COMPUTING edge

- Security and Privacy
- Machine Learning
- Autonomous Vehicles
- Software

APRIL 2021

www.computer.org

75 YEARS IEEE COMPUTER SOCIETY

IEEE

IEEE COMPUTER SOCIETY JOBS BOARD

Evolving Career Opportunities Need Your Skills

Explore new options—upload your resume today

www.computer.org/jobs

Changes in the marketplace shift demands for vital skills and talent. The **IEEE Computer Society Jobs Board** is a valuable resource tool to keep job seekers up to date on the dynamic career opportunities offered by employers.

Take advantage of these special resources for job seekers:

- JOB ALERTS
- TEMPLATES
- WEBINARS
- CAREER ADVICE
- RESUMES VIEWED BY TOP EMPLOYERS

No matter what your career level, the IEEE Computer Society Jobs Board keeps you connected to workplace trends and exciting career prospects.

75 YEARS IEEE COMPUTER SOCIETY

IEEE

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

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.

IEEE Computer Society Magazine Editors in Chief

Computer
Jeff Voas, *NIST*

Computing in Science & Engineering
Lorena A. Barba, *George Washington University*

IEEE Annals of the History of Computing
Gerardo Con Diaz, *University of California, Davis*

IEEE Computer Graphics and Applications
Torsten Möller, *Universität Wien*

IEEE Intelligent Systems
V.S. Subrahmanian, *Dartmouth College*

IEEE Internet Computing
George Pallis, *University of Cyprus*

IEEE Micro
Lizy Kurian John, *University of Texas at Austin*

IEEE MultiMedia
Shu-Ching Chen, *Florida International University*

IEEE Pervasive Computing
Marc Langheinrich, *Università della Svizzera italiana*

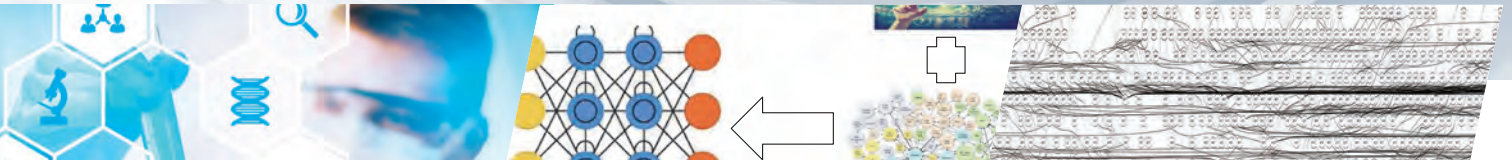
IEEE Security & Privacy
Sean Peisert, *Lawrence Berkeley National Laboratory and University of California, Davis*

IEEE Software
Ipek Ozkaya, *Software Engineering Institute*

IT Professional
Irena Bojanova, *NIST*

APRIL 2021 • VOLUME 7 • NUMBER 4

COMPUTING edge



8 Buying Your Genetic Self Online: Pitfalls and Potential Reforms in DNA Testing

16 Knowledge Graph Semantic Enhancement of Input Data for Improving AI

23 Towards Explainability in Machine Learning: The Formal Methods Way

Security and Privacy

8 **Buying Your Genetic Self Online: Pitfalls and Potential Reforms in DNA Testing**
ANDELKA M. PHILLIPS

14 **Policies on Privacy**
STEVEN M. BELLOVIN

Machine Learning

16 **Knowledge Graph Semantic Enhancement of Input Data for Improving AI**
SHREYANSH BHATT, AMIT SHETH, VALERIE SHALIN, AND JINJIN ZHAO

23 **Towards Explainability in Machine Learning: The Formal Methods Way**
FREDERIK GOSSEN, TIZIANA MARGARIA, AND BERNHARD STEFFEN

Autonomous Vehicles

28 **Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy**
ROBIN BLOOMFIELD, HEIDY KHLAAF, PHILIPPA RYAN CONMY, AND GARETH FLETCHER

37 **Validation of Autonomous Systems**
CHRISTOF EBERT AND MICHAEL WEYRICH

Software

46 **Queens of Code**
EILEEN BUCKHOLTZ

54 Mom, Where Are the Girls?
IPEK OZKAYA

Departments

4 Magazine Roundup
7 Editor's Note: Who's Doing What with Your Data?
57 Conference Calendar

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

Dynamic Assurance Cases: A Pathway to Trusted Autonomy

The authors of this article from the December 2020 issue of *Computer* propose a system architecture that facilitates dynamic assurance of autonomous systems embedding machine learning-based components. They also introduce dynamic assurance cases as a generic framework to provide justified confidence in these systems.

Computing in Science & Engineering

Optimal Kernel Design for Finite-Element Numerical Integration on GPUs

This article from the November/December 2020 issue of *Computing in Science & Engineering* presents the design and optimization of the GPU kernels for numerical integration, as it is applied in the standard form in finite-element codes. The optimization process employs auto-tuning, with the main emphasis on the placement of variables in the shared memory or registers. OpenCL and the first-order finite-element method (FEM) approximation are selected for code design, but the techniques are also applicable to the CUDA programming model and other types of finite-element discretizations (including discontinuous Galerkin and isogeometric). The auto-tuning optimization is performed for four example graphics processors and the obtained results are discussed.

IEEE Annals of the History of Computing

Olivetti ELEA Sign System: Interfaces Before the Advent of HCI

This article from the October–December 2020 issue of *IEEE Annals of the History of Computing* uses the case of ELEA 9000, the first Olivetti computer series, to demonstrate the close relationship between industrial design, semiotics, ergonomics, and the history of computing. A focus on the Olivetti ELEA series invites scholars to reconsider the history of computer interface design well before the emergence of HCI as a widely recognized field of research. The console and racks of the mainframe computer were designed by Ettore Sottsass Jr. (1917–2007) along with Andries van Onck (1928–2018) at the end of the 1950s. Aiming to launch the ELEA computer on the international market, Olivetti developed the idea of a visual language for human–computer interaction that could be learned by any operator, regardless of their native language. The task of designing this sign system was assigned to Tomás Maldonado (1922–2018). Together with Gui Bonsiepe (b. 1934), Maldonado designed a visual language that incorporated grammatical and syntactic reasoning. Later discarded, the sign system for ELEA prefigured the contemporary use of icons in computer interfaces.

IEEE Computer Graphics and Applications

Move&Find: The Value of Kinaesthetic Experience in a Casual Data Representation

The value of a data representation is traditionally judged based on aspects like effectiveness and efficiency that are important in utilitarian or work-related contexts. Most multisensory data representations, however, are employed in casual contexts where creative, affective, physical, intellectual,

 2469-7087/21

and social engagement might be of greater value. The authors of this article from the November/December 2020 issue of *IEEE Computer Graphics and Applications* introduce Move&Find, a multisensory data representation in which people pedaled on a bicycle to exert the energy required to power a search query on Google's servers. To evaluate Move&Find, they operationalized a framework suitable to evaluate the value of data representations in casual contexts and experimentally compared Move&Find to a corresponding visualization. With Move&Find, participants achieved a higher understanding of the data.

IEEE Intelligent Systems

Parallel Urban Rail Transit Stations for Passenger Emergency Management

In this article from the November/December 2020 issue of *IEEE Intelligent Systems*, a parallel urban rail transit station (URTS) system for passenger emergency management is presented based on the artificial systems, computational experiments, and parallel execution (ACP) approach. The agent-based modeling technology is applied to build the artificial URTS system, which contains the models of people, trains, facilities, events, environments, and center control and decision units. The computational experiments are performed on the artificial system to analyze and evaluate emergency management strategies. The mechanism of parallel execution between the actual system and artificial system is presented to manage and optimize the emergency strategy, which is capable of guiding the actual URTS system through real-time online supervision and adjustment and of providing an active optimization of passenger emergency management.

IEEE Internet Computing

Signing Blockchain Transactions Using Qualified Certificates

Blockchain technology is increasingly being considered among both private enterprises and public services. However, it poses a challenge with regard to aligning its identity management scheme with the public key infrastructure and the qualified digital certificates issued by qualified trust service providers. To solve this challenge, the authors of this article from the November/December 2020 issue of *IEEE Internet Computing* present an architecture reference model that enables enterprises and public services to leverage blockchain technology by integrating qualified electronic signatures with blockchain transactions. The evaluation of the architecture reference model is provided through the design of a blockchain-based trusted public service and a use-case scenario example. The proposed architecture reference model is based on the CEF building blocks EBSI, eSignature, and eID compliant with eIDAS.

IEEE micro

History of IBM Z Mainframe Processors

IBM Z is both the oldest and among the most modern of computing platforms. Launched as S/360 in 1964, the mainframe became synonymous with large-scale computing for business and remains the workhorse of enterprise computing for businesses worldwide. Most of the world's largest banks, insurers, retailers, airlines, and enterprises from many other industries have IBM Z at the center of their IT infrastructure. This article from the November/December 2020 issue of *IEEE Micro* presents an overview of the evolution of the IBM Z microprocessors over the past six generations. It discusses some of the underlying workload characteristics and how these have influenced the microarchitecture enhancements driving the performance and capacity improvements. The article then describes how the focus shifted over time

from speeds and feeds to new features, functions, and accelerators.

IEEE MultiMedia

WarpClothingOut: A Stepwise Framework for Clothes Translation From the Human Body to Tiled Images

With the increasing popularity of online shopping, searching for products with images for item retrieval has gradually become an effective approach. This trend is especially evident in the fashion industry. In common media, clothing items are usually worn on the human body. They can be straightforwardly segmented from the source media by utilizing detection or parsing algorithms. However, this may be deleterious to retrieval performance due to distortion, occlusion, and different backgrounds. In this article from the October–December 2020 issue of *IEEE MultiMedia*, a stepwise translation framework using a generative adversarial network and thin plate spline is developed to transfer human body images to tiled clothing images, which can be directly used for clothing retrieval. Experimental results demonstrate the effectiveness of the resultant tiled images produced from the framework compared to other methods.

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

Edge Computing for Legacy Applications

Edge computing was motivated by the vision of new edge-native applications that are compute-intensive, bandwidth-hungry, and latency-sensitive. The authors of this article from the October–December 2020 issue of *IEEE Pervasive Computing* show how infrastructure deployed for such futuristic applications can also benefit virtual machine (VM)-encapsulated Windows or Linux closed-source legacy applications. They present a new capability for legacy applications called edge-based virtual desktop infrastructure (EdgeVDI) and discuss example use cases that it enables.

IEEE SECURITY & PRIVACY

End-to-End Verifiable E-Voting Trial for Polling Station Voting

On 2 May 2019, during the United Kingdom's local elections, an e-voting trial was conducted in Gateshead using a touchscreen, end-to-end verifiable system. This was the first test of its kind in the United Kingdom, and it presented a case study to envisage the future of e-voting. Read more in this article from the November/December 2020 issue of *IEEE Security & Privacy*.

IEEE Software

Information Needs: Lessons for Programming Tools

Why is programming sometimes so frustrating and annoying and other times so fast and painless? This article from the November/December 2020 issue of *IEEE Software* surveys a few of the important lessons emerging from studies of programming and the new programming tools they motivate.

IT Professional

The IT Challenges in Disaster Relief: What We Learned From Hurricane Harvey

This article from the November/December 2020 issue of *IT Professional* explores the information systems involved in disaster relief supply chain for Hurricane Harvey survivors. The authors interviewed three organizations—the United Way, the BakerRipley, and the American Red Cross—on how information systems were used in this concerted effort of long-term recovery. They found that data sharing is the major challenge, and it is further constrained and complicated by legal concerns. They also observed that organizations used *ad hoc* technology solutions to accommodate different relief project needs; an integrated open-source system would not only save cost but also improve overall productivity.

Editor's Note

Who's Doing What with Your Data?

The Internet and Internet-connected devices enhance our lives in myriad ways, but they can also expose us to surveillance and data collection that put our privacy at risk. While some governments have enacted privacy-protecting regulations—most notably the European Union's General Data Protection Regulation—policymakers can do more to safeguard privacy in the digital age. Two articles from *IEEE Security & Privacy* take on contemporary privacy concerns and recommend related policies.

The first article, "Buying Your Genetic Self Online: Pitfalls and Potential Reforms in DNA Testing," discusses privacy issues related to direct-to-consumer genetic testing, a popular service provided by companies such as Ancestry and 23andMe. The article proposes regulations and standardization that could help mitigate the risks of companies obtaining genetic information. The second article, "Policies on Privacy," invites readers to consider the ethical implications of various types of data collection and use. The author asserts that societies should create privacy regulations to reflect their values.

Another ethical issue in technology today is explainability in machine learning (ML). *IEEE Internet Computing*'s "Knowledge Graph Semantic Enhancement of Input Data for Improving AI" discusses an iterative-optimization approach to knowledge graphs that helps improve ML explainability. *IT Professional*'s "Towards Explainability in Machine Learning: The Formal Methods Way" argues for using formal methods in ML to discover precise reasons behind algorithmic decisions and actions.

ML-based systems are important parts of autonomous vehicles. *Computer*'s "Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy" presents a framework for better dependability in autonomous systems such as self-driving cars. The authors of *IEEE Software*'s "Validation of Autonomous Systems" argue that employing intelligent validation and testing will help build public trust in autonomous vehicles.

The final two articles in this *ComputingEdge* issue celebrate women in software engineering and encourage gender diversity in the field. "Queens of Code," from *IEEE Annals of the History of Computing*, highlights 12 female programmers who worked for the US National Security Agency in the 1950s through the 1980s. In "Mom, Where Are the Girls?," from *IEEE Software*, the author describes how she came to recognize the gender diversity problem in software engineering and encourages individuals and organizations in the software community to advocate for diversity.

EDITORS: Khaled El Emam, kelemam@cheo.on.ca
Katrine Evans, k.evans@haymanlawyers.co.nz

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 17, no. 3, 2019

DEPARTMENT: PRIVACY INTERESTS

Buying Your Genetic Self Online:

Pitfalls and Potential Reforms in DNA Testing

Andelka M. Phillips, ***University of Waikato***

Today's world is one of constant monitoring and tracking—sometimes driven by us, sometimes driven by others. Developments in the field of health and identity are no exception. New technologies, such as wearable devices, and other technologies in consumer-centered health care allow us to track our fitness and health data, and they connect us with others.

Similarly, the rise in direct-to-consumer (DTC) genetic testing services, sometimes known as *personal genomics or commercial genomics*, can be viewed both as an example of emerging technology and also as disruptive innovation. These services have created a commercial market for genetic tests, allowing people to buy their own DNA tests online without a medical intermediary.

However, as with wearable health devices, DTC potentially affords opportunities for other entities to access and compile those data and subject us to profiling. Consumers, therefore, need to understand what's involved when we buy our so-called genetic self online.

This article provides a brief introduction to the world of DTC and its potential traps for the unwary. It discusses some short- and longer-term regulatory measures that may help to iron out the most serious risks to consumer privacy. In particular, it concludes that the industry needs more oversight and consumers need more control of their genetic data and personal data in the DTC context.

THE GROWTH OF DTC GENETIC TESTING

The market for DTC has experienced significant growth in the last couple of years with some prominent DTC companies having databases with several million consumers' samples.

Ancestry testing is particularly popular, but the industry varies widely with a broad spectrum of available services. The best-known ancestry and health tests are provided by prominent companies, such as 23andMe, AncestryDNA, Orig3n, MyHeritage, and FamilyTreeDNA. However, there are also companies offering lesser-known tests that are often more dubious, including assessing child talent, peace-of-mind paternity, and infidelity (often dubbed *surreptitious testing*). Several of these tests raise privacy and ethical concerns.

The proliferation and variety of services offered are increasingly attracting attention from researchers. My own research (due to be published as a book later this year) included a review of the online contracts of 71 DTC companies providing tests for health purposes. It found that a number of terms commonly included in these contracts were problematic from a consumer protection standpoint. Some companies, such as Soccer Genomics, have also raised concern from research scientists, with Stephen Montgomery at Stanford University launching a parody Yes or No Genomics website in response. Another parody website, DNA Friend, is a useful resource to highlight the sensitive nature of these services. However, these parodies do, to some extent, assume a level of knowledge about genetics, and we really need more efforts to assist the public in understanding the risks here.

While there is increasing public awareness of ancestry and health tests, what is less well understood is that these tests are generally not standardized and that any entity collecting genetic data could potentially use that data for secondary research or share it

Digital Object Identifier 10.1109/MSEC.2019.2904128
Date of publication: 14 May 2019

 Published by the IEEE Computer Society

with third parties, such as law enforcement. This article explores the problems that can arise as a result. It also discusses the existing and potential mechanisms that might help to resolve those problems.

A LACK OF STANDARDIZATION

In relation to DTC tests for health purposes, many tests for common complex diseases are not harmonized, and the validity of their findings is open to dispute.

In particular, DTC companies often do not provide whole genome scans and instead focus on portions of an individual's genome. Also, they can focus on different genetic variants and also frame their populations differently. As a result, it is possible to get contradictory disease-risk estimates from different companies.

The more common ancestry tests have also not been standardized, and it is similarly possible to obtain contradictory ethnicity estimates from different companies. There have even been instances of DTC companies providing DNA test reports on canine samples without distinguishing them from human samples. For example, in their article "Heredity or Hoax?" Barrera and Fox[1] discussed an example where a man had sent a dog DNA sample to a company (under a human name) and received an estimate of 20% First Nations ancestry.

This means that consumers need to be cautious about these services. At the very least, the public needs to be provided with more information about the limitations of testing because the utility of the service being sold may be less than expected.

SECONDARY USE OF GENETIC DATA

The potential for genetic data to be used in ongoing research is high. A number of the most prominent DTC companies have begun to partner with the pharmaceutical industry, and we have also begun to see investment by the insurance industry from these companies. One challenge here is that it is not possible to truly anonymize genetic data. (See, for example, the works by Erlich and Narayanan[2] and Gymrek et al.[3]). If something goes wrong, we cannot change our stored genetic data in the same way that we could change our bank password. So, it is particularly important that where DTC companies engage in such research, they implement strong security practices and infrastructure.

IN RELATION TO DTC TESTS FOR HEALTH PURPOSES, MANY TESTS FOR COMMON COMPLEX DISEASES ARE NOT HARMONIZED, AND THE VALIDITY OF THEIR FINDINGS IS OPEN TO DISPUTE.

It is important for consumers to understand the potential for secondary use here. The source of profit for DTC companies will often be partnerships and mergers with other entities, and there is a significant level of uncertainty here in relation to the variety of ways in which genetic data could be used in the future.

Use for law enforcement is also attracting increasing attention. In the last year, there was much media coverage of the genetic genealogy database GEDmatch's involvement in the investigation of the Golden State Killer case, where law enforcement accessed its database to find a potential suspect, through the process of familial DNA matching.[4] Since this revelation, it has emerged that more than 100 other DNA profiles from cold cases have been uploaded to GEDmatch.[5] In early 2019, it also emerged that the DTC company FamilyTreeDNA has been working with the U.S. Federal Bureau of Investigation to investigate violent crime (see, for instance, the work by Haag[6]).

GENETIC DATA ARE SENSITIVE IN NATURE

Genetic data are generally viewed as sensitive and can do real harm in the wrong hands. It is also much more than a method of identification in criminal proceedings. Genetic data have certain characteristics, which means that it can pose long-term privacy risks for individuals and their relatives.

Once you have a genetic test, your genetic code is digitized and that digital data can be stored potentially indefinitely and used for purposes beyond the primary purpose for which you gave it. It can also serve as a unique identifier for you, and since you share much of your DNA with your genetic relatives, it can also be used to trace those relatives. The impact of a data leak may be substantial, and it does not decrease over time.

The industry also operates internationally. Typically, consumers can purchase a test through a website, and then they will receive a sample collection kit in the mail. This is normally used for the collection of a saliva sample or a cheek swab, which is then sent back to the company for processing. Although services vary, companies will generally provide results through a web interface.

UNDER BOTH THE GDPR AND EU CONSUMER PROTECTION LEGISLATION, THERE ARE REQUIREMENTS FOR THESE DOCUMENTS TO BE IN PLAIN AND INTELLIGIBLE LANGUAGE.

From a regulatory perspective, the international nature of the industry creates complexity. The physical sample may be sent overseas and processed and stored by a company in a different country from where the consumer resides. The sequenced genetic data generated from this physical sample may or may not be stored in that same country. Also, DTC companies may collect other forms of personal data from their consumers through surveys and other research activities. Where this is stored may also vary, and again, it may be different from where the consumer resides.

These features, among others, affect how we need to think about regulation of businesses that handle genetic data.

THE IMPACT OF THE GENERAL DATA PROTECTION REGULATION ON DTC

Europe's data protection law, the General Data Protection Regulation (GDPR), is supposed to put users back in control of their data. It has direct relevance to the DTC industry: any company that sells or provides services directly to consumers based in the European Union (EU) needs to ensure that it complies with the GDPR.

Genetic data are included in the prohibition on processing of special categories of data in article 9 of the GDPR. Consequently, to comply with the GDPR, companies should be obtaining explicit and informed consent from their consumers for a DNA test. A more traditional notice-and-choice model is insufficient. In my research to date on the regulation of DTC, it seems likely that many businesses will need to alter their consent mechanisms to meet this higher standard.

Part of the problem is that e-commerce-based services have relied on their online information (including contracts and privacy policies) to govern relationships with consumers. However, providing clear online information about complex subjects can be a challenge. Also, we have all grown accustomed to ignoring terms and conditions and privacy policies on websites. This is due to a number of factors. One of the most significant problems is that people often lack the time to read these documents, and even where they do take the time, they may struggle to understand the contents. Many businesses have created longer contracts and privacy policies that are heavily skewed in favor of their interests, rather than those of their consumers. There has also been a lack of oversight of these documents. Consumers are deterred from reading them and may believe that they are not capable of challenging or changing the use of their information in any case.

However, under the GDPR, a high standard of consent is required for data processing, and it is not going to be acceptable to bury consent in a lengthy contract or to only make company policies accessible after a consumer has registered for a service. Under both the GDPR and EU consumer protection legislation, there are requirements for these documents to be in

plain and intelligible language. Because contracts and privacy policies are often linked together, problematic terms in contracts, which could be challenged on consumer protection grounds, may also be found to be problematic from a data protection perspective as well. EU consumer protection legislation also restricts the inclusion of terms that may be deemed to be unfair and limits their enforceability.

As the GDPR beds in, consumers are also starting to realize that they have genuine mechanisms to challenge what companies are doing with their data. The recurring and self-serving rhetoric expressed by some key players in big tech who say "privacy is dead" is changing. We are starting to see a shift with wide-reaching laws, such as the GDPR, together with growth in mega data breaches, resulting in calls for further regulation. Privacy is not only still alive—it is kicking. For example, the most recent annual report released by the Irish Data Protection Commissioner[7] (which is the first line of regulation for many tech companies in Europe) demonstrates that people do care about their privacy and that complaints lodged under the GDPR are likely to increase.

Many countries outside the EU are also reforming their privacy and data protection laws to cater for new developments. Simply stopping marketing DTC services to EU consumers, to avoid coverage by the GDPR, is therefore unlikely to be a viable solution. DTC companies will increasingly need to meet similar legal requirements for consumers based outside of the EU.

SUGGESTIONS FOR REFORM

The DTC industry has grown in the last two decades with relatively little oversight, during which time the potential of the technology has grown considerably. A number of policy documents have been released by diverse bodies, which could be drawn upon in improving industry governance. For example, the Science and Technology Committee of the United Kingdom has recently begun an inquiry into Commercial Genomics and is seeking public submissions. There is hope that this inquiry will lead to improved oversight of the DTC industry in the United Kingdom and may provide useful guidance for other countries considering how to regulate the industry. The disbanded Human Genetics Commission from the United Kingdom also previously developed a Common Framework of Principles, which could be drawn upon in developing new legislation or industry codes of conduct.

More suggestions for both short-and long-term strategies are provided next. There is no perfect solution, but a number of steps could lead to significant improvements for consumers and for improving standards across the industry.

THE DTC INDUSTRY HAS GROWN IN THE LAST TWO DECADES WITH RELATIVELY LITTLE OVERSIGHT, DURING WHICH TIME THE POTENTIAL OF THE TECHNOLOGY HAS GROWN CONSIDERABLY.

Short-Term Strategies

- The public needs more independent informational resources to assist them in making informed decisions about whether or not to utilize DTC services. Data protection authorities and privacy regulators as well as consumer regulators could release statements in relation to the industry. The Office of the Canadian Privacy Commissioner has already begun to take steps in this direction. It has released a number of documents in relation to DTC, including recommendations for questions that consumers could ask DTC companies and questions that they should ask themselves when considering purchasing a test. This example could provide a useful model for other regulators exploring these issues.
- Existing regulators should also consider developing industry codes of conduct and model privacy policies and consumer contracts. One potential foundation for such a code is the Future of Privacy Forum's paper,[8] which was developed in collaboration with some prominent DTC companies. This document makes a number of positive commitments in relation to privacy, but it is voluntary. It remains to be seen how businesses will adhere to this. Unlike the Future of Privacy Forum paper, though, any code should make it clear that American companies selling genetic tests to consumers based in the EU should still be complying with the GDPR.

- Another model is to make codes of conduct mandatory for the industry to follow. There may be reasonable support for such a move: DTC companies that wish to engage in health research and maintain consumer trust have an interest in showing that they comply with the law and support improvement of industry standards. They will wish to distance themselves from more dubious types of tests.
- Businesses should rethink their drafting of contracts and privacy policies. In relation to contracts, clauses that significantly limit consumers' rights should be avoided. For example, if businesses wish to be compliant with the GDPR and applicable consumer protection legislation, then they should not include clauses that allow them to change their terms at any time without notice to the consumer.
- Businesses should also think about their interface design. Given the sensitive nature of genetic data and the complex nature of some health test results, consumers should not be rushed into making a purchase. Putting speed bumps into the process, which encourage reflection and allow consumers to change their minds, could help to achieve compliance with the GDPR. It would be beneficial for businesses to allow for a cooling-off period as well in between purchase and processing of the sample.
- Businesses should also improve their practices in relation to deletion and destruction of physical samples and data. It should be possible for any company performing a genetic test to provide their consumers with the option of deleting the data and destroying the sample after sending the consumer their test results. Guardiome is an interesting example here because they offer consumers their whole genome sequence on a device, and their approach seems to be more privacy centric.
- Businesses should also keep in mind the GDPR's principles in relation to data processing. In the context of DTC, adhering to the data minimization principle could be particularly beneficial.
- At the national level, privacy and data protection regulators as well as consumer protection regulators should play a role in improving industry governance. Compliance reviews of privacy policies, contracts, and personal data practices, particularly in relation to security practice, would all be beneficial for improving industry governance.

Longer-Term Strategies

- We need more specific oversight of the industry to improve standards and ensure the protection of privacy and consumer rights more generally. One possibility is the creation of new regulatory bodies with a mandate to regulate all businesses that handle genetic data. This could draw upon existing models of data protection authorities and financial services regulators, and in some countries, this could be a new body that was under the oversight of the data protection authority.
- Tests of more dubious validity, such as surreptitious tests and child talent, should be banned, and regulators should help to alert the public about the most problematic services. In the United Kingdom, the Human Tissue Act makes it an offense to analyze DNA without appropriate consent, and it is likely that any company offering surreptitious tests to U.K. consumers will be in breach of this.
- New legislation is needed that deals more specifically with individual rights in genetic data. The recent Canadian Genetic Non-Discrimination Act could provide a useful model for other countries considering how to strengthen the rights of citizens in their genetic data.
- New industry-specific legislation should also be introduced at a national level, and international collaboration to develop more universal standards that could be followed globally could also help consumers given the international nature of these services.

This article has provided an introduction to the world of DTC and the challenges the industry poses for privacy. It is vital to understand that there is also a lot of uncertain risk in this context. We do not know all of the ways that our genetic data could be used in the future, but reform is needed given that we cannot change our genetic data and that it can always

potentially be linked back to us, can be used for many different purposes, and can also be used to trace our family members. People do need protection of their rights in this space and businesses should also view this as an opportunity to do things differently.

REFERENCES

1. J. Barrera and T. Fox, "*Heredity or hoax?*" CBC News, June 13, 2018. [Online]. Available: https://newsinteractives.cbc.ca/longform/dna-ancestry-test
2. Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Rev. Genetics*, vol. 15, pp. 407–421, 2014. [Online]. Available: https://www.nature.com/articles/nrg3423
3. M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.
4. R. Becker, "*Golden State Killer suspect was tracked down through genealogy website GEDmatch*," The Verge, Apr. 26 2018. [Online]. Available: https://www.theverge.com/2018/4/26/17288532/golden-state-killer-east-area-rapistgenealogy-websites-dna-genetic-investigation
5. M. Molteni, "*The key to cracking cold cases might be genealogy sites*," Wired, June 1, 2018. [Online]. Available: https://www.wired.com/story/police-will-crack-a-lot-more-cold-cases-with-dna/
6. M. Haag, "*FamilyTreeDNA admits to sharing genetic data with F.B.I.*," NY Times, Feb. 4, 2019. [Online]. Available: https://nyti.ms/2DVnK3x
7. Data Protection Commission. (2018). *Annual report: 25 May–31 December. Data Protection Commission*. Dublin, Ireland. [Online]. Available: https://www.dataprotection.ie/sites/default/files/uploads/2019-03/DPC%20Annual%20Report%2025%20May%20-%2031%20December%202018.pdf
8. FPF. (2018, July 31). *Privacy best practices for consumer genetic testing services*. Future of Privacy Forum. Washington, D.C. [Online]. Available: https://fpf.org/wp-content/uploads/2018/07/Privacy-Best-Practices-for-Consumer-Genetic-Testing-Services-FINAL.pdf

ANDELKA M. PHILLIPS is a senior lecturer at Te Piringa Faculty of Law, the University of Waikato, New Zealand, and a research associate at the Centre for Health, Law, and Emerging Technologies (HeLEX), University of Oxford, United Kingdom. Contact her at andelka.phillips@waikato.ac.nz

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 18, no. 2, 2020

Policies on Privacy

Steven M. Bellovin, *Columbia University*

Privacy is a hotly debated topic. But there isn't just one question—"Should we have more privacy?"—to answer. Rather, there are many, and until we reach consensus on the answers—and their consequences—we cannot agree on what regulation, if any, is appropriate. Bear in mind that the answers can be different for governments and for the private sector, and that this question in particular will be answered very differently by different people or different cultures.

The first set of questions concerns what sort of information can be used. Can an entity obtain information from others about a subject, or is it restricted to information it directly collects? Note that this interacts very directly with the issue of use controls: should collected data be used only for the specified purposes, or can it be repurposed? Secondary use of data—using data for something other than the reason it was originally collected—is one of the biggest sources of privacy problems. This is especially true if multiple datasets are combined.

What, though, constitutes direct collection? If I tag an online picture with someone else's name, is the site entitled to make the association between that person and the picture? Between me and the person I tagged? Between that person and me?

Direct collection is even murkier when it comes to web advertising. Is an on-page advertiser a direct collector? Is it the site hosting the page? Both?

If we want use restrictions, how do we define the categories of uses? What if someone changes his or her mind? Do we want exceptions for, e.g., medical research if identities are protected by contracts?

Digital Object Identifier 10.1109/MSEC.2020.2966900
Date of current version: 19 March 2020

These questions are common. Two less common issues are the existence of dossiers and the existence, in essence, of time machines.

A dossier is a large compilation of data about a particular individual, similar to what is compiled by credit bureaus and data brokers. These dossiers can be very powerful, but they're what Paul Ohm has referred to as *databases of ruin*. Note, too, that these databases need not contain personally identifiable information to be dangerous; a pseudonymous TiVo account can be just as violative to privacy as one with a real name, since the viewing history can often be deanonymized and linked to a real person.

Dossiers can enable time machines, the ability to see what someone did in the past, before they were of interest to someone else. Governments, of course, love that—but so do marketers. Should such dossiers be allowed to exist? Who should be allowed to query them? Should the information in them "expire" after a while? After how long?

Perhaps, for dossiers, we need revocable anonymity, so that law enforcement can get at the information, but not marketers. That, too, involves a policy decision, albeit a more legalistic one: what are the constraints on police?

It is important for society, not marketers, to answer the questions. For most answers, there are privacy-preserving cryptographic techniques that can at least approximate today's abilities where needed, but without endangering privacy or creating databases of ruin. There are already schemes for things like privacy-preserving targeted ads, verifiable income reporting with anonymous accounts and payment schemes, age verification credentials that don't show a name but are demonstrably valid, and more. I strongly suspect that most other necessary functions can be handled the same way, as soon as the requirements are agreed upon.

 Published by the IEEE Computer Society

There are certainly other important components to privacy, such as a requirement for clear and precise privacy policies by businesses—no more weasel words like *sometimes*, *may*, and *business partners*. But the important thing is to start by making explicit choices about the many different aspects of privacy.

STEVEN M. BELLOVIN is a professor of computer science and affiliate law faculty at Columbia University. Contact him via https://www.cs.columbia.edu/~smb.

EDITOR: Amit Sheth, amit@sc.edu

DEPARTMENT: KNOWLEDGE GRAPHS

This article originally appeared in IEEE Internet Computing vol. 24, no. 2, 2020

Knowledge Graph Semantic Enhancement of Input Data for Improving AI

Shreyansh Bhatt, *Amazon**

Amit Sheth, *University of South Carolina*

Valerie Shalin, *Wright State University*

Jinjin Zhao, *Amazon*

Intelligent systems designed using machine learning algorithms require a large number of labeled data. Background knowledge provides complementary, real-world factual information that can augment the limited labeled data to train a machine learning algorithm. The term Knowledge Graph (KG) is in vogue as for many practical applications, it is convenient and useful to organize this background knowledge in the form of a graph. Recent academic research and implemented industrial intelligent systems have shown promising performance for machine learning algorithms that combine training data with a knowledge graph. In this article, we discuss the use of relevant KGs to enhance the input data for two applications that use machine learning—recommendation and community detection. The KG improves both accuracy and explainability.

Machine learning algorithms trained with a large labeled data have shown promising performance in solving problems from various domains.[1] One of the most challenging aspects associated with using such algorithms is the availability of training data. On the other hand, symbolic knowledge representation has been a key area of Artificial Intelligence research since the mid 1970s, yielding a number of vetted background knowledge bases. The AI community started to use the term "Ontology" in the 1980's to refer to such background knowledge.[2] A patent filed in 2000 described the use of background knowledge to power commercial faceted and semantic search, semantic browsing, semantic personalization, and semantic advertisement.[3] Subsequently, background knowledge has played a key role in various tasks ranging from search and classification to personalized recommendations. In this decade, several researchers have explored the role of background knowledge to enhance the natural language processing and machine learning.[17]

In this article, we first describe the history of KGs and their application in research and industry, and introduce the problem of augmenting training data with the contents of a KG. We then review the most common approaches to augmenting data with knowledge, contrasting simple, explicit association of graph content with input data and approaches that depend on deep learning to combine separate KG and input content. We list the challenges associated with these and provide an overview of an alternative joint optimization-based approach for KG enhanced machine learning. We report four case studies in different domains that used this approach.

* Work done prior to joining Amazon

Digital Object Identifier 10.1109/MIC.2020.2979620
Date of current version 29 April 2020.

 Published by the IEEE Computer Society

HISTORY

Google introduced its "Google Knowledge Graph" in 2012, acknowledging its central role in their entity search[1]. Although a number of knowledge representation alternatives have been used, the graph has been one of the most popular formats to represent domain knowledge[2]. Apart from Google Search, a number of commercial products, such as Apple Siri and Amazon Alexa, are powered by a Knowledge Graph (KG).

A KG is a collection of facts where entities (nodes) are connected with typed relationships. The scope of the knowledge captured by a KG can vary with broad-based coverage that may involve many generic domains (e.g., DBpedia and Yago), a specific domain (e.g., Bio2RDF and UMLS for aspects of biomedical or medical domains), an industry or an enterprise.[18] To extract a KG for a domain of interest[4] domain specific KG creation approaches start from one or more entities (concepts) in the KG. The graph of interest is then created by traversing from those initial entities. However, the method does not gener as even a 2-3 hop traversal may end up including more than 50% of the source content.[4] To control traversal for generic sources, traversing is restricted through relevant relationships (edges) that can be identified by computing a specificity score of relationship to the domain. Research in creating domain specific sub-KGs has shown promising applications.[4]

KGs have potential applications in augmenting training data for machine learning algorithms. Training data augmented by a KG has been shown to enhance performance of applications that have limited training data, such as sentiment analysis, named entity recognition, recommendation, question answering, and object detection.[5–7, 16] However, the training data may not be available in the same form as the KG, hindering a facile augmentation of training data.

SIMPLE DATA ELABORATION USING A KG

KGs provide auxiliary factual information about the entities that are present in the training data. A simple approach to augmenting training data is to enhance the training data with auxiliary information extracted from the KG.[18] Consider a sentiment classification task on Tweets. If a Tweet mentions the president of the USA, a KG, such as DBpedia has the information that Donald Trump is the current president of the USA. The input data augmentation approach enhances the Tweet representation with concepts associated with the president of the USA in DBpedia. One of the early approaches for sentiment analysis used this strategy and reported an F1-score improvement with Tweets. Specifically, for each extracted entity (e.g., iPhone) from Tweets, this approach adds its semantic annotation (e.g., "Apple product") as an additional feature, and measures the correlation of the added concept with negative/positive sentiment.[5] Treating the concepts obtained from the KG as one of the Tweet features results in a 6.5% increase in the F1-score for sentiment classification.

KGs provide rich information that not only includes a type associated with the concept but also other related concepts. Training data augmentation with the KG content requires relative weighting of the following.

1. Concepts present in the training data.
2. KG concepts that map to the concepts in the training data, e.g., Tweet about Donald Trump maps to the dbpedia:Donald Trump in the DBpedia KG.
3. KG concepts that are associated with different types of relationships with the concept in the training data. E.g., dbpedia:President of the United States is associated with Donald Trump in DBpedia with dbpedia:type relationship.

AUGMENTED DEEP LEARNING WITH KGS

Separate training data, knowledge concepts, and related concepts from KGs can be the input to the neural network, which finds the appropriate importance of these different modalities. Artificial neuron patterns or neural network architectures that compound training data, KG concepts, and related concepts can differ depending on the nature of the training data (text or user-item interaction and time series or static). Most of the approaches use the first input layer of the deep neural network architecture as the layer that augments training data with the KG. The remaining layers are application and task specific with loss computed at the last layer of the deep neural network. The

end-to-end training of such a network results in learning the relative weighting between the training data and different concepts from the KG to solve the application or task, such as sentiment analysis, machine reading, recommendation etc. A key advantage of using the neural network to augment the training data with the KG is that a neural network can handle the nonlinearity involved in merging the training data and the KG. A set of artificial neurons, so-called neural network layer, considers the input from different modalities of data. Each neuron on this layer combines these different inputs and applies a (nonlinear) activation function. Hence, deep network with multiple layers can learn the appropriate nonlinear combination of the training data and the KG. This approach has shown promising results for various applications, such as sentiment analysis, recommendation, machine reading, and collaborative filtering.

Augmented deep learning for sentiment analysis: Kumar *et al.* proposed an approach to sentiment analysis that augments an input text word with concepts from WordNet KG.[6] The proposed model takes a sentence as input to a BiLSTM that computes a hidden representation of words in the sentence. The hidden representation is then passed through an attention layer along with the KG concepts related to the word. The attention layer then computes a weighted vector of the hidden representation of the input word and KG concepts. Weighted vectors of a sentence are passed through another attention layer, the output of which predicts sentiment. The whole network is trained to predict the correct sentiment of a given sentence in the training data.

Personalized news recommendation using KG: Wang *et al.* reported that a KG plays a key role in personalized news recommendation.[7] During the inference, the model predicts a click through rate for a given user and a given news story. The model generates user representations from their prior history of news click. The user representation is then concatenated with a given news story's representation to generate a *user* context vector. The resulting user context vector predicts the click through rate. For training, authors represented each news story with the entities found in the news story and context (neighborhood of the entity in the KG) of each entity. A multichannel representation is used for each word of a news story that corresponds to an entity in the KG. For example, one channel corresponds to the KG's representation for the entity and another channel corresponds to a representation of related entities. A convolutional neural network is then applied on such representation that combines word level information with the KG's information. Such a representation of each news story is then combined using an attention network to generate a user representation, which in turn is combined with the candidate news story representation to predict the click through rate.

Machine translation using KG: One of the challenges in using concepts from a KG to augment text data is a relative weighting of the actual word in the sentence and the word's representation from the KG. Yang *et al.* reported that while using background knowledge for machine reading, it is crucial to have both the relative weighing of word in the text and its KG-based representation, as in some cases the text context properly overrides the context-independent background knowledge available in KG.[8] They use a sentinel vector that combines the word and its related concepts in the KG. Their model uses a BiLSTM where the hidden representation from each BiLSTM unit is combined with related concepts from the KG corresponding to that word using the sentinel vector. The resulting vectored representation is used as the BiLSTM cell's hidden representation. Training this network results in learning the weights for the sentinel vector.

KG for recommendation: Zhang *et al.* showed that a KG can be used to solve the data sparsity issue arising from collaborative filtering for item recommendation.[9] They use an item's representation computed from the KG in user-item feedback matrix. This item's representation is computed by concatenating the visual and textual item's representation available in a KG. They showed that such a combined item's representation in the user-item matrix can improve collaborative filtering for recommendation. Other work on augmenting training data with a KG in recommendation, such as the Personalized Entity Recommendation[10] and Factorization Machine with Group lasso,[11] treat KG as a heterogeneous information network, and extract meta-path/metagraph-based latent features to represent the connectivity between users and items along different types of relation paths/graphs.

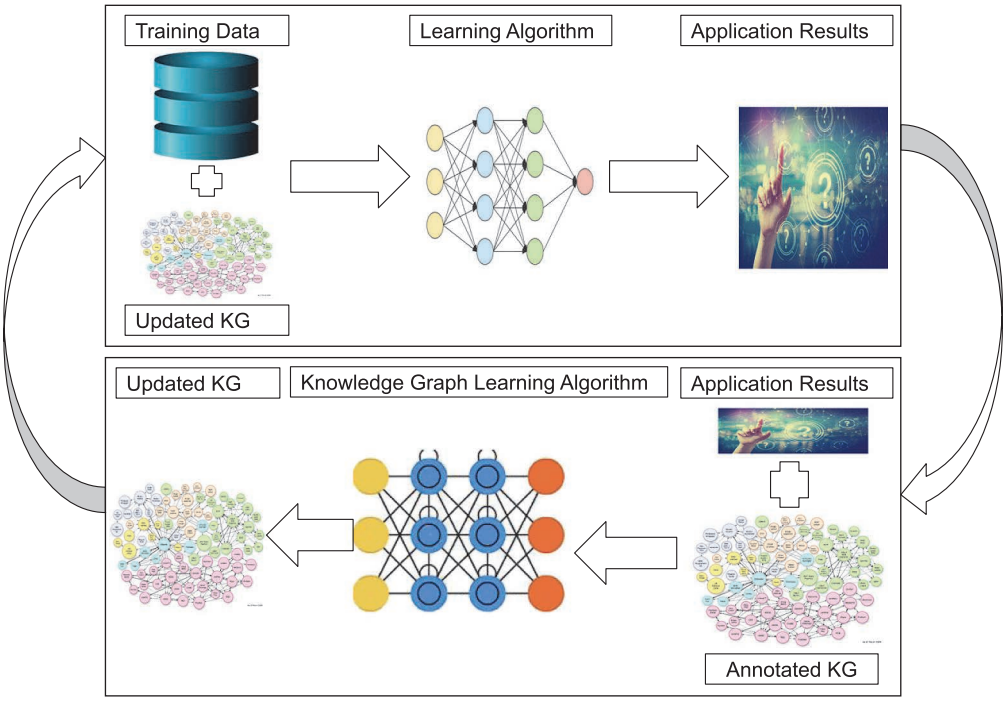


FIGURE 1. Iterative optimization for knowledge enhanced machine learning. Training data is linked/augmented with the KG. The learning algorithm is applied on the augmented data to find the results. The results then drive the annotated KG-based learning to identify the updated KG. The updated KG is then used in the training data augmentation.

CHALLENGES FOR KG AUGMENTED MACHINE LEARNING

As noted above, the data format of the input data and KG are often different from each other and require different processing algorithms and architectures. For example, most NLP tasks have sentences as the input data while background knowledge bases are available in graph form. To augment the input data with a KG requires either converting the knowledge into the format of input data or representing input data in the KG.

Differences in processing algorithm and architectures mean that augmenting the input data with a KG and converting into a common format may not lead to the best representation. KGs represent different kinds of information about a concept indicated by different types of relationships. For example, in DBpedia the concept dbr:ohio is connected with the concept dbr:USA by a hierarchical relationship dbo:country, whereas dbr:Ohio is connected to columbus with dbo:Capital relationship. The algorithms exploiting the KG must be cognizant of the different types of relationships. Moreover, these algorithms should find weights appropriate for the different relationships for the given task instead of using a generic relationship weighting of learned for a KG completion task.

Moreover, as the knowledge is fused with the training data before applying a machine learning algorithm with nonlinearity, the KG may not facilitate explainability.

To address these challenges, recent approaches propose specialized algorithms or neural network architectures for the input data and for the KG. We review these approaches as iterative optimization for enhanced machine learning using a KG.

ITERATIVE OPTIMIZATION FOR KNOWLEDGE ENHANCED MACHINE LEARNING

In order to augment input data effectively with a KG and to preserve the explainability potential of the KG, recent approaches iteratively optimize the task specific objective for the input data and for the KG representation of the data.[12, 13] As shown in Figure 1, these approaches start with an

initial KG representation. The input data are augmented with the initial KG representation. Optimizing the application-specific objective on such input data provides application-specific results. The knowledge graph is updated based on the results and the application-specific objective optimization is then run for a KG, which leads to an improved (application-specific) representation of the KG. The whole process is repeated until convergence or a predefined number of epochs. The initial generic KG representation augments the input data and iteratively updates the KG representation. Next, we review four approaches designed based on this concept.

KG enhanced recommendation: Wang *et al.* proposed a multitask feature learning approach for the KG enhanced recommendation.[12] For a given user-item interaction matrix, the item's representation is initially computed from the KG. A collaborative filtering technique is then applied on such a user-item interaction matrix. As a second step, the user-item interaction is represented in the KG with a concept specific to the item in KG and the other related item based on user-item matrix. The KG-based recommendation is then solved for predicting item similarity. Iterative optimization of these two objectives leads to learning an application specific KG representation that can be used for explanation and also to enhance performance for collaborative filtering. Wang *et al.* proposed to use "cross and compress" units for combining a KG's representation to item and user-item representation to KG.

KG enhanced community detection and characterization: KGs can also help improve our understanding of networked structured data (graph structured data), such as social networks. An attributed graph consists of nodes, attributes associated with nodes, and relationships (links) that connect nodes. For example, in a social network, users are nodes, location or user posts are attributes, and users are connected with friendship relationships. A group of users form a community when the number of relationships within a group exceeds the number of relationships across a group. It is often hard to divide a graph in communities. Node attributes can help explain certain communities. However, communities are often formed because a group of users share a generic concept. For example, a group of users may be friends with each other as they live in the same county. Readily apparent user attributes, such as a city name may not inform the county characterization for the group. The Crowdsourced KGs, such as DBPedia, have the information that connects cities to a county.

Bhatt *et al.* proposed the KG enhanced community detection.[13] They used iterative optimization over an attributed social network graph and a hierarchical KG to detect and characterize communities. The hierarchical KG can represent real-world communities or clusters. For example, a hierarchical KG for the geolocation domain represents the United States of America as a root concept with California and Ohio as subsuming concepts. Attributes on the nodes of the social network graph are mapped to the KG. Hence, each node is represented in the KG. Initially, each node is considered to be in its own community with a relationship weight computed according to the distance between nodes in the hierarchical KG. In each iteration, first the community detection objective is applied on the graph. Depending on the communities identified in the graph, the hierarchical KG is broken into multiple hierarchical KG representing each community. Hence, the communities identified in the graph inform KG representation of nodes and the input graph is modified based on the distance of nodes in the modified hierarchical KGs.

Unknown relevant domain: The initial mapping of the input data to the KG may represent multiple, or unknown domains. However, the objective optimization for the input training data may only depend on a certain domain or an intersection of multiple domains. Hence, if we augment the KG with the training data prior to optimizing the application specific objective, we may miss the appropriate background knowledge domain.

Social media-based wisdom of crowd analysis is one of the application domains where the objective optimization on the input data depends on the appropriate domain in the KG. Recent research shows that diverse crowds bring diverse perspectives in decision making.[14] Such a decision results in a more accurate forecast than a decision made by a randomly selected or homogeneous crowd. As users share their opinion on social media, we can use social media data to infer diverse crowds. Diverse crowd selection can be solved as subset selection, maximizing diversity within the

subset. As we want to measure diversity in perspectives in the given domain of interest, we can use data on individual users' attribute and augment it with the KG. However, it is hard to identify the domain of interest from the KG in the context of selecting a diverse crowd. Hence, we can find the diverse crowd by starting with a generic KG-based user attribute augmentation, and then find the appropriate domain of interest for the given set of diverse crowds. For example, a crowd may be diverse in the domain of politics, whereas it may not be diverse in the domain of sports. This results in the new domain of interest for diverse crowd selection and helps identifying the appropriate diverse crowd.

Generative applications: Domain specific short-text generation suffers from limited training data. Short and diverse text generation can benefit from the domain knowledge. Recent research shows that domain knowledge captured in the form of word2vec vectors improves text generation quality.[15] Here, the specific training data can be limited, whereas the data to generate word2vec vectors can consist of signals related to generic text generation, such as grammar and sentence structure. The use of domain specific KGs can further improve diverse text generation quality as it captures words and rules associated with the domain. However, it is challenging to identify the appropriate domain in the KG that helps the particular text generation. Iterative optimization can help such diverse text generation.

CONCLUSION

KGs play a key role in machine learning. Crowdsourced KGs can complement the available training data for machine learning algorithms and improve performance for a number of applications. Iterative optimization can further improve accuracy and also help explain the data in the context of the application. This approach is particularly useful when the entities present in the input data are associated with a concept in knowledge graph that is present in the multiple domains. An iterative approach can identify the appropriate domain in the context of the application.

ACKNOWLEDGMENTS

The authors would like to thank Manas Gaur and Ruwan Wickramarachchi for their review and suggestions. Dr. Sheth and Dr. Shalin's work was funded in part by NSF under award #1513721 titled "TWC SBE: Medium: Context-Aware Harassment Detection on Social Media." Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Alon Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, Mar./Apr. 2009.
2. T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *Int. J. Human-Comput. Stud.*, vol. 43, no. 5/6 pp. 907–928, 1995.
3. A. Sheth, et al., "System and method for creating a semantic web and its applications in browsing, searching, profiling, personalization and advertising," U.S. Patent No. 6,311,194. 30 Oct. 2001.
4. S. Lalithsena, "Domain-specific knowledge extraction from the web of data," 2018.
5. H. Saif, et al., "Semantic sentiment analysis of twitter," in *Proc. Int. Semantic Web Conf.*, 2012, pp. 508–524.
6. A. Kumar, et al., "Knowledge-enriched two-layered attention network for sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2018, vol. 2, pp. 253–258.
7. H. Wang, et al., "DKN: Deep knowledge-aware network for news recommendation," in *Proc. World Wide Web Conf.*, 2018, pp. 1835–1844.
8. B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMS for improving machine reading," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1436–1446.
9. F. Zhang, et al., "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 353–362.
10. X. Yu, et al., "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 283–292.
11. H. Zhao, et al., "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 635–644.

12. Ho. Wang, et al., "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 2000–2010.
13. S. Bhatt, et al., "Knowledge graph enhanced community detection and characterization," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 51–59.
14. S. Bhatt, et al., "Who should be the captain this week? Leveraging inferred diversity-enhanced crowd wisdom for a fantasy premier league captain prediction," in *Proc. Int. AAAI Conf. Web Social Media*, 2019, vol. 13. no. 01, pp. 103–113.
15. A. Nalamothu, "Abusive and hate speech tweets detection with text generation," 2019.
16. Y. Fang, et al., "Object detection meets knowledge graphs," 2017.
17. A. Sheth, et al., "Knowledge will propel machine understanding of content: Extrapolating from current examples," in *Proc Int. Conf. Web Intell.*, 2017, pp. 1–9.
18. A. Sheth M. Gaur, U. Kursuncu, and R. Wickramarachchi, "Shades of knowledge-infused learning for enhancing deep learning," *IEEE Internet Comput.*, vol. 23, no. 6, pp. 54–63, Nov./Dec. 2019.

SHREYASH BHATT is a machine learning scientist with Amazon.com, Seattle, WA, USA. Contact him at bhattshr@amazon.com.

AMIT SHETH (Fellow, IEEE) is the director of Artificial Intelligence Institute, University of South Carolina, Columbia, SC, USA (http://ai.sc.edu). He is a fellow of AAAI and AAAS. Contact him at amit@sc.edu.

VALERIE SHALIN is a cognitive scientist and a professor of Psychology with Wright State University, Dayton, OH, USA. Contact her at valerie.shalin@wright.edu.

JINJIN ZHAO is a machine learning scientist with Amazon.com, Seattle, WA, USA. Contact Jinjin at jinjzhao@amazon.com.

This article originally appeared in
IT Professional
vol. 22, no. 4, 2020

Towards Explainability in Machine Learning: The Formal Methods Way

Frederik Gossen, Tiziana Margaria, and Bernhard Steffen

Classification is a central discipline of machine learning (ML) and classifiers have become increasingly popular to support or replace human decisions. We encounter them as email spam detectors, as decision support systems, for example in healthcare, as aid in interpreting X-rays in breast cancer detection, or in the financial and insurance sector, for financial and risk analysis. For example, Facebook uses classifiers to predict the likelihood that users will navigate or click in a certain way, at scale, for millions and millions of users every day [9]. They also play a significant role in various areas of computer vision, where traffic signals and other objects need to be identified in order to "read" a situation during assisted or autonomous driving. Because we rely on classifiers not only for ease and comfort but also in business or safety critical systems, they need to be precise and reliable.

Classifiers foot on a wide variety of techniques: neural networks, statistical learning like Bayesian networks, instance leaning like in K-Nearest Neighbor, separability of classes in a vector space like in support vector machines, or logics, like in decision trees, random forests, and rule-based classifiers. ML classifiers were traditionally judged mostly in terms of precision, ease of training and fast response. In many cases, however, small differences in the sample led to spectacularly wrong decisions. Meanwhile, AI failure stories populate various sites[1] including fails by popular AI platforms like IBM's Watson.

When something goes wrong, it is good to know why. In cases where legal action follows a misclassification, as in the recent CervicalCheck cancer scandal* that rocked Ireland's Health Service,[2] it is important to be able to find out exactly why a certain classification verdict was issued. Ease of explanation is also particularly important when the proposed classification is correct, but apparently counter-intuitive. This is why *Explainability* is now a new hot topic in ML, and this is where formal methods can play an essential role. Let us show the power of the formal methods way in combination with random forests.

Random Forests are one of the most popular logic-based classifiers in ML. The larger they are, the more precise the outcome of their predictions. Figure 1 shows a random forest with 100 tree elements that was learned from the Iris Classification[3] problem of the popular UCI dataset.[4] The dataset lists dimensions of Iris flowers' sepals and petals for three different species of flowers: *iris setosa*, *iris virginiana,* and *iris versicolor*. These are our classes. Random Forests are a collection of many decision trees, each learned from a random sample of the training dataset. All trees have different structure, represent different decision functions, and can produce different decisions for the same input data. The training method is easy to understand and to implement, and at the same time achieves impressive classification accuracies in many applications.

Once we have the random forest, to classify previously unseen input data every decision tree is evaluated separately, potentially in parallel. The overall decision of the random forest is then typically derived as the most frequently chosen class, an aggregation commonly referred to as majority vote. Key advantage of this approach is the reduced variance compared to single decision trees. But can we explain how and why this decision was taken?

EXPLAINABILITY

Neural networks and random forests are considered

* The CervicalCheck cancer misdiagnosis was human, and not due to machine learning.

Digital Object Identifier 10.1109/MITP.2020.3005640
Date of current version 17 July 2020.

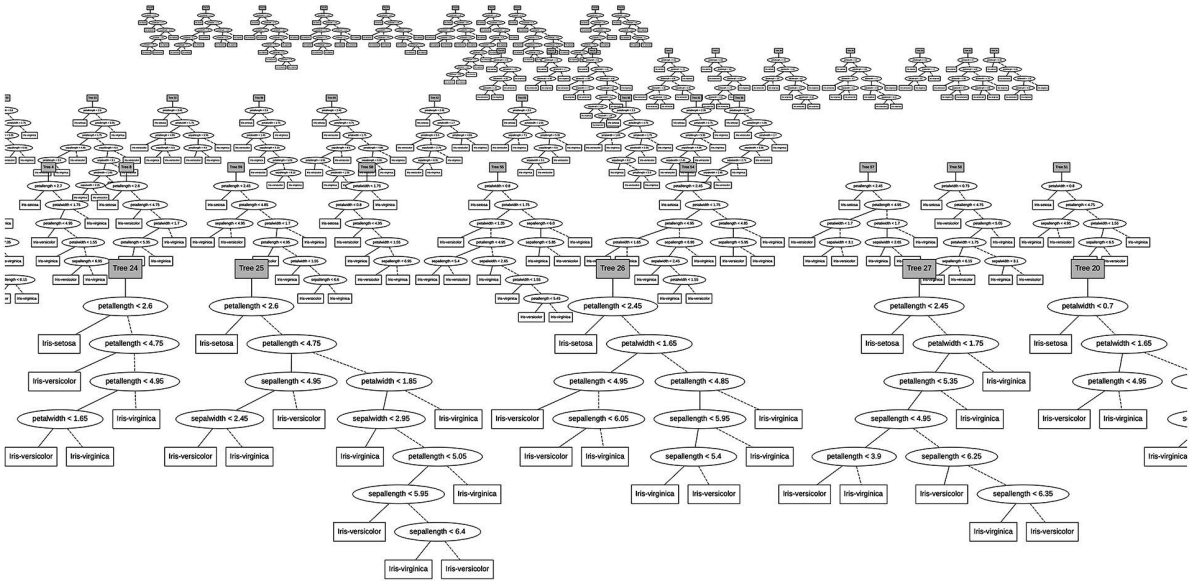


FIGURE 1. Excerpt from the considered Random Forest, which contains 100 trees with a total of 1312 nodes. Evaluation means majority vote-based aggregation of the evaluation results of the individual trees. Even only trying to understanding the reason for a certain classification on this basis is considered hard (outcome explanation problem).

black-box models because of their highly parallel nature: following the execution of neural networks means following sequences of parallel execution steps that result from a complex interplay of the value of all neurons (or nodes). The execution of a random forest is simpler, but it still requires to aggregate the results of each of its often many hundreds trees after having executed all of them individually. The results of such black-box executions are hard to explain to a human user even for very small examples.

In contrast, decision trees are considered *white-box models* because of their sequential evaluation nature. Even if a tree is large in size, a human can easily follow its computation step by step by evaluating (simple) decisions at each node from the root to a leaf. Indeed, the set of decisions along such an execution path precisely explains why a certain choice has been taken.

Popular methods towards explainability try to establish some user intuition. For example, they may hint at the most influential input data, like highlighting or framing the area of a picture where a face has been identified. Such information is very helpful, and it helps in particular to reveal some of the "popular" drastic mismatches incurred by neural networks: if the framed area of the image does not contain the "tagged" object, the identification is clearly incorrect. However, even in a correct classification, the tag by itself gives no reason why the identification is indeed correct.

More ambitious are methods that try to turn black-box model into white-box models, ideally preserving the semantics of the classification function. For random forests this has been achieved for the first time using algebraic transformations.[5] In fact, the proposed method is based on Algebraic Decision Diagrams (ADDs)[6] and Binary Decision Diagrams. An ADD is essentially a decision tree where redundant subparts are merged. A Binary Decision Diagram is an ADD over the algebra of Boolean values, i.e., the leaves are Boolean (true/false, yes/no). The method solves the following three explainability problems with absolute precision.

The *Model Explanation Problem* is solved in terms of an ADD that specifies precisely the same classification function as the original random forest.

The *Class Characterizations* Problem is solved in terms of a BDD that precisely characterizes all samples that the original random forest will classify as the considered class.

The *Outcome Explanation Problem* is solved in terms of a minimal conjunction of (negated) decisions that are sufficient to guide the sample into the considered class.

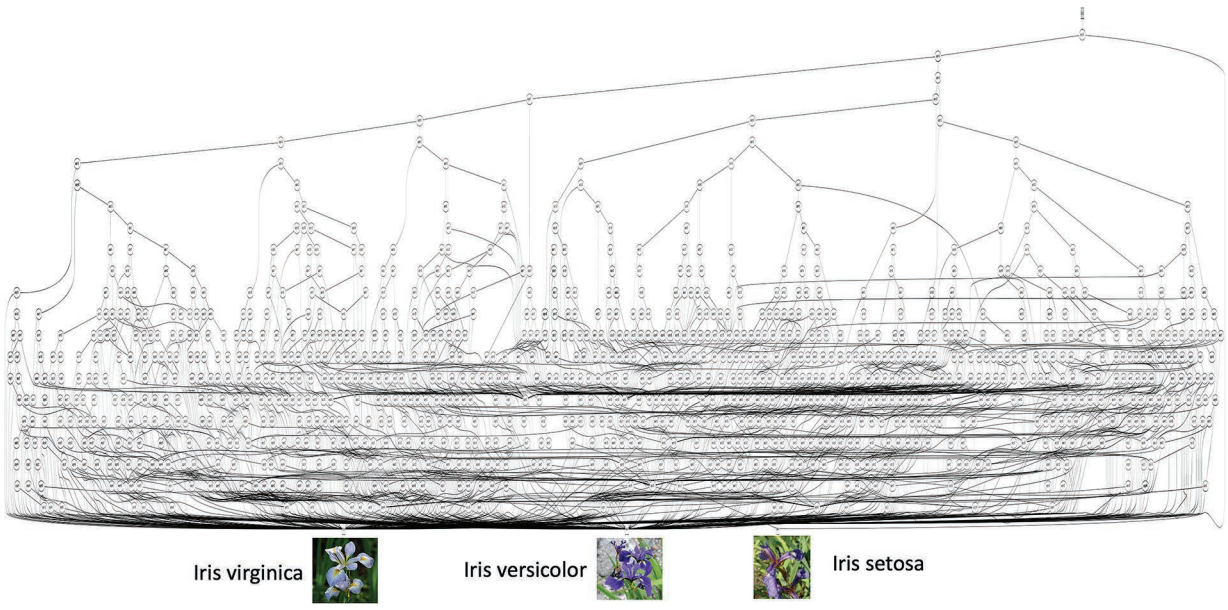


FIGURE 2. Model Explanation. This graph with its 1077 nodes is considered a white-box model for the Random Forest as individual classifications can be explained simply by looking at the corresponding classification path whose lengtt, in this case, never exceeds 20. Note that there are individual trees in the original Random Forest with paths of length 10.

We will now illustrate this approach and the three forms of explainability starting from the random forest with hundred trees for the Iris classification shown in Figure 1.

The black box character of this forest is obvious: given a sample, how can a human follow the 100 individual trees evaluations, grasp their essence and then understand the impact of the following majority vote-based classification? In the next section, we will see that this is an inherently hard task, because also the canonical white box model with its more than thousand nodes we are able to construct is still quite hard to understand.

MODEL EXPLANATION PROBLEM

The *canonical white box model* corresponding to the random forest of Figure 1 can be constructed compositionally by taking the individual trees of the random forest and successively "adding" their corresponding ADDs. This solves the Model Explanation Problem.

Figure 2 sketches the result of this construction: A canonical white box model with 1077 nodes. Admittedly, this model is still frightening, but given a sample, it allows one to easily follow the corresponding classification process, and in this case it may require at most twenty individual decisions based on the petal and sepal characteristics. This decision set is our set of predicates. The conjunction of these predicates is a solution to the *Outcome Explanation Problem*. However, more concise explanations are derived from the *class characterization BDD* discussed in the *"Class Characterization Problem"* section.

This construction exploits algebraic properties: intuitively, we "add" the entire decision trees. This is technically possible because ADDs inherit the algebraic structure of their leaf set. In this case, the algebra of the leaf set is the set of vectors that have one component for each class that counts how often this particular class has been chosen under the conditions represented by the paths to this very leaf. To add two ADDs of this set, we use the component-wise addition of the underlying vector structure.[5]

CLASS CHARACTERIZATION PROBLEM

The class characterization problem is particularly interesting because it allows on to "reverse" the classification process. While the direct problem is *"given a sample, provide its classification,"* the reverse problem sounds *"given a class, what are the*

FIGURE 3. Class characterization. This Class Characterization Model has only 53 nodes, a size that can be considered comprehensible by humans. Its maximal path length is 18. The path for our sample is 14, and the corresponding reduction leads to an outcome explanation with only 4 predicates. Note that with four classification parameters the outcome explanation can never have more than 4 predicates.

characteristics of all the samples belonging to this class?" The BDD shown in Figure 3 is a minimal characterization of the set of all the samples that are guaranteed to be classified as Iris Setosa.

Being able to reverse a learned classification function has a major practical importance. Think, e.g., of a marketing research scenario where data have been collected with the aim to propose best-fitting product offers to customers according to their user profile. This scenario can be considered as a classification problem where the offered product plays the role of the class. Now, being able to reverse the *customer → product* classification function provides the marketing team with a tailored *product → customer* promotion process: for a given product, it addresses all customers considered to favor this very product as in the corresponding patent.[7]

OUTCOME EXPLANATION PROBLEM

The path highlighted in Figure 3 defines an outcome *classification formula* for the sample

petallength = 2,49
sepalwidth = 2,45
sepallength = 7,15

As the conjunction of the following 13 predicates:

NOT petallength < 2.45
  petalwidth < 1.65
  petalwidth < 1.45
NOT sepallength < 7.05
  sepalwidth < 2.65
  petalwidth <1.35
  petalwidth < 0.8
  petalwidth < 0.7
NOT sepalwidth < 2.25

petallength < 5.0
petallength < 2.7
petallength < 2.6
petallength < 2.5

The classification formula expresses the collection of "conditions" that this sample satisfies, and it provides therefore a precise justification why it is classified in this class.

Despite the fact that the class characterization BDD is canonical, it is easy to see that there are some redundancies in the formula. For example, a petallength < 2.5 is also inherently smaller than 2.6 and 2.7; therefore, for this specific sample those two predicates are redundant. This is the result of the imposed predicate ordering in BDDs: all the BDD predicates are listed, and they are listed in a fixed order. After eliminating these redundancies, we are left with the following precise minimal *outcome explanation*: this sample is recognized as belonging to the class Iris Setosa because it has the properties

2.45 < petallength < 2.45
petalwidth < 0.7
7.05 < sepallength
2.25 sepalwidth < 2.65

CONCLUSIONS AND PERSPECTIVES

Explainable AI is a new direction aiming at the maturation of a field that has experienced a boost in particular because of its fancy heuristics and corresponding breakthroughs in specific applications like the AlphaGo program for the game Go. In this context, the typical concept of "explanation" is still comparatively weak. For example, highlighting the most important pixel for a certain image classification is not really a comprehensive explanation, but rather a hint, an indication that helps pinpoint situations where things went drastically wrong. In contrast we take a formal methods-based path, originally established in STTT,[5] where the concept of "explanation" is interpreted as a precise characterization of the considered phenomenon. Our illustration on how much information about the *how* and *why* can be extracted with exact methods from a random forest consisting of 100 trees indicates that such characterization may indeed turn out to be practical.

The concise class characterization has a particularly high application potential, e.g., when reversing the learned classification function for tailored product presentations in order to obtain an optimized customer list for a product campaign. Moreover, the size and therefore the comprehensibility of class characterization seem to hardly explode. In our example with only three classes, the model characterization ADD had more than 1100 nodes, while all the class characterization ADDs have less than 60 nodes, a size still within the range of a visual investigation.

Of course, these are first steps in a very ambitious new direction and it has to be seen how far the approach carries. Scalability will probably require decompositions methods, perhaps in a similar fashion as illustrated by the difference between model explanation and the considerably smaller class characterization. More work is needed also on techniques that aim at limiting the number of involved predicates.

Promising results reported in sttt2 lift the approach we illustrated from random forests to binary neural networks. They indicate that true explainability may well be in reach even for neural networks, on the formal methods way.

REFERENCES

1. [Online]. Available: https://www.lexalytics.com/lexablog/stories-ai-failure-avoid-ai-fails-2020
2. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/CervicalCheck_cancer_scandal
3. R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
4. "UCI machine learning repository: Iris data set," Retrieved 2017-12-01, [Online]: Available: archive.ics.uci.edu
5. F. Gossen and B. Steffen, "Algebraic aggregation random forests: Towards explainability and rapid evaluation" to be published.
6. R. I. Bahar, et al., "Algebraic decision diagrams and their applications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. IEEE Comput. Soc. Press*, 1993.
7. H. Hungar, B. Steffen, and T. Margaria, "Methods for generating selection structures, for making selections according to selection structures and for creating selection descriptions," USPTO Patent number: 9141708, Granted Sep. 22, 2015. [Online]. Available: https://patents.justia.com/patent/9141708

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

DEPARTMENT: CYBERTRUST

This article originally appeared in
Computer
vol. 52, no. 9, 2019

Disruptive Innovations and Disruptive Assurance: Assuring Machine Learning and Autonomy

Robin Bloomfield, *Adelard LLP and City University of London*
Heidy Khlaaf, Philippa Ryan Conmy, and Gareth Fletcher, *Adelard LLP*

Autonomous and machine learning-based systems are disruptive innovations and thus require a corresponding disruptive assurance strategy. We offer an overview of a framework based on claims, arguments, and evidence aimed at addressing these systems and use it to identify specific gaps, challenges, and potential solutions.

The advancement and adoption of machine-learning (ML) algorithms constitute a crucial innovative disruption. However, to benefit from these innovations within security and safety-critical domains, we need to be able to evaluate the risks and benefits of the technologies used; in particular, we need to assure ML-based and autonomous systems.

The assurance of complex software-based systems often relies on a standards-based justification. But in the case of autonomous systems, it is difficult to rely solely on this approach, given the lack of validated standards, policies, and guidance for such novel technologies. Other strategies, such as "driving to safety," that use evidence developed from trials and experience to support claims of safety in deployment are unlikely to be successful by themselves,[1,2] especially if the impact of security threats is taken into account. This reinforces the need for innovation in assurance and the development of an assurance methodology for autonomous systems.

Although forthcoming standards and guidelines will eventually have an important, yet indirect, role in helping us justify behaviors, we need further development of assurance frameworks that enable us to exploit disruptive technologies. In this article, we focus on directly investigating the desired behavior (e.g., the safety property or reliability) of a system through an argument- or outcome-based approach that integrates disparate sources of evidence, whether from compliance, experience, or product analysis. We argue that building trust and trustworthiness through argument-based mechanisms, specifically the claims, arguments, and evidence (CAE) framework (see "The Assurance Framework"), allows for the accelerated exploration of novel mechanisms that would lead to the quality advancement and assurance of disruptive technologies (see Figures S1 and S2 in the "The Assurance Framework" sidebar).

The key advantage of a claim-based approach is that there is considerable flexibility in how the claims are demonstrated since different types of arguments and evidence can be used as appropriate. Such a flexible approach is necessary when identifying gaps and challenges in uncharted territory, such as the assurance of ML-based systems. Indeed, CAE is commonly used in safety-critical industries (such as defense, nuclear, and medical) to assure a wide range of systems and devices and support innovation in assurance.

We are developing a particular set of CAE structures that is generically applicable and helps identify how to construct trustworthy ML-based systems by explicitly considering evidence of sources of doubt, vulnerabilities, and mitigations addressing the behavior of the system. In doing this, we not only assure and determine challenges and gaps in behavioral properties but also self-identify gaps within the assurance

Digital Object Identifier 10.1109/MC.2019.2914775
Date of publication: 27 August 2019

 Published by the IEEE Computer Society

framework itself. In the remainder of this article, we describe our systematic approach to identifying a range of gaps and challenges regarding ML-based systems and their assurance.

IDENTIFYING ASSURANCE CHALLENGES

The decision to trust an engineering system resides in engineering argumentation that addresses the evaluation and risk assessment of the system and the role of the different subsystems and components in achieving trustworthiness. Although previous abstractions, models, and relationships have been constructed in CAE for the assurance of traditional software systems, it is not clear if the said existing blocks are sufficient to provide compositional argumentation enabling trustworthiness in ML-based systems. For example, domain-specific abstractions and arguments may need to be developed in CAE to specifically target ML subcomponents.

To develop a detailed understanding of such assurance challenges, we use CAE to create an outline of an overall assurance case, proceeding from top-level claims, concerning an experimental autonomous vehicle and its social context, down to claims regarding the evaluation of subsystems, such as the ML model (Figure 1). The case study autonomous vehicle, as is typical with similar state-of-the-art vehicles, contains a heterogeneous mixture of commercial off-the-shelf (COTS) components, including image recognition, lidar, and other items. Apportioning the trustworthiness, dependability, and requirements of each component to consider the real-time and safety-related nature of the system is challenging. In traditional safety-critical engineering, there would be diversity and defense in depth to reduce the trust needed in specific ML components; yet we do not know whether this is practicable for ML-based systems. Argumentation blocks may need to be further developed within CAE to determine how experimental data can allow for the comparison and assessment of diverse subsystems' contribution to defense in depth. This, in turn, can also inform future architectures of autonomous systems.

Beyond the study of the applicability of CAE to assure ML-based systems, the lens of the assurance case is used to identify gaps and challenges regarding techniques and evidence aimed at justifying desired system behaviors. This is further informed by a review of literature, a case study-based assessment of the experimental vehicle, and an investigation of our industry partners' development processes to assess the current state of the vehicle and the short- to medium-term future vision of its use case (approximately two years). To see how and whether security is addressed in the product lifecycle, we used the new U.K. Code of Practice PAS 11281, *Connected Automotive Ecosystems—Impact of Security on Safety*.[4]

In the subsequent sections, we discuss some of the gaps identified regarding technical capabilities that may enable trust of system behaviors. We highlight three areas: requirements, security, and verification and validation (V&V). There are also issues of ethics, advanced safety analysis techniques, defense in depth, and diversity modeling that we do not address.

GAPS AND CHALLENGES

Innovation, trust, and requirements

There is a need to address the realities of the innovation lifecycle and progressively develop requirements, including those for trustworthiness and assurance. In this innovation approach, the vehicle is gradually developed from a platform to trial technologies to the final product (Figure 2). There is an assurance gap in that, when analyzing how much the technologies need to be trusted, there must be an articulated vision of what they will be used for. If the vision of how something will be used is not clearly formulated, we cannot assess how much we need to trust it or what the risks are.

THE ASSURANCE FRAMEWORK

The claims, arguments, and evidence (CAE) framework supports the structured argumentation for complex engineering systems. It is based on an explicit claim-based approach to justification and relates back to earlier philosophical work by Wigmore[S6] and Toulmin[S7] as well as drawing on theory and empirical research in recent years in the safety and assurance cases areas (see John Rushby's analysis[S4] for a rigorous review of the field).

At the heart of the CAE framework are three key elements (Figure S1). Claims are assertions put forward for general acceptance. They are typically statements about a property of the system or some subsystem. Claims asserted as true without justification are assumptions, and claims supporting an argument are subclaims. Arguments link evidence to a claim, which can be deterministic, probabilistic, or qualitative. They consist of "statements indicating the general ways of arguing being applied in a particular case and implicitly relied on and whose trustworthiness is well established" (see Toulmin[S7]), together with validation of any scientific laws used. In an engineering context, arguments should be explicit. Evidence serves as the basis for justification of a claim. Sources of evidence can include the design, the development process, prior experience, testing (including statistical testing), or formal analysis.

In addition to the basic CAE concepts, the framework consists of CAE blocks that provide a restrictive set of common argument fragments and a mechanism for separating inductive and deductive aspects of the argumentation (Figure S2). These were identified by empirical analysis of actual safety cases.[S5] The blocks are as follows:

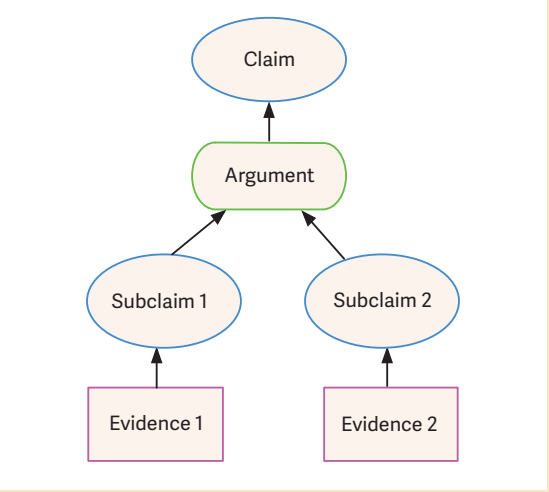


FIGURE S1. The CAE notation.

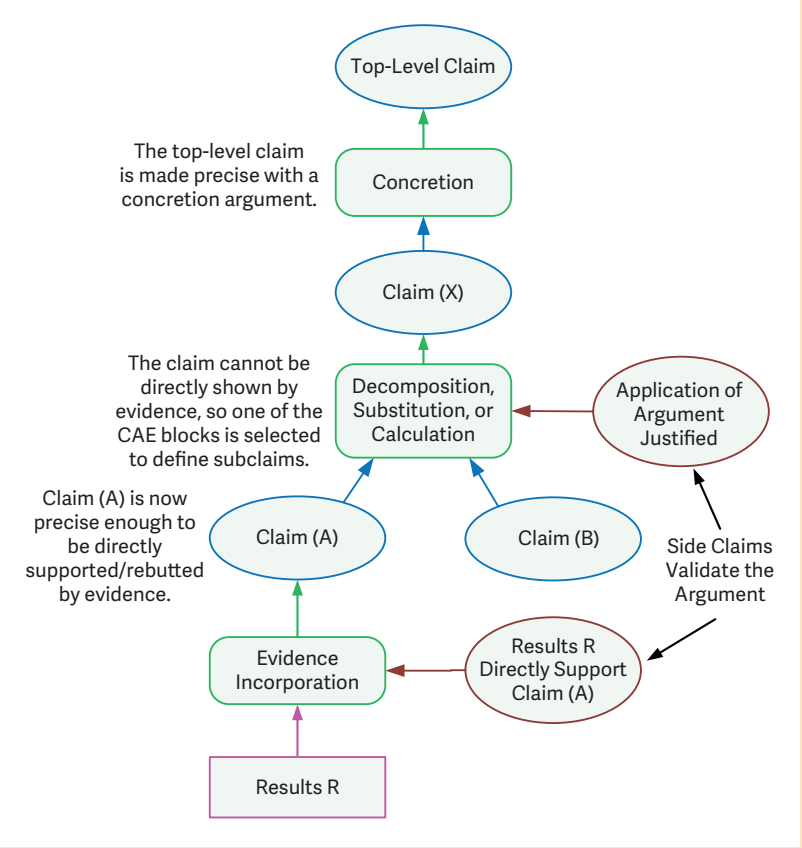


FIGURE S2. An example of CAE block use.

THE ASSURANCE FRAMEWORK (CONT.)

- *Decomposition*: There is partition of some aspect of the claim, or divide and conquer.
- *Substitution*: A claim about an object is refined into a claim about an equivalent object.
- *Evidence incorporation*: Evidence supports the claim, with an emphasis on direct support.
- *Concretion*: Some aspect of the claim is given a more precise definition.
- *Calculation or proof*: Some value of the claim can be computed or proved.

The framework also defines connection rules to restrict the topology of CAE graphical structures. The use of blocks and associated narrative can capture challenges, doubts, and rebuttals and illustrates how confidence can be considered as an integral part of the justification.

The basic concepts of CAE are supported by an international standard,[S1] IAEA guidance,[S3] and industry guidance.[S2] To support CAE, a graphical notation can be used to describe the interrelationship of evidence, arguments, and claims.[S3,S5] In practice, top desirable claims, such as "the system is adequately secure," are too vague or are not directly supported or refuted by evidence. Therefore, it is necessary to create subclaim nodes until the final nodes of the assessment can be directly supported or refuted by evidence.

REFERENCES

S1. *Systems and Software Engineering—Systems and Software Assurance, Part 2: Assurance Case*, ISO/IEC 15026-2:2011, 2011.

S2. P. G. Bishop and R. E. Bloomfield, "A methodology for safety case development," in *Industrial Perspectives of Safety-Critical Systems: Proceedings of the Sixth Safety-Critical Systems Symposium, Birmingham 1998*, F. Redmill and T. Anderson, Eds. London: Springer-Verlag, 1998, pp. 194–203.

S3. International Atomic Energy Agency, "*Dependability assessment of software for safety instrumentation and control systems at nuclear power plants*," IAEA Nuclear Energy Series NP-T-3.27, 2018. [Online]. Available: https://www-pub.iaea.org/books/IAEABooks/12232/Dependability-Assessment-of-Software-for-Safety-Instrumentation-and-Control-Systems-at-Nuclear-Power-Plants

S4. J. Rushby, "*The interpretation and evaluation of assurance cases*," SRI Int., Menlo Park CA, Tech. Rep. SRI-CSL-15-01, July 2015.

S5. R. Bloomfield and K. Netkachova, "Building blocks for assurance cases," in *Proc. IEEE Int. Symp. Software Reliability Engineering Workshops (ISSREW)*, Nov. 2014, pp. 186–191. doi: 10.1109/ISSREW.2014.72.

S6. J. H. Wigmore, "The science of judicial proof," *Virginia Law Rev.*, vol. 25, no. 1, pp. 120–127, Nov. 1938. doi: 10.2307/1068138.

S7. S. E. Toulmin, *The Uses of Argument*. Cambridge Univ. Press, United Kingdom. 1958.

This is particularly important for security and systemic risks, where the scale and nature of the deployment (such as a key part of an urban transport system) will lead to more onerous requirements that have to be reflected in the earlier technology trials and evaluations. Alternatively, more agile approaches would be to progressively identify these trust requirements as the innovation proceeds. But this might lead to solutions that do not scale and, in the extreme, could not be deployed. We believe that the innovation lifecycle subsequently presented is typical for many players in the industry and will be increasingly adopted as the ML components become more productized.

Security

Security is a fundamental and integral attribute of the technical themes of the project, in the requirements, V&V, and assurance research. While the requirements of the new PAS 11281 Code of Practice may be met in a mature implementation of the vehicle being studied, on the whole, the security will be challenging for industry, and advice must be provided on partial and project-specific implementation of the PAS that allows for maturity growth.

The security aspects need to be integrated into the entire lifecycle: systems are not safe if they are not secure. This applies to the vehicle as a whole as well as to the ML subsystems; most ML systems have not been designed with a systematic attention to security.[10] The PAS clauses address the following areas and are equally applicable to the vehicle and its components:

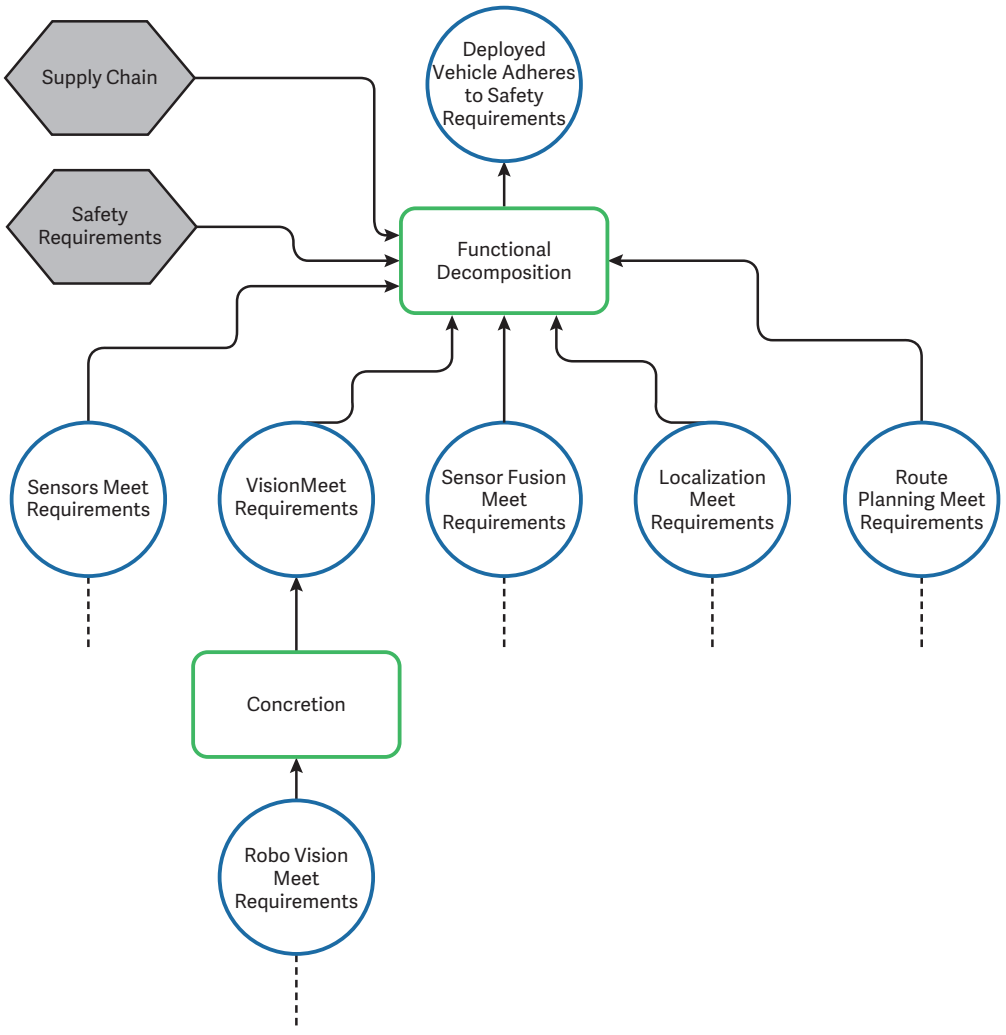


FIGURE 1. A high-level example of an assurance subcase in CAE.

1. security policy, organization, and culture
2. security-aware development process
3. maintaining effective defenses
4. incident management
5. secure and safe design
6. contributing to a safe and secure world.

As we noted previously, the deployment of autonomous technologies may follow an innovation lifecycle that first focuses on functionality and seeks to progressively add additional assurance and security. This will make the development of the assurance and safety cases and associated security and safety risk assessments particularly challenging. From our experience, we recommend the following:

1. Explicitly define the innovation cycle and assess the impact and feasibility of adding assurance and security.
2. Address the approach to security-informed safety at all stages of the innovation cycle. If safety, security, and resilience requirements are largely undefined at the start of the innovation cycle, the feasibility of progressively identifying them during the cycle should be assessed, together with the issues involved in evolving the architecture and increasing the assurance evidence.
3. Apply PAS 11281 to systematically identify the issues. Use a CAE assurance case framework and map PAS clauses to this to provide a

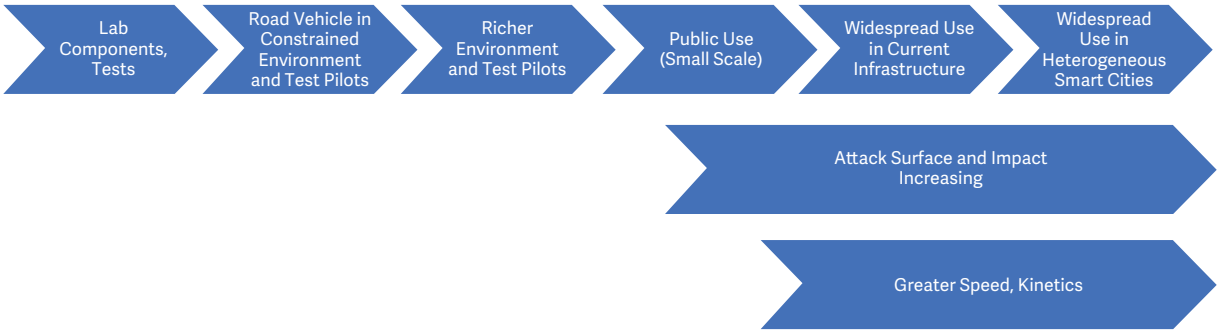


FIGURE 2. The typical stages of development from innovation to products.

systematic approach to applying the PAS.

4. Consider a partial and project-specific implementation of the PAS to meet the innovation cycle.
5. Collect experience in developing a security-informed safety case and integrating security issues into the safety analyses needed to implement the PAS.

V&V

We use the assurance case in CAE top-down to identify the claims we wish to support and bottom-up to evaluate the evidence that could be provided by them and, hence, systematically assess gaps, challenges, and solutions. This is shown schematically in Figure 3. As part of this analysis, we assessed state-of-the-art formal methods for autonomous systems and observed that their maturity and applicability are lacking for sufficiently justifying behavioral and vulnerability claims.

Consider the issue of adversarial attacks and perturbations,[5,6] which has been particularly challenging with regard to the robustness of ML algorithms. Verification researchers have focused on the property of pointwise robustness, in which a classifier function f' is not robust at point x if there exists a point y within η such that the classification of y is not the same as the classification of x. That is, for some point x from the input, the classification label remains constant within the neighborhood η of x, even when small-value deltas (i.e., perturbations) are applied to x. A point x would not be robust if it were at a decision boundary, and adding a perturbation would cause it to be categorized in the next class. Generally speaking, the idea is that a neighborhood η should be reasonably classified as the given class.

However, proposed pointwise robustness verification methods[8–10] suffer from the same set of limitations.

- There is a lack of clarity on how to define meaningful regions η and manipulations.
  - The neighborhoods surrounding a point x that are currently used are arbitrary and conservative.
- We cannot enumerate all x points near which the classifier should be approximately constant; that is, we cannot predict all future inputs.

Furthermore, researchers have been unable to find compelling threat models that required perturbation indistinguishability,[12] and it has been demonstrated that l_p, which defines the neighborhood region η, is a poor proximity for measuring what humans actually see.[13] Finally, adversarial perturbations can be achieved by much simpler attacks that do not require ML algorithms (e.g., covering a stop sign). Thus, the extent to which these techniques can provide us with any level of confidence is not very high.

Other verification techniques[7,9] aim to verify more general behaviors regarding ML algorithms, instead of just pointwise robustness. Such techniques require functional specifications, written as constraints, to be fed into a specialized linear-programming solver to be verified against a piecewise linear constraint model of the ML algorithm. However, the generalization of these algorithms is challenging, given the requirement of well-defined and bounded traditional system specifications, devoid of specifications regarding the

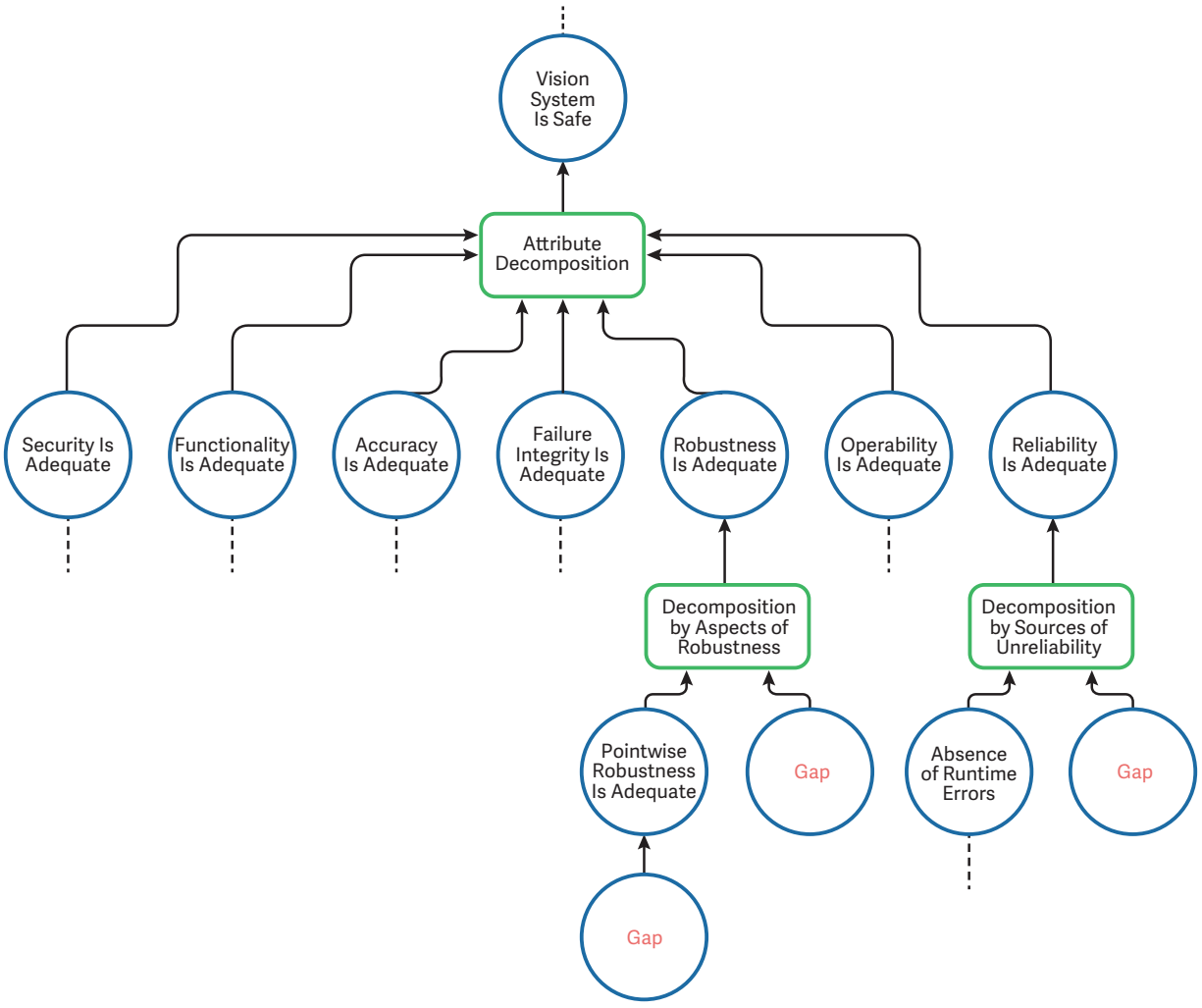


FIGURE 3. The use of CAE to assess V&V gaps.

behavior of the ML algorithm itself. These techniques are thus applicable to well-specified deterministic ML algorithms and cannot be applied to perception algorithms, which are notoriously difficult to specify, let alone verify.

Apart from the ML algorithm, the assurance of the non-ML supporting components of an autonomous system is challenging, given that the use of COTS or open source components leads to uncertain provenance. Errors within non-ML components can propagate and affect the functionality of the ML model.[14] It is, therefore, important to explore how traditional V&V methods—in particular, static analysis of C code—can provide assurance for the larger ML system, offering confidence beyond the component level. In the following, we provide a preliminary list of results from analyzing YOLO, a commonly used open source ML vision software, and a number of different run-time errors that were identified:

- a number of memory leaks, such as files opened and not closed, and temporarily allocated data not freed, leading to unpredictable behavior, crashes, and corrupted data
- a large number of calls to free where the validity of the returned data is not checked [this could lead to incorrect (but potentially plausible) weights being loaded to the network]
- potential "divide by zeros" in the training code (this could lead to crashes during online training, if the system were to be used in such a way)
- potential floating-point divide by zeros, some of

which were located in the network cost calculation function (as noted above, this could be an issue during online training).

These errors would be applicable only to languages such as C and C++. Not all errors would be relevant to a language such as Python, used in the implementation of numerous ML libraries and frameworks, as the semantics and implementation of the language itself do not enable overflow/underflow errors, defined by Hutchison et al.[14] However, Python is a dynamically typed language, bringing about a different set of program errors not exhibited by statically typed languages (such as type errors). Unfortunately, no static analysis techniques or tools exist to allow for the analysis of Python code. Furthermore, it is unclear how potential faults arising from dynamic languages could affect the functionality of an ML model itself. This is a large gap within the formal methods field that needs to be addressed immediately, given the deployment of autonomous vehicles utilizing Python.

There is a need for disruptive innovation in the assurance of autonomous and ML-based systems. We provided a summary of the outcome-focused, CAE-based framework we are evolving to address these systems and used it to identify specific gaps and challenges; we also discussed some solutions. We demonstrated the feasibility of deploying the best of existing work (e.g., advanced static analysis techniques) and identified the need for new approaches.

Overall, there is a need for stronger evidence and techniques to assure the dependability of ML components and for autonomous systems as a whole. Indeed, there is common good in sharing techniques and strategies regarding development lifecycles, diversity, security, and V&V algorithms in sufficient detail for independent analysis and research. We hope to play our part in this by sharing our generic developed assurance case and providing, in the public domain, the more detailed report this article is based on. If we can achieve our goal of disruptive assurance, this can have a positive impact on innovation in a wide range of industries and technologies, not just ML-based ones.

ACKNOWLEDGMENTS

This article discusses work undertaken within the Towards Identifying and closing Gaps in Assurance of autonomous Road vehicleS (TIGARS) project. The project is a collaboration between Adelard, Witz, the City University of London, the University of Nagoya, and Kanagawa University. This work is partially supported by the Assuring Autonomy International Programme, a partnership between Lloyd's Register Foundation and the University of York. We acknowledge the additional support of the U.K. Department for Transport.

IT IS UNCLEAR HOW POTENTIAL FAULTS ARISING FROM DYNAMIC LANGUAGES COULD AFFECT THE FUNCTIONALITY OF AN ML MODEL ITSELF.

REFERENCES

1. N. Kalra and S. Paddock, *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* Santa Monica, CA: RAND Corporation, 2016. [Online]. Available: https://www.rand.org/pubs/research_reports/RR1478.html
2. P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE Int. J. Transp. Safety*, vol. 4, no. 1, pp. 15–24, 2016.
3. R. Bloomfield, P. Bishop, E. Butler, and R. Stroud, "Security-informed safety: Supporting stakeholders with codes of practice [Cybertrust]," *Computer*, vol. 51, no. 8, pp. 60–65, Aug. 2018.
4. *Connected Automotive Ecosystems—Impact of Security on Safety*, British Standards Institution, PAS 11281, 2018.
5. C. Szegedy et al., Intriguing properties of neural networks. 2013. [Online]. Available: https://arxiv.org/abs/1312.6199
6. I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learning Representations—Computational and Biological Learning Society*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6572v3
7. L. Pulina and A. Tacchella, "An abstraction-refinement approach to verification of artificial neural networks," *Computer Aided Verification, CAV* 2010, Lecture Notes in Computer Science, vol 6174, T. Touili, B. Cook, and

P. Jackson, Eds. Berlin: Springer, pp. 243–257.
8. X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, Safety verification of deep neural networks. 2016. [Online]. Available: https://arxiv.org/abs/1610.06940
9. G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, Reluplex: An efficient SMT solver for verifying deep neural networks. 2017. [Online]. Available: https://arxiv.org/abs/1702.01135
10. N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, Towards the science of security and privacy in machine learning. 2016. [Online]. Available: https://arxiv.org/abs/1611.03814
11. W. Ruan, X. Huan, and M. Z. Kwiatkowska, Reachability analysis of deep neural networks with provable guarantees. 2018. [Online]. Available: http://arxiv.org/abs/1805.02242
12. J. Gilmer, R. Adams, I. Goodfellow, D. Andersen, and G. Dahl, Motivating the rules of the game for adversarial example research. 2018. [Online]. Available: https://arxiv.org/abs/1807.06732
13. Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, 2009.
14. C. Hutchison et al., "Robustness testing of autonomy software," in *Proc. IEEE/ACM 40th Int. Conf. Software Engineering: Software Engineering in Practice Track*, Gothenburg, Sweden, May 27–June 3, 2018, pp. 276–285.

ROBIN BLOOMFIELD is with Adelard LLP and the City University of London. Contact him at reb@adelard.com or reb@csr.city.ac.uk.

HEIDY KHLAAF is with Adelard LLP. Contact her at hak@adelard.com.

PHILIPPA RYAN CONMY is with Adelard LLP. Contact her at pmrc@addelard.com.

GARETH FLETCHER is with Adelard LLP. Contact him at gtf@adelard.com.

EDITOR: Christof Ebert, Vector Consulting Services, christof.ebert@vector.com

COLUMN: SOFTWARE TECHNOLOGY

This article originally appeared in
IEEE Software
vol. 36, no. 5, 2019

Validation of Autonomous Systems

Christof Ebert and Michael Weyrich

FROM THE EDITOR

Autonomous systems are widely used. Yet, for lack of transparency, we are increasingly suspicious of their decision making. Traditional validation, such as functional testing and brute force, won't help, due to complexity and cost. To achieve dependability and trust we need dedicated, intelligent validating techniques that cover, for instance, dynamic changes and learning. Michael Weyrich and I provide industry insights into validating autonomous systems. I look forward to hearing from both readers and prospective authors about this article and the technologies you want to know more about. —*Christof Ebert*

Society today depends on autonomous systems, such as intelligent service systems, self-driving trains, and remote surgeries.[1] The ultimate validation of the Turing test is that we often do not recognize autonomous systems. This growing usage poses many challenges, such as how to provide transparency, which rules or learning patterns are applied in a complex situation, and if these rules are the right ones. Validation is the key challenge, of which we will provide an overview in this article.

With machine learning and continuous over-the-air upgrades and updates, a core tenant of any quality strategy is continuous verification and validation. Corrections and changes must be deployed in a fluid and continuous scheme, reliably over the air. We will face future scenarios where software-driven systems, and maybe whole infrastructures, must not be started if they do not include all of the latest software upgrades. Automobiles and manufacturing processes that are safety critical fall into that category. Even more demanding are medical devices, which must provide a hierarchical software assurance because there is no room for failure.

Autonomous systems have multiple complex interactions with the real world. They perceive and act in the environment, based upon the reflections of an intelligent control system, and they have an increasing impact on our lives as they implement and execute high-level tasks without detailed programming or direct human control. Unlike automated systems, which execute a carefully engineered sequence of actions, they are self-governing their course of action to independently achieve their goals.

Figure 1 indicates the five steps from automation to autonomy as we know them from human learning, where we advance from novice to expert. Those steps exemplify the progress of a simple and "assisted behavior" from low-level sensing and control toward "full cognitive systems" with a very high degree autonomy. Automated systems are gradually enhanced to develop a skilled behavior along with enhanced mission planning and control and execution capabilities that will eventually lead to the full cognitive actions of an autonomous system. It is expected that

Digital Object Identifier 10.1109/MS.2019.2921037
Date of publication: 20 August 2019

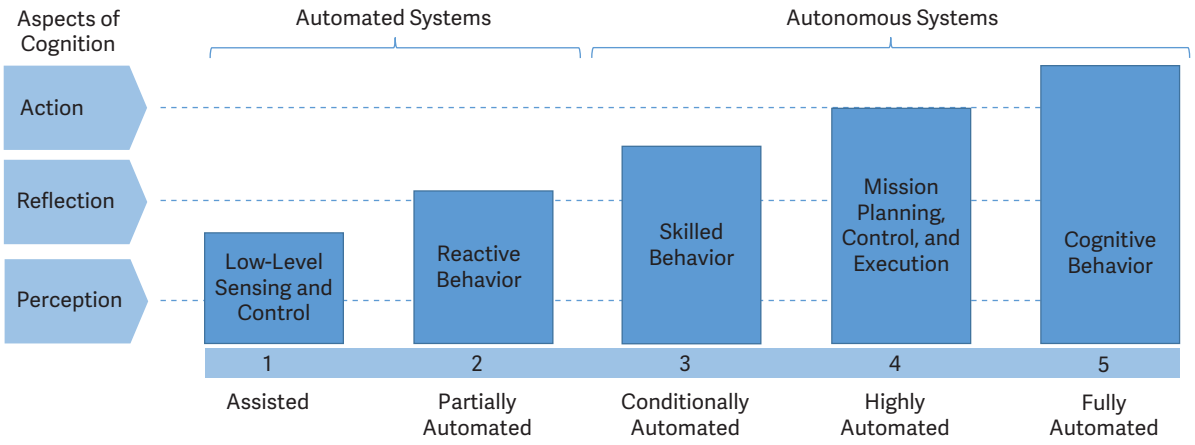


FIGURE 1. The five steps from automation to autonomy.

an intelligent behavior can be identified by acquiring knowledge and understanding, which entails system functionalities such as perception, reflection, and action in terms of a cognition.

A completely autonomous car on level 5 is supposed to drive with no human intervention, even in dire situations. This implies that the car must have intelligence on par with or better than humans to handle not just regular traffic scenarios but unexpected ones. Although several players, such as Google and Uber, are granted permission to operate their self-driving services, deadly incidents put our faith in these cars to a test.[2] It is quite apparent that existing validation measures aren't enough.[3] We need new test methods that can envision fatal traffic situations that humans haven't encountered yet. In addition, testing cannot simply be isolated to the final development stages. It must be part of every phase in the product lifecycle. A sensible engineering process must be adopted in the development of autonomous cars that lays enough emphasis on testing and validation.

Unlike an automated system, which cannot reflect on the consequences of its actions and cannot change a predefined sequence of activities, an autonomous system is meant to understand and decide how to execute tasks based on its goals, skills, and a learning experience. While contemplating the deficiencies of autonomous systems, we should acknowledge that humans have natural limits, in terms of processing speed, repeatability of tasks, handling complexity, and so forth. In fact, in aerospace, we already trust autonomous flying, and for automotive applications, automation is forecast to reduce deadly accidents by 90%.[4] Autonomous systems can become an aid in the future, in areas such as automated and autonomous driving, flying, and production robotics.

VALIDATION OF AUTONOMOUS SYSTEMS

Autonomous systems provide efficiency and safety as they relieve human operators from tedious manual activities. For instance, the widespread use of self-driving cars could eliminate as much as 50% of a person's daily commuting time.[4] As exciting as this may sound, the question "Can we trust the autonomous systems?" will grow for years to come. Public confidence in autonomous systems depends heavily on algorithmic transparency and continuous validation.

Recently, we have seen several dramatic accidents, such as an automated car misinterpreting a white truck as a white cloud, and another one overlooking pedestrians on a road, thus, killing people. One spectacular accident happened when an automated vehicle continued along while its driver had a heart attack and could not supervise it. Within a few seconds, the automated vehicle killed a mother and child as it tried to avoid colliding with a tree. Hitting the tree might have killed the driver, but innocent people in the surrounding environment would have been safe.

There are many open questions about the validation of autonomous systems: How do we define reliability? How do we trace back decision making and judge it after the fact? How do we supervise these systems? How do we define liability in the event of failure?

Figure 2 provides an overview of validation technologies for autonomous systems. We distinguish, horizontally, the transparency of the validation. *Black box* means that we have no insight to the method and coverage, while *white box* denotes transparency. The vertical axis classifies the degree to which we can automate validation techniques and, for instance, facilitate regression strategies through software updates and upgrades.

Let us look at traditional testing techniques (see Figures 1 and 2) and evaluate their behaviors. Table 1 provides the complete evaluation of static and dynamic validation technologies for autonomous systems. Negative requirements (such as safety and cybersecurity) are typically implied and not explicitly stated in the system specifications.[5] The following sections explain how these methods are applied to validate autonomous cars.

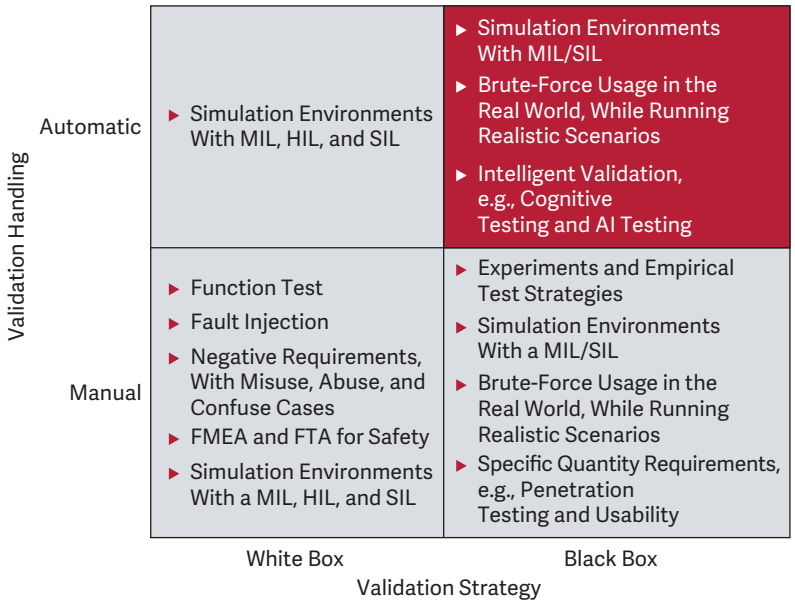


FIGURE 2. The validation technologies for autonomous systems. FMEA: failure mode and effects analysis; FTA: fault tree analysis; AI: artificial intelligence; MIL: model in the loop; HIL: hardware in the loop; SIL: software in the loop.

Fault Injection

Fault injection techniques make use of external equipment to insert faults into a target system's hardware, with or without direct contact. By having direct contact, faults, such as forced current addition, forced voltage variations, and so forth, can be injected to observe the behavior of the system. Faults can be introduced without making physical contact by using methods such as heavy-ion radiation, exposure to electromagnetic fields, and so on. Such fault injections can cause bit flips, hardware failure, and similar events that are not tolerated in safety-critical systems.

Functionality-Based Testing

Functionality-based test methods categorize the intelligence of a system into three classes: 1) sensing, 2) decision, and 3) action functionalities. The idea behind such methods is that an autonomous vehicle should be able to retrieve various functionalities for a given task analogous to human beings. For example, a vehicle should be able to recognize other cars and trucks, pedestrians, and so forth for vision-based functionality. Combinations of these recognized objects can act as inputs to decision functionality, and several decisions can lead to actions. Functionality-based testing breaks down the scenarios into various operational components that can be tested individually.

Hardware in the Loop

Although simulation tries to encapsulate the real world as closely as possible, inherent limitations invariably create a void between the two. Hardware in the loop (HIL) closes this gap a little by using physical components for certain aspects of simulation. For example, a camera model in a simulation technique can be replaced by an actual camera. The input to the camera can be fed by means of a computer screen where videos of various real-time traffic conditions are played to validate the behavior of car. A more advanced technique has been proposed for autonomous systems that are tested by robots, for instance, vehicle HIL, where the simulated vehicles in traffic have been replaced by moving robots. This has the advantage that, in addition to the camera, radar and lidar hardware can be tested using HIL.

TABLE 1. The evaluation of validation technologies for autonomous systems

Method	Characteristics	Tool support and technologies	Coverage	Regression strategy	Strengths	Weaknesses	Effectiveness	Efficiency
Modeling and simulation environments with SIL, HIL, and MIL	Static and dynamic	Model checker, e.g., MATLAB, dSPACE, Vector VT System, NovaCarts, Vires, PreScan	0	Repeat impacted scenarios (low efficiency)	• Reduces validation costs • Decouples hardware and software development	• Brute force, for high coverage • Requires a large amount of computation power • Tests only for known scenarios • Scenario banks are not comprehensive to validate autonomous systems • Intransparent dependencies	0	0
Function test	Dynamic, all functions	Modeling tool for functional abstraction, with unit test tools (for example, JUnit and PHPUnit) Dedicated test environments for stub generation	0	Repeat functional test cases for impacted operations	• Tests all AI aspects: sensing, decision making, and action taken • Validates all the functional requirements	• Insufficient to validate complete systems	+	+
Integration test	Dynamic	Test suites, test management, combinatorial tools (such as AETG and Citrus and so forth)	0	Regenerate test cases	• Tests component integration	• Large number of interfaces; E to miss some links • Fault localization is difficult	+	+
Fault injection	Static, for residual defect estimation	Test environment and defect modeling, e.g., beSTORM, Security Innovation	–	Introduce few selected defects	• Provides an estimate of residual defects and coverage • Exposes weaknesses, enabling designers to strengthen vulnerabilities	• Requires a concrete understanding of underlying system architecture and behavior	+	–
Negative requirements, with misuse, abuse, and confuse cases	Static, specifically for safety, security, and usability	Directly modeled and traced with requirements tools, e.g., DOORS, Visure, PTC, PREEvision, Enterprise Architect, HP ALM	0	Reuse situational negative cases	• Good for identifying scenarios to be avoided • Formalizes nonfunctional requirements • Strengthens system security	• Difficult to set up systematically • The tests cases do not necessarily cover all possible negative cases	+	+

Continued

TABLE 1. The evaluation of validation technologies for autonomous systems (cont.).

Method	Characteristics	Tool support and technologies	Coverage	Regression strategy	Strengths	Weaknesses	Effectiveness	Efficiency
FMEA and FTA	Static, specifically for safety-critical systems	FMEA worksheets, component abstractions, and reuse library	0	Retest for the changed components	• Well established for safety and security (attack tree) • Enables designers to foresee system interface failures	• Depends heavily on human knowledge • Labor intensive	+	+
Experiments and empirical test strategies	Empirical test generation for load test, performance, thermal, etc.	Experiment specific test tools, such as Parasoft DTP, EggPlant, Thermal imager, etc.	+	Repeat the test strategies for changed function	• Relatively easier to frame the test cases • Covers a wide range of electrical systems	• Depends heavily on human knowledge • Labor intensive • Very little or no test automation	+	0
Specific quality requirements tests, for instance, penetration testing and fuzzing	Dynamic, specifically for quality requirements	Dedicated test tools, for example, automatic fuzzing extensions, OWASP ZAP, Vega, and so on	–	Retest for impacted components	• Well established for security • Effective in ensuring that the system meets known quality requirements	• Often insufficient to validate complete system security and safety	–	+
Brute-force usage in the real world while running realistic scenarios	Dynamic, for ensuring situational coverage	Recording and replay with actual scenario libraries with data loggers from various sensor systems, e.g., Tecnomatix, CarMaker, EB Assist, CANape	0	Repetition (low efficiency)	• Closest to real world and thus highly effective • Validates all systems at once • Comprehensive view and coverage • Standardizes scenario storage format and tagging	• High effort for coverage • Unclear coverage • Most of the test cases are redundant • Intransparent situational coverage	+	–
Intelligent validation, for instance, cognitive and AI testing	Dynamic test generation and selection depending on situation and environment	Machine-learning frameworks such as Tensorflow, Apache Spark, and so on; open data sets, such as nuScenes	+	Reuse generated test cases from the dependency database	• Improved transparency • Automatically considers dependencies to external environment and internal functions • Automates major part of test procedure • Standardizes scenario storage format and tagging • Sharing test scenarios across V-Model abstraction levels	• High effort to set up AI-based test environment • Needs large computation power • Growing discipline, that is, not many methods and tools available	+	+

Vehicle in the Loop

Human interaction can have a drastic influence on the behavior of partially automated cars. The methods specified earlier fail to account for this reality. In vehicle-in-the-loop simulations, real cars are used, though in a safe environment. A driver is shown simulated feeds of the external environment to capture his interaction with the car. The car travels across a ground devoid of obstacles, simulating inertial effects and simultaneously responding to the external feed. The greatest advantage to this method is safety: Since there are no real obstructions involved, no harm will be incurred by the test drivers, even if they encounter dangerous situations.

Simulators

Simulators are closed, indoor cubicles that act as substitutes for physical systems. They can replicate the behavior of any system by using hardware and a software model. The behavior of a driver can be captured by immersing him a replicated external environment. Since simulators employ hydraulic actuators and electric motors, the inertial effects they generate feel nearly the same as the real-life version. They are used for robots in industrial automation, surgery planning in medicine, and railway and automotive applications.

Brute Force

Nothing can come closer to the real world than the real world itself. This is perhaps the final validation phase, where a completely ready system is physically driven onto roads with actual traffic. The sensor data are recorded and logged to capture behavior in critical situations. They are analyzed to accommodate and fine-tune the system according to everyday scenarios. The challenge in this stage, however, lies in the sheer amount of test data that are generated. A stereo video camera, alone, generates 100 GB of data for every kilometer driven. In such situations, big data analysis becomes extremely important.

Intelligent Validation Techniques

Intelligent validation techniques tend to automate the complete testing process or certain aspects of testing. This eliminates the potential errors associated with manual derivations of test cases, since humans may fail to recognize and think about certain scenarios. It also eradicates the enormous amount of time that needs to be invested to obtain the test cases. The "Intelligent Testing" section summarizes some approaches that attempt to derive such validation techniques.

Truly transparent validation methods and processes assume the utmost relevance and will be challenged by the progress of technology through the five steps toward autonomous behavior that are sketched in Figure 1. Although they are still relevant, traditional validation methods aren't enough to completely test the growing complexity of autonomous cars. Machine learning, with situational adaptations and software updates and upgrades, demands novel regression strategies. Figure 2 provides a map of the different testing techniques.

INTELLIGENT TESTING

With AI and machine learning, we need to satisfy algorithmic transparency. For instance, what are the rules, in a neural network that is obviously no longer algorithmically tangible, to determine who gets a credit or how an autonomous vehicle might react with several hazards at the same time? Classic traceability and regression testing will certainly not work. Future verification and validation methods and tools will include more intelligence based on big data exploits, business information, and the processes' ability to learn about and improve software quality in a dynamic way.[4]

A key question concerns which way AI can support the process of validation. Obviously, there are many AI approaches, ranging from rule-based systems, fuzzy logic,[6] and Bayesian nets to the multiple neural network approaches to deep learning. However, the process of validating an autonomous system is multilayered and rich in detail. Various levels of validation testing can be distinguished, such as the systems level, the components, and the modules.

The potential for intelligent testing is manifold. On a system level, there are questions about which test cases must be executed and to what extent. This means that intelligent validation is required to help with the selection and even the creation of test cases. A first step in that direction would be an assistance functionality that helped to identify priorities in an existing set of cases. As a result, a validation expert

COGNITIVE TESTING FOR AUTONOMOUS SYSTEMS

In our industrial projects, we often face the challenge of how systems can be validated, and safety assured, when they undergo a change during operation. Updates over the air are commonly used for functional modifications of software-based automated systems. Be they in manufacturing, automotive applications, or intelligent building, automated systems are mostly component based; they consist of multiple control units that are distributed. Each unit is in a certain location and has a specific functionality that it provides to the overall system.

Unwanted behavior and basic functional errors might occur somewhere in a distributed system because of an alteration elsewhere. How can such a system be safeguarded when changes in its components occur during runtime? How can safety and security certifications be maintained after a software modification happens within a single module?

A test certification requires an understanding of the effect of a change that is triggered somewhere in a software module and has impacts elsewhere. How can this interaction be deduced and the consequences for all modules be verified without testing the whole system again from scratch? The method presented here applies an artificial intelligence (AI) that can ascertain the consequences of an individual change in all the control units.

From our industry experience, we recommend a three-step approach to assess the impacts of software updates and upgrades (see Figure 3). First, the alteration in the system needs to be identified in terms of its origin in a module and its localization in the network. Second, a logical model of the overall system is composed to understand the impact on other modules. However, this model is distributed and needs to be automatically processed from the multiple submodules of the components that are available.

Third, a process of functional verification is required to check how the change is propagated and what it means with respect to potential malfunctions in the distributed system. This AI can be used to test and safeguard following a stepwise procedure for testing. It only requires the specification of the control models and their intended interaction with the other modules, upon which the overall functionality can be deduced and test certificates can be obtained on request.

would be able to test faster and with a better coverage of situationally relevant scenarios. On the level of a component or module,[7] testing it is also required to identify relevant cases. This can range from a simple support mechanism for how to feed a system with adequate inputs and checks on the outputs, to complex algorithms that automatically create test cases based on code or a user interface. Figure 3 provides an overview of intelligent testing as we ramp up for autonomous systems. Unlike brute force, intelligent testing considers the white-box and black-box dependencies and, thus, balances efficiency and effectiveness. See "Cognitive Testing for Autonomous Systems" for a concrete case study.

PERSPECTIVES

Verification and validation depend on many factors. Every organization implements its own methodology and development environment, based on a combination of several of the tools presented in this article. It is important not only to deploy tools but to build the necessary verification and validation competences. Too often we see solid tool chains but no tangible test strategies. To mitigate these purely human risks, software must increasingly be capable of detecting its own defects and failure points. Various intelligent methods and tools will evolve that can assist with smart validation of autonomous systems. However, even with the support of the smartest intelligent algorithms, the question remains how to build the public's trust that autonomous systems can be validated while considering ethical dilemmas, such as the accident when the mother and child were killed.

With the growing concern of users and policy makers about the impact of autonomous systems on our lives and society, software engineers must ensure that autonomy acts better than humans. Clearly, we

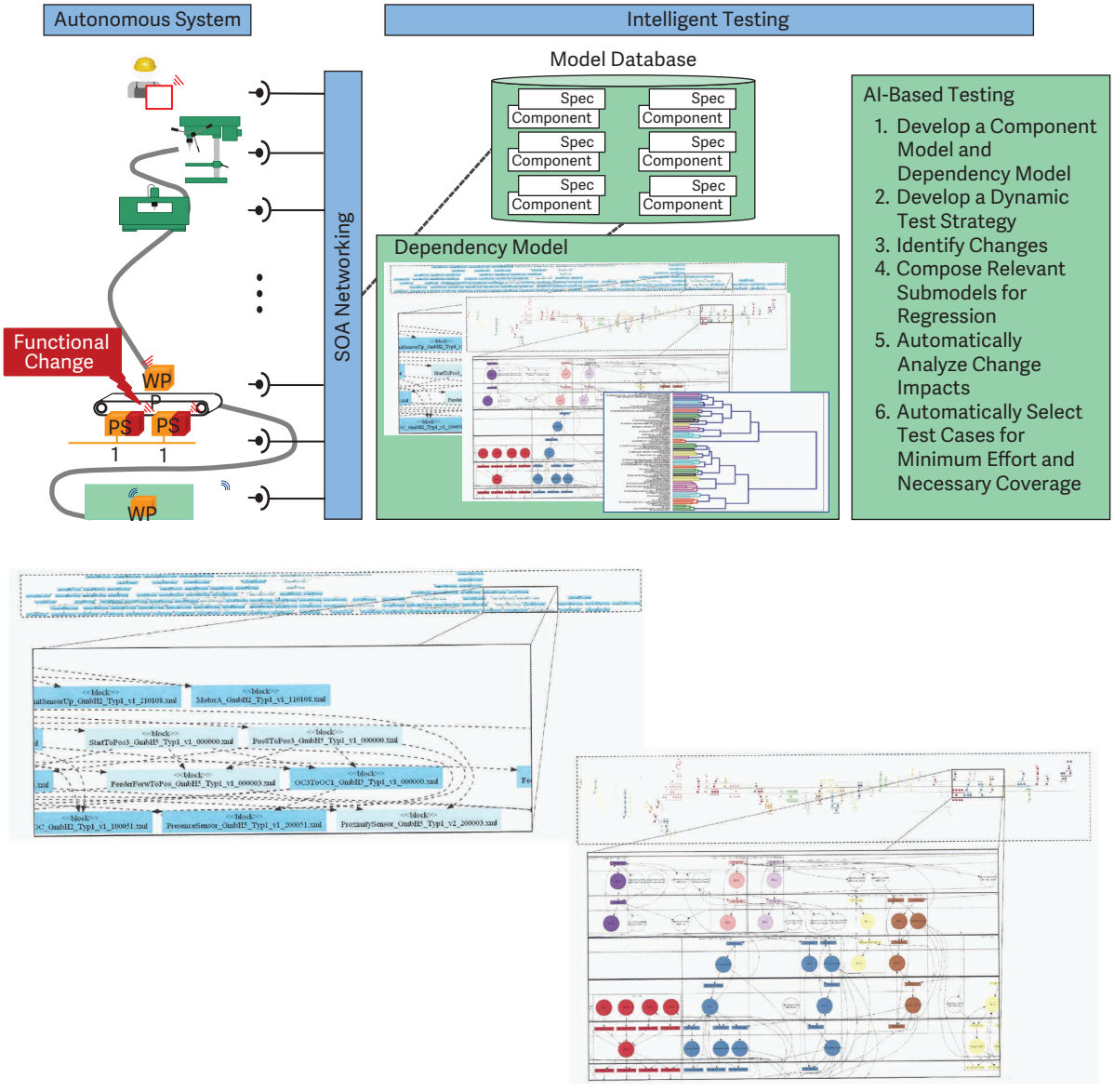


FIGURE 3. Intelligent testing for autonomous systems. SOA: service-oriented architecture; P: process; PS: production sensor; WP: work package.

are not talking about few percentage points. To build trust, we need a level of quality at least one order of magnitude higher than human-operated systems. It is, above all, a question of validation to achieve trust. Alan Turing, who was one of the first to consider AI in real life, remarked wisely, "We can only see a short distance ahead, but we can see plenty there that needs to be done." This remains true for a rather long transition period, and intelligent validation will play a pivotal role.

REFERENCES

1. M. Weyrich and C. Ebert, "Reference architectures for the Internet of Things," *IEEE Softw.*, vol. 33, no. 1, pp. 112–116, Jan.–Feb. 2016.
2. M. Santori and D. A. Hall. (2016). *Tackling the test challenge of next generation ADAS vehicle architecture*. National Instruments. Austin, TX. [Online]. Available: http://download.ni.com/evaluation/automotive/Next_Generation_ADAS_Vehicle_Architectures.pdf
3. M. Rodriguez, M. Piattini, and C. Ebert, "Software

verification and validation technologies and tools," *IEEE Softw.*, vol. 36, no. 2, pp. 13–24, Mar. 2019.
4. P. Gao, .H.-W. Kaas, D. Mohr, and D. Wee, (2016, Jan.) *Automotive revolution: Perspective towards 2030*. McKinsey & Co., New York. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/industries/high%20tech/our%20insights/disruptive%20trends%20that%20will%20transform%20the%20auto%20industry/auto%202030%20report%20jan%202016.ashx
5. *Road vehicles—Safety of the indented functionality*, International Organization for Standardization, 21448, 2019.
6. C. Ebert, "Rule-based fuzzy classification for software quality control," *Fuzzy Sets Syst.*, vol. 63, no. 3, pp. 349–358, May 1994. doi: 10.1016/0165-0114(94)90221-6.
7. A. Zeller and M. Weyrich, "Composition of modular models for verification of distributed automation systems," in *Proc. 28th Int. Conf. Flexible Automation and Intelligent Manufacturing (FAIM2018)*, Columbus, OH, 2018, pp. 870–877.

CHRISTOF EBERT is the managing director of Vector Consulting Services. He is on the IEEE Software editorial board and teaches at the University of Stuttgart, Germany, and the Sorbonne in Paris. Contact him at christof.ebert@vector.com.

MICHAEL WEYRICH is the director of the Institute of Industrial Automation and Software Engineering at the University of Stuttgart, Germany. Contact him at michael.weyrich@ias.uni-stuttgart.de.

This article originally appeared in
IEEE Annals of the History of Computing
vol. 42, no. 2, 2020

Queens of Code

Eileen Buckholtz, *Director, Queens of Code Project*

INTRODUCTION TO THE QUEENS OF CODE

Queens of Code is a women's technology history project—a collection of stories, experiences, and insights from women who worked in information technology at the National Security Agency (NSA) in the 1960s, 1970s, and 1980s. NSA's computing women programmed and managed the most sophisticated systems of their day and I was one of them. I started this project in 2018 to collect the stories of the agency's women technology pioneers and recognize their contributions because I believed that if we did not document these stories now while many of us are still living, our history would never be told. The National Cryptologic Museum and NSA's historians offered encouragement. I reached out to women I had worked with, and dozens signed up. Participants were asked to complete a detailed questionnaire and write their stories. All material had to be approved through NSA's prepublication review. We have been networking online for almost two years and have more than 75 women in the group. The goals for the project are recognition of the Queens of Code in the history of computing, expanding the understanding of how women worked in early computing, and inspiring more young women to pursue STEM careers. We are sharing our stories in presentations, articles, and interviews.

Because these NSA women's jobs were often top secret and they worked on the most sensitive national security programs, they could not discuss what they did, even with their families. In many cases, they could not even confirm they worked for NSA. They and their computing activities have been, for practical purposes, a secret for more than 50 years.

Women have always been in the workforce—although their contributions to science have often gone unrecognized. In the 20th century, women worked for the U.S. government and military, not just in clerical, nursing, and other "women's" positions, but in specialized technical fields such as cryptology, mathematics, and computing. The U.S. military during World War II actively recruited educated and talented women, including those from some of the best colleges, to fill critical vacancies and to "free a man to fight." These women often found themselves doing tedious work, but gained a foothold in the technical workplace.

According to Liza Mundy's *Code Girls*, over 10,000 women were a critical part of the cryptologic mission, some working with the early computing machines.[1] In the U.S. many women who had technical skills were sent home after the war to free the jobs for men returning from war. More generally, women's place in computer history has not been publicized because it has largely been HIStory, focusing on hardware and the male inventors,[2] as I saw on my visit to the Computer History Museum in Mountain View, California, in July 2018.

Fortunately, modern cryptology, in particular, was welcoming to women from the start. Elizebeth Friedman and Agnes Driscoll led the way in the 1920s and 1930s.[3] The work of the "Code Girls" during World War II was critical for winning the war. Like their contemporaries at NASA, whose story was told in the bestselling book and hit movie *Hidden Figures*, the women at NSA, walking in the footsteps of their World War II sisters, have broken ground from the 1960s on as they contributed to advances in computing in the world of cryptology.[4]

Digital Object Identifier 10.1109/MAHC.2020.2982751
Date of current version 29 May 2020.

 Published by the IEEE Computer Society

Many of the Queens of Code were recruited by NSA right after college and worked in computing technology for 30, 40, and even 50-year careers. I was one of those queens, hired in 1970, with one of the first undergraduate degrees in computer science in the country. Starting from data systems interns and rising to senior leaders and computer science experts, we were on the forefront of computer technology development. In the 1960s, 1970s, and 1980s, our agency had the most sophisticated computers in the world as well as the most challenging information processing requirements. By 1968, NSA had more than 100 computers spread over five acres of computer rooms.[5] The inventory grew rapidly over the next decades as we and our male colleagues worked with many vendors to drive new system development to meet our big data processing needs.

Our stories may also provide some insight to companies today that struggle to recruit and retain women in tech. In contrast to corporations and institutions in various other sectors, NSA did a lot right over a 50-to-60-year period to recruit, develop, and retain their computing women. They had learned from previous experience with the Code Girls during World War II that women were a valuable asset to their mission. They invested in us through training, intern programs, and advanced degrees, paid equal starting salaries for men and women, gave women responsibility and credit, promoted many women to senior management and technical positions, and provided a good work/life balance. Fortunately for us, most of the men we worked with were supportive as well. Of course, there were some struggles along the way, including a class-action lawsuit over fair promotion in the 1970s,[6] but we prevailed. The Queens of Code made a daring leap into a new career field of computer science and found innovative, exciting, and rewarding careers that contributed to the high-tech world we live in today

The rest of this article highlights some of the experiences I, and many other women, shared working on our first computers.

FIRST ENCOUNTERS OF A BINARY KIND

If you grew up in the 1980s or 1990s as part of the millennial generation, your first experience on a computer might have been with a personal computer at home or at school. You might have learned to program in basic using my Micro Adventure books[7] on an Apple II, Radio Shack TRS-80, Atari, or IBM PC. If you are part of Generation Z, you probably played games on your first computer tablet or smart phone maybe as early as a toddler. E-books, apps, and online shopping and learning are things you took for granted.

That was not the case when the Queens of Code were young. The ARPANET (the early version of the Internet that had just begun to come online in 1969) only connected some dozens of government agencies, universities, and other research organizations, and the World Wide Web had not been invented. Back in the early 1970s, one of our offices did have a terminal that we could use a modem to dial into the National Bureau of Standards' ARPANET. From NBS, we could connect to the Stanford Research Institute—and it took dozens of steps to send a line of text along with manually calculated checksums (a digit that was the sum of the other digits in a piece of data used to detect errors).

When we were growing up, there was little digital computing technology in the schools we attended before college. Pocket-size calculators made their debut in the 1970s. Before then, in high school or college, we used a slide rule (the manual device invented in 1620) for math, chemistry, or physics courses.

Many of us were 18-to-22-year olds when we met our first computer, perhaps an IBM 1620, 1401, or even 360 (after its release in 1964) at our college or university. Often the Queens of Code's first computing experiences were on their initial assignment at NSA or at college. These computer installations could be huge and expensive, especially those in NSA's extensive basement sometimes taking up spaces as big as a couple of basketball courts, cooled by water under the floors to make the rooms so cold that you had to wear a heavy sweater or jacket when working there.

Some of our first computing experiences were on computers with limited capacity and programming done in assembly language or even octal, and that was not easy. We had to be crafty to make the programs work within the constraints. FORTRAN, the first commercially available computer language to use a compiler was released in 1957. A compiler meant that the code could be written with higher level and easier to manage commands that would automatically

generate the assembly language or machine code needed for a specific machine. FORTRAN was designed to provide a language for the scientific community, and NSA certainly fit in that box. As computer technology advanced and memory size increased and became less expensive, programmers could write code with less computer specific restrictions.

Dottie Blum, a legendary computing woman at our agency, was using FORTRAN as early as 1954 even as it was being developed by John Backus and team at IBM. At first, people wondered if using a compiler would produce code as efficient as writing in assembly language. But over time, computer speed and memory size increased and convenience won out. Another benefit was that programs could be ported (moved over) to run on other machines that had a FORTRAN compiler. It was a big improvement over having to rewrite programs in another assembly language every time a new computer came into our collection.[8]

Programming was a little like cooking. You had input (like your ingredients) and then steps that processed the ingredients. If all worked well, you have something to eat for supper. Fortunately, I was a better programmer than a cook.

Our first programs were either assignments at school or "toy" programs we were assigned to learn how the computer worked. On the earliest computers from the 1950s like the special purpose ones built in house, there was only the basic documentation, so you had to figure things out for yourself. Sarah, one of our first programmers, used to say that when she started at NSA the computers took up a whole room and you were lucky to find a small notebook with instructions. By the time she retired, the computers were small enough to fit on your desk, and you had a bookcase of manuals and online documentation.

In our environment, the programming process worked as follows. The first step was to define the problem. In our case for application programs, this meant talking to the analyst to understand the problem that needed solving. The problem was often to automate a time-intensive manual process such as an attack on a cryptographic code we had collected by analyzing signals or language translation. NSA processed tons of data to produce intelligence reports for government decision makers including the President and the military. NSA was doing "big data," long before the phrase was coined in 2005. Programs were written to support requirements at the time.

The programmer would then break the process down into small steps that would provide a solution. Programmers often use flowcharts to block out the steps that need to be taken. We used plastic templates back then to draw the flow charts.[9] Now there are many software tools and applications to help with program design.

Next, we had write the code in a programming language like FORTRAN, PL1, or C or in assembly language in the earlier days. Then, we had to debug it, resolving all the problems that we could find. After that, we tested with our real-world users; and, when all was working, officially declared the program live. Of course, there would always be more bugs that popped up, and we had to fix those in a timely manner.

At the agency, system programmers who worked on the operating systems and networking were in the C (for Computer) Organization (which was later reorganized and renamed T (for Telecommunications and Computer) Organization. It seemed that every three or four years we have a major reorganization, sometimes corresponding to a new Director's arrival. Some application programmers started out as part of C, but later moved out to sit with the users in the production organizations. All the reorganizations and reassignments were confusing. One of my bosses had a sign in his office that read, "Perfect reorganization is only achieved by groups on the verge of collapse."

LEGACY QUEEN'S FIRST ENCOUNTER WITH A COMPUTER

Dottie Toplitzky Blum, 1950

Dottie had worked with the Electronic Adding Machines (EAM) equipment and the Army's version of the BOMBE, an electromechanical device developed by Joe Desch of National Cash Register during WWII to decode Enigma messages. Another of Dottie Blum's earliest binary encounters was with the Standards Eastern Automatic Computer (SEAC), which was built in Washington, DC, USA, for the National Bureau of Standards. The SEAC was one of the first U.S. stored program computers. Dottie then worked for AFSA, the Armed Forces Security Agency, NSA's predecessor. AFSA did not have their own computer

but the support organization did manage a number of calculating and cipher machines including the Navy's and Army's BOMBEs that were used in the war effort. These earlier devices were not actually computers since they lacked memory or ability to do anything outside of their limited computational functions such as compiling and comparing text, searching for cribs (plain text), or calculating statistics.

It was 1950 and Dottie and Sam Snyder, one of her coworkers whose computer history writings documented this story, got an urgent request from the Navy's Communications Security Division. The job required the verification of a few hundred involutory 4 x 4 matrices[10] that were used in the Navy callsign system. The SEAC's memory was only 512 (45 bit) words, which was pretty limiting. Back then, they had to negotiate time on the SEAC to debug the program, and the only time NBS would allow them to purchase (at $24 an hour) was after midnight or on Sunday afternoons.

The program was written but they needed test data. Dottie, who was working for the Machine Production Organization as an IBM specialist, produced thousands of random numbers on punched tape to be used to test the application. The SEAC took between 8 and 15 seconds to process each matrix and then printed out the ones that met the "useful" criteria. With a lot of work and some late nights and weekends, Dottie and Sam got the information to the Navy in a timely manner to help solve the problem. Sam said, "Those who participated in this task found the experience 'frustrating, exhilarating sense of accomplishment and participation in making history."[11]

MORE FIRST-HAND ENCOUNTERS FROM OUR QUEENS OF CODE

Carol McWilliams, 1967–1970

My first programming experience was assembly language on a CP818 (UNIVAC 1224) for field installation. We "wrote" our programs on a Kleinschmidt—something like a typewriter, but it produced punched paper tape with one instruction per line (e.g., "clear register"). You could fix an error by wrapping Scotch tape over the holes in the line and repunching the line! Fortunately, the readers were not sensitive to the opacity of the tape, just the holes. The resulting paper tape was wrapped butterfly style in a figure eight with a paper clip in the center and stored until you had time on the computer.[12]

The first programs I wrote were standalone processes. I had data input from magnetic tape, ran the program, and produced data output. I scheduled computer time and, when it was my time, I took my paper tape and mag data tape to the computer room. After loading the paper tape into memory, I put my magnetic tape on the spool and initiated the program. When I was done, I took my program tape, mag tape, and results off the computer, cleaned the heads for the next programmer, and took everything back to my desk to assess the results and debug my program. Very much a hands-on process!

Eileen Buckholtz, 1968-1969

I had transferred to Ohio State University (OSU) as a math major and was taking a fourth course in calculus while struggling with the theoretical proofs in the class. I remember the professor covering a big blackboard with the proof of the Heine–Borel Theorem. He got near the end, realized that he had made a mistake and started to erase half his scribblings. My eyes were glazing over. What was I doing here?

Later that afternoon, I heard that OSU was opening their computer science department and they were looking for students. My boyfriend Howard was in engineering and he heard the same thing. Turns out they were offering degrees in both the Arts and Sciences Department, where I was enrolled as well as an engineering computer science degree. We both signed up, became OSU's first computer science graduates, and have been computing together for over 50 years.

There were only several dozen students in the first computer science classes. It was love at first byte for me when I took my first programming course. The initial assignment was a simple sort. The next was to use a random number generator to simulate shuffling a deck of cards and a matrix for holding all the hands. The idea that you could learn a language like FORTRAN and make an enormous computer do your bidding with structured commands was just fun. As we got into more advanced programs, it became challenging as well.

An IBM 360 installation including CPU, tape drives, IO controllers, and other peripherals were housed in

a big computer room that took up much of an upper floor of the engineering building. We could see into the computer room through big windows but we were not allowed to go in. After class, we would design our programs and then punch up Hollerith cards (one line of code per card) on keypunch machines, submit them over the counter, and then wait 5 or 6 hours for them to run and get our output back. If we made an error, we had to correct that and submit again. No wonder the four women in the computer science program were all dating guys also in the program. Who else would want to spend Friday and Saturday nights debugging programs?

Elaine Mills, 1965

As part of a work-study program at Towson State College (known today as Towson University), I was studying to become an elementary school teacher. I was privileged to be assigned to a special project to "computerize" all the records for the five Maryland state colleges. Using Hollerith cards and becoming proficient in writing FORTRAN programs in the mid-1960s was a "blast" that actually proved to be a tremendous personal boost a few years later at NSA.

Kathleen Jackson, 1967

My orientation started with a tour of the "basement," a huge area where the computer I would be using, the UNIVAC 1108, was located. The system was so large that it nearly filled the whole computer room, since it had several printers and other pieces of peripheral equipment attached to it. My job was to remove computer printouts from one of the printers, review the data, look for data "anomalies," and adjust the FORTRAN software as needed to fix them. It seemed challenging and interesting at first but, as the days and weeks rolled by, reviewing rows and rows of 1s and 0s became a little tedious to put it mildly. However, I persisted. After completing my tour, I looked forward to my next assignment. Over the years, I sometimes thought about that initial assignment, and how different it was from all the other work I had done at the Agency.

Several years later, I came across a report that contained Agency historical information. It included information about the data that was processed on that UNIVAC 1108 computer I was supporting during my first assignment in 1967. This report identified how critical those data were to national security at that time. Suddenly, I became acutely aware that the many hours I had spent reviewing those 1's and 0's were definitely worth the challenge of the task. In the end, I determined that this work was probably some of the most significant work I did during my entire career at the Agency. I took pride in knowing that this work was very important to the security of our nation.

Kathleen Reading, 1982

"Oh great, another girl!" Imagine hearing those words upon meeting your supervisor for the first time. I was 21 years old and just beginning my 34-year career in the Information Assurance Directorate (IAD), in the Agency's print shop. I was taken aback by my boss's comment, but did not say anything as I was just starting a new job and did not know what to expect. I do remember thinking I was going to do everything in my power to change my boss's mind about what "girls" could do.

My job title was "Reproduction Worker," and I was one of three women working in the shop. I found that job title pretty funny. I first started working in the bindery, and then also ran a printing press, large Xerox machines and printers, and eventually worked in the Electronic Printing and Publishing (EPP) branch. In the EPP branch, for the first time, documents to be printed were sent electronically via computer by Agency customers; and documents also were sent electronically to the printers for printing. One of the documents printed on the night shift was a daily report that was couriered each day to the White House.

As it turns out, I did prove to my boss that women are good workers. I was promoted several times, and was also one of the first women selected to participate in the Agency's first ever production trade program.

Mary Clulow, 1977

"I will rule these machines; they will not rule me." Quietly determined, I spoke these words late one evening at work while trying to complete a typing task. I was using an IBM Magnetic Card Selectric (aka MagCard) typewriter. This was quite a bold statement for an entry level Clerical Assistant at the National Security Agency. However, I had been challenged by this machine more than once since learning to use it two years prior, in 1977.

For those who may not be familiar with the Mag-Card, it was quite state of the art for its time and was the upgrade to its predecessor, the Magnetic Tape typewriter. Basically, after typing on bond paper, it recorded one page at a time, provided you inserted a magnetic card (much like an IBM card, but Mylar and magnetic) and pressed Record, *before* turning the machine off—or else it was not saved. Once recorded, the file could be edited by picking the related card to the desired page from the labeled envelope, inserting it into the card reader, and then pressing Read. The file then could be played back one page, one line, one word, or one character at a time. The playback was rather quick, making it easy to miss the mark. Sometimes, the paper ripped during a return motion. This was one of those moments; blame it on user error, but I finally thought, enough was enough, and enrolled at the local community college.

This was the beginning of my journey into advanced learning, leading to a BS degree in information management, but definitely not the end of using other machines that would enter NSA's workspaces. They provided more word processing technologies, office automation, and then advanced further into end user computing.

Maureen McHugh, 1969

I graduated from Marywood College in May, 1969. My experience with computers was limited. I was a Math major in a college whose curriculum focused mainly on training teachers. I did not want to be a teacher. It was obvious in 1969 that computer work would be an exciting field and I could get in on the ground floor. As a senior in college, I took a FORTRAN II class.

The teacher was a business professional who taught a few night classes at the college. He had a customized van in which he had a card punch machine, a printer, and sorter. It was only accessible a few hours a week including class time. Writing and debugging our "toy" programs was difficult, to say the least. We submitted our punch card deck to the teacher, and he would return it the following week after running it on a computer back in his office. One turnaround per week! A single typo could set you back two weeks. I think I got a B in the course, but certainly did not feel as though I had mastered FORTRAN.

Peggy Strader, 1969

During my intern tour, I was introduced to the UNIVAC 494 and the SPRYE assembly language. This was an octal-based system so I learned and became proficient in reading dumps in octal. On this system I honed my skills in SPRYE, FORTRAN, and ALGOL. I believe this group was responsible for the first Information Storage and Retrieval System named TIPS (Technical Information Processing System) and its retrieval language named TIPS Interrogation Language. On this system, we were able to put our queries on model 35 teletypes and it would search magnetic tapes of data or magnetic drums for the information requested. I became the user/customer interface for these systems, often teaching the Boolean logic and constructs necessary to retrieve the information needed.

Lois Gutman, 1970

I had no real computer or programming experience other than creating small card decks for overnight runs on a cardpunch machine in a summer job at Johns Hopkins University. NSA's high-level programming language at the time was IMP,[13] running on the operating system FOLKLORE, NSA's homegrown time-sharing system, developed by the Institute for Defense Analyses in Princeton, NJ, USA. Everyone in my office used CDC-6600 computers and sat in a large open tube room, a room full of Cathode-ray tube (CRT) terminals connected to a mainframe where programmers could work on their code in the NSA Headquarters building basement. Operators hung large magnetic tape reels for users. We stacked the reels on our desks (under sheets of black cloth for security) and made hanging decorations from colored plastic write rings.

Toby Merriken, 1970

Fresh out of college, I went to work for NSA in 1966. I started out in the Cryptanalysis Intern Program and became certified as a professional in that field. Shortly after that, in 1970, I joined a newly created branch dedicated to using computer science for the first time for cryptanalytic applications. With no computer training or experience, I wrote programs in FORTRAN and learned a lot on the job.

I wrote each program in longhand and took it to a staff of key punch operators to transfer to keypunch

cards. I then took the cards to the remote job entry (RJE) room housing a printer and a card reader, into which I manually fed the cards. The input went to the computer mainframe and was printed out on the printer in the RJE room. The computers were high tech for the time but not interactive. A great deal of time transpired between the writing of a program and the implementation of it. I left this branch in 1974 to return to cryptanalysis and eventually became a linguist.

Marie Rowland, 1970

When I started at NSA, I can honestly say I knew nothing about computers. I had gone to an all-women's college and majored in math. The only exposure to computers we experienced came one afternoon when a guest speaker explained to us that we would all have to learn a language called FORTRAN. So, when I landed at the Agency with my group of new mathematicians, management decided that with my background I should start at the beginning.

I was assigned to the Research organization, where they handed me a small box and explained they were doing research on how small computers might one day be used to schedule jobs for large computers. My task was to teach the small box to tell time. Now, I was naïve enough to believe this and did not realize that it was actually a good exercise for me to learn about how to program computers and really understand them from the inside out.

I started reading the manuals and some library books. Each morning I plugged in my little box and got it going with a paper tape from a teletype, starting a heartbeat interrupt, a periodic signal that the hardware generated to indicate its working or to synchronize other parts of the system. I could count these beats and get up to a second, then a minute and so forth, and thus tell time. The little machine only had a few instructions such as *load, compare,* and *store* so it seemed much easier than the FORTRAN description. The biggest headache was working with the teletype and paper tape. An "all thumbs" affliction was to plague me through card decks and keyboards, through all my years of writing code.

The library books said I could name my variables anything I wanted. I took this to heart and called them names from the book I was reading, *The Hobbit*. Thus, "BILBO" became the second counter. Eventually, the person guiding me looked at my work and gently mentioned that it was traditional to name the variables after the function they performed so other people could follow the program. DUH! Later, as I was finishing the project, I asked if he thought I should put in a routine to handle Y2K—something I had discovered in my library research. I do not know how he kept a straight face when he replied that it probably was not necessary for the purpose of this project. I often remembered this Y2K-innocence when it struck with a vengeance years later.

As it turned out, teaching this little machine to tell time was a very good introduction to the world of computers. Programming it illustrated the "edge" of the hardware and software divide and left me completely fearless to wade into all kinds of hardware–software issues. I realized what a leap it was from the early computer greats made in the 1940s and 1950s when their research allowed them to move from purpose-built machines to building machines that kept both the instructions and the data that the instructions worked on in the same form. I went on to write many programs and eventually received an MS in Computer Science from Johns Hopkins, but I was always able to view the complexity of tasks through the lens of my first project.

REFLECTION

We all had memorable encounters with our first computers and went on to have rewarding careers in technology. Our Queens of Code are good examples of how women were working with early computer technology. NSA gave us opportunities to excel in this exciting new career field.

Over the past 50 years, women have continued to bring their talents and skills to the technology revolution. We hope our project will encourage other women computer pioneers in both the public and private sectors to step forward and tell their stories. While much has been written on the low percentage of women graduating with computer science degrees[14] and problems with retaining female technical employees, it is critical for the future of tech that women's ideas and points of view be part of future developments. We hope our stories will inspire more women to pursue STEM careers.

Look for more stories from the Queens of Code and follow our journey on Facebook: https://www.facebook.com/QueensofCode/.

Our website: https://Queensofcode.com.

On Twitter: @QueensofCode

REFERENCES

1. Liza Mundy, *Code Girls: The Untold Story of the American Women Code Breakers of World War II*. New York, NY, USA: Hachette, 2017.
2. J. Abbate, "Women and gender in the history of computing," *IEEE Annu. History Comput.*, vol. 25, no. 4, pp. 4–8, Oct.–Dec. 2003.
3. Friedman is the subject of two recent books: G. S. Smith's, *A Life in Code: Pioneer Cryptanalyst Elizebeth Smith Friedman*. Jefferson, NC, USA: McFarland & Company, Inc., Publishers, 2017 and J. Fagone's, *The Woman who Smashed Codes*. New York, NY, USA: Dey Street, an imprint of William Morrow, 2017. Agnes Driscoll's mostly unknown life is the subject of K. W. Johnson's, *The Neglected Giant: Agnes Meyer Driscoll*. Ft. George G. Meade, Center for Cryptologic History, 2015. Another little-known early female cryptologist, Genevieve Young Hitt, is the subject of B. R. Smoot's, "An accidental cryptologist," *Cryptologia*, vol. 35, no. 2, pp. 164–175, 2011.
4. Hidden Figures: *The American Dream and the Untold Story of the Black Women Mathematicians Who Helped Win the Space Race*. New York, NY, USA: William Morrow, 2016.
5. NSA's Key Role in Major Developments in Computer Science, Part Two, Accessed: 2019. [Online]. Available: https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/nsa-early-computer-history/6586785-nsa-key-role-in-major-developments-in-computer-science.pdf
6. R. Predmore-Lynch. Accessed: 2019. [Online]. Available: https://www.nsa.gov/About-Us/Current-Leadership/Article-View/Article/1620951/renetta-predmore-lynch/
7. I was a co-creator of the popular Micro Adventure series (Scholastic 1984, with Ruth Glick). [Online]. Available: http://www.microadventure.net/Home/About/. We had a talented team of mostly women writers and programmers and some teens to create the series. 1984–1986.
8. Dottie T. Blum, Hall of Honor Inductee. 2004. [Online]. Available: https://www.nsa.gov/About-Us/Current-Leadership/Article-View/Article/1622398/, Accessed: 2019.
9. Read more about flowcharting templates in the Peggy Aldrich Kidwell's article in the IEEE Annals of the History of Computing (vol. 41, no. 1). Accessed: 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8667955
10. Involutory matrices can be used in visual cryptography to transform the data. A more detailed example of the Hill cipher algorithm can be found at Accessed: 2020. [Online]. Available: https://pdfs.semanticscholar.org/9626/7d1194c8beffc8abcf8b142f9870051bdb7c.pdf
11. S. Snyder, Earliest Application of the Computer at NSA (Snyder), 1973.
12. 12.Details on the history of Punched paper tape. Accessed: 2020. [Online]. Available: https://en.wikipedia.org/wiki/Punched_tape
13. 13.Read more about IMP at: Accessed: 2020. [Online]. Available: http://www.liquisearch.com/imp_programming_language https://www.seas.harvard.edu/courses/cs152/2016sp/lectures/lec05-imp.pdf
14. 14.Only 18% of computer science degrees were earned by women in 2016. Accessed: 2020. [Online]. Available https://www.computerscience.org/resources/women-in-computer-science/

EILEEN BUCKHOLTZ is a Maryland-based computer scientist and author of 40 books. She received a M.S. degree in computer science from the University of Maryland, a B.S. in computer and information science from Ohio State University, and a Chief Information Officer certificate from the National Defense University. For over 30 years, she had an exciting and distinguished career working on cutting-edge technology for the National Security Agency.

EDITOR IN CHIEF: Ipek Ozkaya, Carnegie Mellon Software Engineering Institute, ipek.ozkaya@computer.org

This article originally appeared in IEEE Software vol. 38, no. 2, 2021

DEPARTMENT: FROM THE EDITOR

Mom, Where Are the Girls?

Ipek Ozkaya

During the fall semester of 2005, I was working hastily on the finishing touches of my Ph.D. dissertation at Carnegie Mellon University. That semester, I also was the teaching assistant for the Methods of Software Development graduate course taught by Dr. Mary Shaw and Dr. Jim Herbsleb. It was a busy time, with the challenges of finishing graduate school; getting ready for a new job; fulfilling responsibilities such as grading and helping students; and parenting my then three-and-a-half-year-old daughter. Methods of Software Development was a demanding course with a lot of reading and reflection assignments. Students took abundant advantage of the office hours. Those meetings always went better if I remembered the students' names, but with that was all going on, my brain did not always comply, so I had a hack. I had printed all of their photos and hung them right above my desk.

IT WAS DURING ONE OF OUR "LET'S PLAY AT MOMMY'S OFFICE" VISITS WHEN I FIRST BECAME AWARE OF THE DIVERSITY ISSUE IN SOFTWARE ENGINEERING.

As with all graduate students, my weekends consisted mostly of work. I often took my daughter into the office with me over the weekends to give her a glimpse of my work life and sneak in some tasks. It was during one of our "let's play at Mommy's office" visits when I first became aware of the diversity issue in software engineering. After staring at the photos of the Methods of Software Development fall class members of 2005 for a couple minutes, my daughter asked who they were. I explained that they were the students with whom I was working. She continued to study the photos, and I started to concentrate on my work. After a couple more minutes, I heard, "Mom, where are the girls?" I did not understand the question at first and asked her to explain. Her three-and-a-half-year-old observant mind was trying to categorize the students as at that age she was just starting to recognize gender.

It is my response in all of its irony that demonstrates one of the reasons that we have the diversity, equity, and inclusion issues. Once I finally understood the question, I, without any doubt, said, "Oh, they are all there, let's find them." I put her on my lap, and we started studying the photos. There were two female students out of a total number of 27 people. At that age, my daughter was content with finding the two girls and moved on to exploring something else. She did ask "more" once or twice as we studied the photos, not in search of more girls, but simply because "more" was her most favorite phrase due to her daycare routines.

I do recall, however, being confused for the first time. Panicking, admittedly not because there were only two female students in the 2005 Masters of Software Engineering class at Carnegie Mellon University, but because I felt like a young, tired, and inexperienced mom showing her daughter a bad example in my very own office. I was supposed to be teaching her that she could become anything she wanted, that she had the power, and that others had paved the way for her. The example I was supposed to set was not that we had to look close to find the two female students in the field of the future at the very university leading the way. I, as a female who consistently had been an underrepresented minority in her field, was not aware of the issue until that very exchange.

Digital Object Identifier 10.1109/MS.2020.3044410
Date of current version: 11 February 2021

 Published by the IEEE Computer Society

BECOME AWARE

On the one hand, this memory reflects my luck. Clearly, I have been fortunate enough to not have felt as underrepresented in a field where I consistently have been. I had worked within teams that made me feel welcome. I had colleagues, supervisors, mentors, and advisors who advocated for me, gave me timely and concrete feedback, motivated me, provided me with opportunities, and appreciated my contributions. On the other hand, it also reflects the unfortunate reality—we cannot assume that people are aware. Even those who themselves are members of underrepresented groups may not be aware of the extent of the issues at hand. We cannot assume that institutions are doing their part, even the best of the best. Carnegie Mellon University has come a long way since 2005—admission committees at that time clearly were not aware or, in any event, aware enough. Today, Carnegie Mellon boasts the ability to achieve close to 50% admission rates of qualified female students across most of its degree programs at the undergraduate level, including in engineering and computer science. Graduate admissions numbers are also definitely way better than two out of 27. In addition, there are several university-wide initiatives to address other diversity, equity, and inclusion challenges. The road ahead to achieve a truly diverse, inclusive, and equitable software engineering community is still long and not smooth as we all are aware.

Many global and national initiatives such as Girls Who Code, Girl Develop It, Black Girls Code, and countless others have also taken it upon themselves to empower, motivate, and educate females to enter careers in software engineering. Bringing more women into software engineering is not a solved challenge despite the significant amount of attention it has received. Overcoming challenges of retention, salary equity, and growth opportunities are still in the works.

But achieving diversity with enough female representation is just the tip of the iceberg. We are finally learning that gender is not a binary identity; those who identify as lesbian, gay, bisexual, transgender, questioning (LGBTQ+) face a number of very different challenges and biases as software engineers.

Gender is one of several aspects of diversity. When teams are diverse, with representation from cultural, ethnic, economic, religious, political, and technical backgrounds, they are more productive and stronger. History has shown us again and again that there are many underrepresented groups that, when empowered, will help move a field forward. But this starts with awareness. Each of these diversity groups may demonstrate similarities; however, each also has its unique challenges. Becoming truly aware takes patience, an open mind, and learning to act the right way. I was fortunate enough that my awakening moment did not involve me being frustrated against an unfair situation, feeling left out, or not finding people like myself with whom I could identify. This is an exception, not the norm.

WHEN TEAMS ARE DIVERSE, WITH REPRESENTATION FROM CULTURAL, ETHNIC, ECONOMIC, RELIGIOUS, POLITICAL, AND TECHNICAL BACKGROUNDS, THEY ARE MORE PRODUCTIVE AND STRONGER.

BEING AN ADVOCATE AND TAKING ACTION

Being an advocate for diversity, equity, and inclusion takes every single one of us to drive meaningful change, starting at the personal level all the way up to organizational levels. Advocacy is any action that speaks in favor of, recommends, argues for, supports, defends, or pleads on behalf of a cause or for others. And the most impactful advocacy is achieved when our allies are diverse and show up with concrete actions in support. The CEO of Girls Who Code recently published a public thank-you note to Jack Dorsey, CEO of Twitter, for his advocacy and financial support for Girls Who Code.[1] She emphasized how the biggest ally of Girls Who Code has, in fact, been a man passionate about empowering women and girls to enter software engineering. We need examples demonstrating allies working together to improve diversity, equity, and inclusion in software engineering. Organizations that hire software engineers need to do their part as well. Large, global software engineering organizations like Google[2] and Microsoft[3] have started publishing yearly diversity, equity, and inclusion reports to share their data. Objectively understanding the state of the situation is one step toward improving diversity,

equity, and inclusion in software engineering. Software engineering research only recently has started to look closer at studying the implications of diversity, equity, and inclusion on software engineering work. In the guest editorial in this issue of the magazine, Albusays and colleagues summarize concepts and related current research work.[4]

IEEE Software has always kept diversity, equity, and inclusion as part of its core values. But we, too, need to do more. Despite our many efforts, the gender diversity of our magazine can improve. We have a long way to go to improve representation from people of color on our boards. While we have always strived to achieve global diversity, we struggle with including enough readers and authors from the Far East. Diversity, equity, and inclusion are our board's collective responsibilities. However, to bring targeted focus and identify actionable steps, we are also launching an initiative dedicated to this cause. We are in the process of establishing a workforce team and will strive to share our data and the steps that we take. This is one way we are taking action, and we will do more, including featuring case studies, experiences from industry, and empirical research results regarding improving diversity, equity, and inclusion in software engineering. We trust that the software engineering community will keep us accountable.

REFERENCES

1. R. Saujani. "Thank you, Jack Dorsey." *Medium*. https://medium.com/@GirlsWhoCode/thank-you-jack-dorsey-cc9a210184b6 (accessed Dec. 2020).
2. "*Google diversity annual report*," Google, Mountain View, CA. Accessed: Dec. 2020. [Online]. Available: https://diversity.google
3. "*Global diversity and inclusion report*," Microsoft, Redmond, WA, 2020. Accessed: Dec. 2020. [Online]. Available: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE4H2f8
4. K. Albusays et al., "The diversity crisis in software development," *IEEE Software*, vol. 38, no. 2, pp. 19–25, Mar.–Apr. 2021. doi: 10.1109/MS.2020.3045817.

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

10 May

- CCGrid (IEEE/ACM Int'l Symposium on Cluster, Cloud and Internet Computing), virtual
- ICFEC (IEEE Int'l Conf. on Fog and Edge Computing), virtual

15 May

- BigDataSecurity (IEEE Int'l Conf. on Big Data Security on Cloud), virtual

17 May

- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium), virtual

18 May

- RTAS (IEEE Real-Time and Embedded Technology and Applications Symposium), virtual

19 May

- TechDebt (IEEE/ACM Int'l Conf. on Technical Debt), virtual

20 May

- CHASE (Int'l Workshop on Cooperative and Human Aspects of Software Eng.), virtual

23 May

- ICCP (IEEE Int'l Conf. on Computational Photography), Haifa, Israel
- ICSE (IEEE/ACM Int'l Conf. on Software Eng.), virtual
- SP (IEEE Symposium on Security and Privacy), virtual

24 May

- ETS (IEEE European Test Symposium), virtual
- SEENG (Int'l Workshop on Software Eng. Education for the Next Generation), virtual

25 May

- ISMVL (IEEE Int'l Symposium on Multiple-Valued Logic), virtual

29 May

- SoHeal (Int'l Workshop on Software Health in Projects, Ecosystems and Communities), virtual

30 May

- GI (Int'l Workshop on Genetic Improvement), virtual

31 May

- SEmotion (Int'l Workshop on Emotion Awareness in Software Eng.), virtual

JUNE

1 June

- ISORC (IEEE Int'l Symposium on Real-Time Distributed Computing), Daegu, South Korea
- Q-SE (Int'l Workshop on Quantum Software Eng.), virtual

2 June

- ICIS (IEEE/ACIS Int'l Summer Conf. on Computer and Information Science), Shanghai, China

7 June

- BCD (IEEE/ACIS Int'l Conf. on Big Data, Cloud Computing, and Data Science Eng.), Macao
- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), virtual
- WoWMoM (IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks), Pisa, Italy

14 June

- ARITH (IEEE Int'l Symposium on Computer Arithmetic), virtual
- ISCA (ACM/IEEE Int'l Symposium on Computer Architecture), virtual

20 June

- SERA (IEEE/ACIS Int'l Conf. on Software Eng. Research, Management and Applications), Kanazawa, Japan

21 June

- CSF (IEEE Computer Security Foundations Symposium), Dubrovnik, Croatia
- DSN (IEEE/IFIP Int'l Conf. on Dependable Systems and Networks), Taipei, Taiwan

26 June

- CSCloud (IEEE Int'l Conf. on

Cyber Security and Cloud Computing), Washington, DC, USA

JULY

5 July

- ICME (IEEE Int'l Conf. on Multimedia and Expo), Shenzhen, China

7 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Washington, DC, USA
- SNPD (IEEE/ACIS Int'l Conf. on Software Eng., Artificial Intelligence, Networking and Parallel/Distributed Computing), Taichung, Taiwan

12 July

- COMPSAC (IEEE Computers, Software, and Applications Conf.), Madrid, Spain
- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), virtual

27 July

- SMC-IT (IEEE Int'l Conf. on Space Mission Challenges for Information Technology), virtual

AUGUST

9 August

- ICKG (IEEE Int'l Conf. on Knowledge Graph), Hong Kong

11 August

- IRI (IEEE Int'l Conf. on Information Reuse and Integration for Data Science), virtual

16 August

- ACSOS (IEEE Int'l Conf. on Autonomic Computing and Self-Organizing Systems), Washington, DC, USA

23 August

- SCC (IEEE Space Computing Conf.), virtual
- SMARTCOMP (IEEE Int'l Conf. on Smart Computing), Irvine, USA

SEPTEMBER

5 September

- SERVICES (IEEE World Congress on Services), Chicago, USA

7 September

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Portland, Oregon, USA
- EuroS&P (IEEE European Symposium on Security and Privacy), Vienna, Austria

8 September

- MIPR (IEEE Int'l Conf. on Multimedia Information Processing and Retrieval), Tokyo, Japan

20 September

- eScience (IEEE Int'l Conf. on eScience), Innsbruck, Austria

OCTOBER

1 October

- ISPA (IEEE Int'l Symposium on Parallel and Distributed Processing with Applications), New York, USA

4 October

- IC2E (IEEE Int'l Conf. on Cloud Eng.), San Francisco, USA
- LCN (IEEE Conf. on Local Computer Networks), Edmonton, Canada

10 October

- MODELS (ACM/IEEE Int'l Conf. on Model Driven Eng. Languages and Systems), Fukuoka, Japan

13 October

- FIE (IEEE Frontiers in Education Conf.), Lincoln, Nebraska, USA

NOVEMBER

15 November

- ASE (IEEE/ACM Int'l Conf. on Automated Software Eng.), Melbourne, Australia

DECEMBER

20 December

- MCSoC (IEEE Int'l Symposium on Embedded Multicore/Many-Core Systems-on-Chip), Singapore

Get Published in the *IEEE Open Journal of the Computer Society*

Submit a paper today to the premier open access journal in computing and information technology.

Your research will benefit from the IEEE marketing launch and 5 million unique monthly users of the IEEE *Xplore*® Digital Library. Plus, this journal is fully open and compliant with funder mandates, including Plan S.

Submit your paper today!
Visit www.computer.org/oj to learn more.

75 YEARS IEEE COMPUTER SOCIETY | IEEE