed externally (for storage, interaction, communication, and to act upon). Trust can be expressed in many ways (e.g., through interactive widgets, emails from one user role to another, oral communication, or bug-reporting systems). To be able to reason about the most effective and efficient manner of externalizing trust, we need to devise generic formalisms to represent trust in machine-readable form.

Expression: An open area of research is the exploration of which visualization and interaction mechanisms are most effective to express trust. Next to visualization and interaction, the coarseness of trust needs to be researched—how many levels are appropriate, are they similar for each trust object, and is their scale linear? Also, we believe the expression of trust depends on the stage, user role, and task. A related question is how to support both expert and novice (non-ML) users. Furthermore, future research should focus on creating a convenient language for users to express their state of trust through interactions.

Flow of trust: An important aspect of the framework is the communication of trust between the different user roles. To support this flow of trust between user roles, we believe interactive visualization is crucial and can act as common ground between the different stages. For example, visualizations can be shared between two subsequent stages and serve as a means of communication between both user roles. Next to design of interactive visualization techniques to support the flow of trust, also provenance plays a role here. A promising research area is how to capture, monitor, and visualize the evolution of trust over time, for exploration, analysis, and presentation.

Guidance: In similar spirit to exploratory visualization, where users are guided and steered toward interesting patterns, trust can also be used for guidance and assisting users in the analysis process of each stage. For example, users can focus on the subgroups with the most stable or highest trust by analyzing how trust evolved over time for a selected output (or group of outputs). Or if trust decreases over time, communicate this to the previous stage, such that this can be investigated and possibly fixed. For this, appropriate interactive visualization techniques should be developed. Similar to expressiveness, the methods and techniques should support guidance of both expert and nonexpert users.

CONCLUSION

Up until now, trust has not been considered as an explicit element in the design and reasoning about VA and ML applications. Rather, trust is an implicit process that takes place in the user's mind. We argue that trust should be externalized and treated as a first-class citizen. We present a framework that creates awareness and helps users to efficiently and effectively build and communicate trust in ways that fit each of the ML process stages. The framework is based on the traditional ML pipeline and extends this with elements of trust formation and interactive visual exploration. Key to our framework is the feedback loop *within* one stage through changing parameters, witnessing the effect or explanation, and providing trust feedback, and *between* stages, through passing or receiving externalized trust objects along the full pipeline (the flow or trust among different user roles). In addition to the framework, we identify and discuss five research directions for future work including trust objects, formalisms, expression, flow of trust, and guidance.

ACKNOWLEDGMENTS

This work originates from the Dagstuhl Seminar on *Interactive Visualization for Fostering Trust in ML* (seminar 22351). The authors were supported by the Academy of Finland decision 327352, by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr Institute for Machine Learning and Artificial Intelligence* (Lamarr22B), and by the EU in the projects *SoBigData++* and *CrexData* (Grant Agreement 101092749).

REFERENCES

1. N. Andrienko, G. Andrienko, L. Adilova, and S. Wrobel, "Visual analytics for human-centered machine learning," *IEEE Comput. Graphics Appl.*, vol. 42, no. 1, pp. 123–133, Jan./Feb. 2022.
2. G. Andrienko, N. Andrienko, J. M. C. Garcia, D. Hecker, and G. Vouros, "Supporting visual exploration of iterative job scheduling," *IEEE Comput. Graphics Appl.*, vol. 42, no. 3, pp. 74–86, May/Jun. 2022.
3. E. D. Ragan, A. Endert, J. Sanyal, and J. Chen, "Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 31–40, Jan. 2016.
4. R. Garcia, A. C. Telea, B. C. da Silva, J. Torresen, and J. L. D. Comba, "A task-and-technique centered survey on visual analytics for deep learning model engineering," *Comput. Graphics*, vol. 77, pp. 30–49, 2018.
5. B. Benato, A. Telea, and A. Falcao, "Semi-supervised learning with interactive label propagation guided by feature space projections," in *Proc. 31st SIBGRAPI Conf. Graphics, Patterns Images*, 2018, pp. 392–399.
6. P. Rauber, S. Fadel, A. Falcao, and A. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 101–110, Jan. 2017.
7. A. Schulz, A. Gisbrecht, and B. Hammer, "Using discriminative dimensionality reduction to visualize classifiers," *Neural Process. Lett.*, vol. 42, no. 1, pp. 27–54, Nov. 2014.
8. M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 91–100, Jan. 2017.
9. R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2019.
10. P. Joia, F. Paulovich, D. Coimbra, and J. A. Cuminato, "Local affine multidimensional projection," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2563–2571, Dec. 2011.
11. P. Tripathy and K. Naik, *Software Evolution and Maintenance: A Practitioner's Approach*. Hoboken, NJ, USA: Wiley, 2015.
12. I. Sommerville, *Software Engineering*. London, U.K.: Pearson, 2015.
13. A. Endert et al., "The state of the art in integrating machine learning into visual analytics," *Comput. Graphics Forum*, vol. 36, no. 8, pp. 458–486, Dec. 2017.
14. F. Sperrle et al., "A survey of human-centered evaluations in human-centered machine learning," *Comput. Graphics Forum*, vol. 40, no. 3, pp. 543–568, Jun. 2021.
15. A. Chatzimparmpas, R. M. Martins, I. Jusufi, K. Kucher, F. Rossi, and A. Kerren, "The state of the art in enhancing trust in machine learning models with the use of visualizations," *Comput. Graphics Forum*, vol. 39, no. 3, pp. 713–756, Jun. 2020.

16. D. Sacha, H. Senaratne, B.C. Kwon, G. Ellis, and D. A. Keim, "The role of uncertainty, awareness, and trust in visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 240–249, Jan. 2016.

STEF VAN DEN ELZEN is assistant professor of visual analytics at the Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612, Eindhoven, The Netherlands. His research interests include VA for explainable AI, network and event visualization. Contact him at s.j.v.d.elzen@tue.nl.

GENNADY ANDRIENKO is a lead scientist responsible for visual analytics research at Fraunhofer Institute for Intelligent Analysis and Information Systems, Sankt Augustin, Germany, PI of the Lamarr Institute for Machine Learning and Artificial Intelligence, and part-time professor at City University London, London, U.K. Contact him at gennady.andrienko@iais.fraunhofer.de.

NATALIA ANDRIENKO is a lead scientist at the Fraunhofer Institute for Intelligent Analysis and Information Systems, Sankt Augustin, Germany, PI of the Lamarr Institute for Machine Learning and Artificial Intelligence, and part-time professor at City University London, London, U.K. Contact her at natalia.andrienko@iais.fraunhofer.de.

BRIAN D. FISHER is a professor of interactive arts and technology at Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, and a cognitive psychologist by training. He conducts scientific research on the role of interactive graphical information environments in expert reasoning. Contact him at bfisher@sfu.ca.

RAFAEL M. MARTINS is an assistant professor at the Department of Computer Science and Media Technology, Linnaeus University, 352 52, Vaxjo, Sweden. His research interests include the area of dimensionality reduction and its uses in complex visual analytics workflows, such as for explainable AI. Contact him at rafael.martins@lnu.se.

JAAKKO PELTONEN is a full professor of statistics and data analysis at Tampere University, 33100, Tampere, Finland. His research interests include statistical machine learning and exploratory data analysis including dimensionality reduction and visualization. Contact him at jaakko.peltonen@tuni.fi.

ALEXANDRU C. TELEA is a full professor of visual data analytics at the Department of Information and Computing Sciences, Utrecht University, 3584, Utrecht, The Netherlands. His research interests include high-dimensional data visualization, visual analytics for explainable AI, image-based information visualization, and multiscale shape processing. Contact him at a.c.telea@uu.nl.

MICHEL VERLEYSEN is a full professor of machine learning at the ICTEAM institute of UCLouvain, 1348, Ottignies-Louvain-la-Neuve, Belgium. His research interests cover high-dimensional data analysis and machine learning, nonlinear dimensionality reduction, small sample data analysis, and biomedical applications of ML. He is a fellow of the IEEE. Contact him at michel.verleysen@uclouvain.be.

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.

EDITOR: Usama Fayyad, u.fayyad@northeastern.edu

This article originally appeared in Intelligent Systems vol. 37, no. 4, 2022

COLUMN: AI INSIGHT

The Secrets of Data Science Deployments

Usama M. Fayyad, *Northeastern University and Open Insights, Boston, MA, 02115, USA*

Much attention is paid to data science and machine learning as an effective means for getting value out of data and as a means for dealing with the large amounts of data we are accumulating at companies and organizations. This has gained importance with the major waves of digitization we have seen, especially with the COVID-19 pandemic accelerating digital everything. However, in reality, most machine learning models, despite achieving good technical solutions to predictive problems wind up not being deployed. The reasons for this are many and have their origin in data scientists and machine learning practitioners not paying enough attention to issues of deployment in production. The issues range all the way from establishing trust by business stakeholders and users, to failure to explain why models work and when they do not, to failing to appreciate the importance of establishing a robust quality data pipeline, to ignoring many constraints that apply to deployed models, and finally to a lack of understanding the true cost of production deployment and the associated ROI. We discuss many of these problems and we provide what we believe is a pragmatic approach to getting data science models successfully deployed in working environments.

There is much talk about the use of artificial intelligence (AI), machine learning (ML), and data science in organizations and enterprises. The reasons for this are obvious: organizations are facing huge amounts of data that are generated from interactions with customers and from business operations. The volume and variety of these data have dramatically increased with digitization; accelerated by the COVID-19 Pandemic dictating the new necessities of remote work and digital customer interactions. Adoption of algorithms to analyze, understand, and utilize the data is a must as human abilities cannot scale to the size and complexity of the data.

While the strong interest of organizations is to leverage AI to optimize operations and customer interactions, and almost all companies talk about this strong interest to leverage AI, very few realize that most working AI solutions are data driven and thus heavily dependent on ML and data science. AI/ML/data science have benefited from the fact that plentiful data have helped to overcome the limitations of AI algorithms and have led to many AI solutions being regression or classification problems applied to labeled training data.

Ironically, most organizations and companies do not realize the fact that practical working AI solutions are primarily dependent on data. Most do not realize that the data needs to be in formats that are usable by the learning algorithms. These algorithms are obviously highly dependent on what the data represent. The algorithms require data as follows.

1) *High quality:* any errors in the data lead to the wrong models being constructed by the algorithms—the often repeated motto: *garbage in, garbage out*.
2) *High granularity:* as the algorithms have to build their models entirely from what the data say. Algorithms rarely have a model of the domain or other useful knowledge—they must induce all knowledge from what the data covers. Unfortunately, most organizations build their data and analytical solutions to serve human analysts. Humans need to see data at an aggregated level and rarely make

1541-1672 © 2022 IEEE
Digital Object Identifier 10.1109/MIS.2022.3196612
Date of current version 12 September 2022.

use of detailed granular data. Unfortunately, this is the opposite of what learning algorithms need.

3) *Reliable labels:* for data to be used to train algorithms, the desired outcome label (the classification or outcome) needs to be recoded reliably and systematically with the data. Since the labels are typically obtained from human workers doing their job, in most situations the data infrastructure does not record the valuable outcome labels. They may be entered into other systems or noted in some documents or notes.
4) *Structured and unstructured data:* much valuable information is contained in unstructured data, such as text, video, audio (e.g. recordings of customer service calls). In fact, Gartner and other firms estimate that the majority of data in any organization is unstructured.[5] Yet most database systems and database administrators and developers use only SQL and thus only structured data—leaving some 90% of the data being stored beyond convenient use in analytics.

These abovementioned problems not only limit the ability to build accurate data science models, but they make it difficult to deploy models in realistic production settings. For example, even if a data science project invests significant resources in getting a clean, high-quality, well-labeled dataset, with structured attributes derived from unstructured data, how do you get that same level of quality during the "production use" of the data science models. A predictive model built on clean and well-extracted data needs to be applied to a stream of production data that goes through the same refinements. This is often unattainable in most production settings. Hence all the work in showing the power of predictive analytics and data science becomes just a proof of concept and is not really usable in production.

Even if the data infrastructure and quality issues were to be solved, there is a long list of other problems that need solving. We discuss these in the following sections.

BEYOND JUST DATA PROBLEMS

I recently participated in a panel at the Predictive Analytics World Conference held in Las Vegas, NV in June 2022. The panel was titled "Most Models Don't Deploy—What Practices Are Needed to Get Them Launched?" and discussed why most machine learning models are never deployed in production. According to Gartner, over 85% of machine learning models that are built and demonstrated to solve a predictive problem are never deployed in practice.[1] The panel was about discussing why this paradox in the age of data and AI.

The reality is that even if we were to address the lack of suitable data for data science purposes (both for modeling and for use in production), many other issues stand in the way. These issues are unfortunately often ignored by ML and data science practitioners.[2] We list them in this section, and we provide our advice on the best approach to mitigate these issues and maximize the chances of data science model deployment.

1) Understanding business realities;
2) understanding regulatory issues;
3) socializing the data science solution;
4) gaining trust and understanding of the data science solution; and
5) proper accounting of the ROI of the solution.

With all the abovementioned hurdles in addition to the data quality and data infrastructure issues, no wonder that just demonstrating the accuracy of a model is insufficient for actual production deployment.

WALK BACKWARD TOWARD SUCCESS

The typical progression of building a data science-based solution to a problem in volves starting from data and demonstrating that a solution can be built through modeling. The reason for this is typically driven by the philosophy of the typical data science team: let us first show that a good model can be built from the data available. This typically involves spending time understanding what the predictive modeling problem is about, talking to domain experts about the importance of and desired properties of a solution, constructing, and optimizing the data science model, and finally evaluating the accuracy of the predictive model on a hold-out or validation dataset. All of this is valid and necessary work. However, it leaves us in a situation where all the problems mentioned in the previous section create a huge risk to deployment in production.

If we were to embark on a path that is more likely to minimize the deployment risk, a different approach would be suggested. This is basically starting at the business end assuming a data science solution exists, but assessing what is needed to make the solution work. Many do not follow this approach because they are keen to prove that a good data science model exists. In my experience, solving the data science problem is typically the easier part. While no guarantee exists that an acceptable model can be constructed, it is usually very likely that some model would achieve reasonable performance. The issue is to make sure all the other blocking issues are solvable: a much harder and mostly non-technical set of problems.

Choose the Right Problem

When approaching domain experts in the initial effort of *problem discovery,* many experts will provide problems that are nice to solve or that they believe is a problem suitable for a data science solution. Sometimes these problems happen to be the favorite problems of the experts as opposed to the most pressing problems for the business. The selection of the problem should be driven by the strong need of the business and by the likely value contributed to help the business perform and/or compete better.

Business Value of Solution

The first thing to verify is that the intended solution (assuming the data science model works) is a high-priority solution. The question to answer here is*: If the solution is possible, does it solve a critical problem for the business?* This determines the prioritization the project would get and the ability to implement the necessary changes to actually deploy the solution in production. If higher priority problems are more important, then it would be wiser to start with one of those that can benefit from a data science solution.

Return on Investment (ROI)

Assuming a good data science model is achievable, it is critical to compute and verify a true ROI estimate for the solution. The ROI should account for not only the development costs, but the changes needed to productionize the model, the needed effort to maintain the models and refresh them, and the required data infrastructure to create the right data content and infrastructure. Engineering or analyst-driven ROI estimates are not sufficient. Typically, it is important to get agreement from finance, legal, and customer service teams to get a credible ROI. Many financial factors in costs, operations, and service are overlooked by data scientists performing ROI estimates.

Building Trust in the Solution

Let us say that a great solution can be derived through a data science model, how do we make sure that the data-driven solution will be acceptable to the stakeholders?[4] For example, if the problem is a predictive maintenance problem where we are going to assess whether a system will need maintenance before it actually fails, what would be needed for operating and managing stakeholders to accept the recommendation? An engineering manager responsible for running a plant would need to believe that the system will indeed need to be shut down to effect the fix. Often this has serious business consequences and costs. The manager would have to have trust in the data science predictions to act on them.

One of the most effective ways to gain trust is to have good explanations for the system's recommendations. The explanations have to make sense to the manager and have to present acceptable evidence. The fact that a model predicts something is typically insufficient.

A great way to build up the trust in the system, its recommendations, and its explanations are to involve the stakeholders in the construction of the model. Running the model on the side and asking for feedback and guidance from the stakeholders ensures they better understand how the predictive model is working and makes them feel that they are playing a critical role in fine-tuning and optimizing the model. Thus it becomes their model rather than some mystery box that is making predictions. Explanations can also include examples of prior situations (data) that preceded a system experiencing a failure. Sometimes seeing the data will elicit great additional knowledge from the experts, which can be very valuable in making the predictions much more robust.

Following the Data Chain Carefully

It is critical to understand what all the needed data inputs are, the availability of these data, their quality, and the series of transformations needed to extract the right attributes from the data. The fact that data scientists can work around the data quality issues and can perform the needed transformations to get a good model to perform is insufficient. These data corrections, transformations, and extractions have to be automated and put into production in order for the data science model to work appropriately. Thus addressing issues of creating a production environment for the needed data processing and transformation is an important and critical requirement.

Regulatory and Legal Considerations

Detecting some of the possible showstoppers is best done in advance of investing in making the data science model work. Are there permission issues with using the needed data? Are the risks acceptable in terms of privacy considerations and biased decisions by the data science models? Are there risks in markets

FIGURE 1. Taking small steps is a more effective approach to surmounting major hurdles on the path toward the solution—figure adapted from a presentation by James Taylor.[6]

(customers, users, and regulators) reacting negatively to errors in modeling and predictions?

CLIMB TO THE SOLUTION IN SMALL STEPS

Many of the problems mentioned can be intimidating and indeed formidable to surmount. However, the best approach is to break them down into smaller steps that are easier to tackle.

Climbing toward the solution not only requires “walking backwards” as suggested previously, but also taking smaller less risky steps so that progress is more continuous and measurable. Figure 1 illustrates the concept of small steps makes a more practical path towards ascending major hurdles over time.

The continual progress in small steps is often critical to making sure that the rest of the organization, management, and operations teams, do not perceive the data science project as stuck. Small progress is much more acceptable than longer waits against hard problems. Getting timely and practical help in achieving the small steps is also significantly easier and less risky than requesting help on more major problems.

BEYOND THE SUCCESSFUL DEPLOYMENT

In most data science-driven projects, the data science teams are focused on getting a predictive modeling solution to work and demonstrating its accuracy. These teams often forget that these models, once they become part of the production chain, need to be maintained, refreshed, and perhaps replaced as the market evolves and the data and assumptions change. Operations teams typically do not have the know-how of detecting when models need change or attention. Thus it is critical to have an approach to detect the health of models in deployment.

Much like any pragmatic system, data science models need a built-in ability to track performance issues and to issue warnings that a model is no longer working as intended or expected. Sadly, most data science deployments do not think about this aspect. This self-monitoring capability is a great tool to prompt the operations team to bring in data science expertise to see what needs to be done to keep the models healthy.

The monitoring capability should ideally be a combination of detecting reduced accuracy and built-in checks for data changes. If the data distribution changes, then it is likely that model performance will become unreliable or even incorrect. Sadly, most of the literature on the topics of building machine learning and data science models does not pay much attention to the problem of measuring model health over time: e.g., Lwakatare et al.’s work[3] does not address maintenance in the process.

CONCLUSION

It is critical to pay a lot more attention to many problems that do not get much attention in the machine learning and data science literature if we are to address the problem of lack of deployment in production. Beyond addressing the organizational issues and priorities in the business, attention must be paid to building the right data infrastructure that collects the right data at the right quality and granularity for data science models to work. We suggested what we believe is a better approach to pursuing data science solutions and hopefully derisking the difficult path to achieving production deployment.

ACKNOWLEDGMENTS

The author would like to thank many teams and customers that he worked with at Open Insights on deploying models and data solutions that helped formulate these thoughts. This column was instigated by the panel at PAW Conference in Las Vegas, especially by conference chair Eric Siegel, panel moderator James Taylor, and co-panelists: Jodi Blomberg and Sharada Narayanan.

REFERENCES

1. Gartner Announcement at Gartner Data & Analytics Summit 2018, Feb. 2018. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence

2. D. Talby, "Why machine learning models crash and burn in production," Forbes Technology Council, Apr. 2019. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2019/04/03/why-machine-learning-models-crash-and-burn-in-production/
3. L. E. Lwakatare et al., "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation," in *Proc. 20th Int. Conf. Agile Softw. Develop.*, 2019, pp. 227–243.
4. U. Fayyad, "Toward trustworthy AI: Bridging the trust gap between humans and intelligent machines," Forbes Technology Council, Mar. 2022. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2022/03/29/toward-trustworthy-ai-bridging-the-trust-gap-between-humans-and-intelligent-machines
5. B. Gitenstein, "Why unstructured data is your organization's best-kept secret," Oct. 2021. [Online]. Available: https://www.geekwire.com/sponsor-post/unstructured-data-organizations-best-kept-secret/
6. J. Taylor, "Get better results by doing less machine learning," in *Proc. Predictive Anal. World Bus.*, 2022.

USAMA M. FAYYAD is an inaugural executive director of the Institute for Experiential AI, Northeastern University, Boston, MA, 02115, USA, where he is also a professor of the Practice in the Khoury College for Computer Sciences. He is also a chairman of Open Insights, a company he founded in 2008 to build data strategy, AI solutions, and data science deployments for large enterprises. His research interests include data science, AI, and machine learning as well as bigData technology. Fayyad received his Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA. He is a fellow of both the ACM and the AAAI. Contact him at fayyad@acm.org.

Conference Calendar

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

MAY

1 May

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

5 May

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

6 May

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

13 May

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

20 May

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

22 May

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

27 May

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

28 May

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

JUNE

3 June

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

4 June

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

10 June

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

16 June

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

17 June

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

19 June

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

24 June

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

25 June

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

26 June

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

27 June

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

2469-7087/24 © 2024 IEEE

29 June

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

JULY

1 July

- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), Nicosia, Cyprus

2 July

- COMPSAC (IEEE Annual Computers, Software, and Applications Conf.), Osaka, Japan

3 July

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

7 July

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

8 July

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

15 July

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

22 July

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

23 July

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

24 July

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

AUGUST

7 August

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

19 August

- Cybermatics (IEEE Congress on Cybermatics), Copenhagen, Denmark

21 August

- RTCSA (IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications), Sokcho, South Korea

25 Aug

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

27 Aug

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

SEPTEMBER

2 September

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

16 September

- ACSOS (IEEE Int'l Conf. on Autonomic Computing and Self-Organizing Systems), Aarhus, Denmark

23 September

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

24 September

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Kobe, Japan
- IC2E (2024 IEEE Int'l Conf. on Cloud Eng.), Paphos, Cyprus

6 October

- ICSME (IEEE Int'l Conf. on Software Maintenance and Evolution), Flagstaff, USA