CODE BLUE

issue 1.0

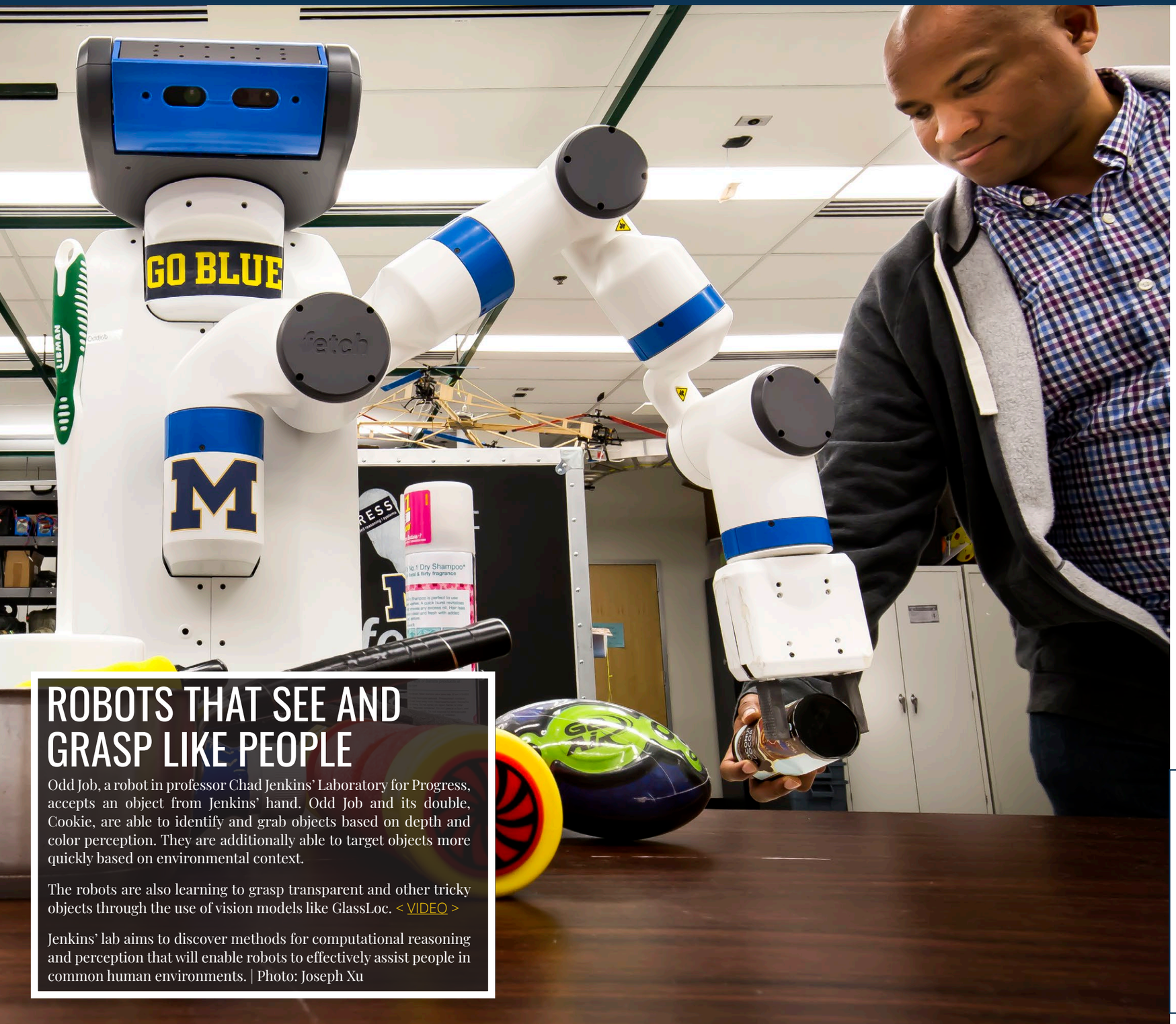
BUILDING A TESTING-FREE FUTURE

How automatically guaranteeing that our most complex programs are secure and trustworthy can save us time, money, and anxiety.

Also inside: HACKING REALITY • CENSORED PLANET • ACCELERATING HEALTHCARE

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

>WHAT'S INSIDE



ROBOTS THAT SEE AND GRASP LIKE PEOPLE

Odd Job, a robot in professor Chad Jenkins' Laboratory for Progress, accepts an object from Jenkins' hand. Odd Job and its double, Cookie, are able to identify and grab objects based on depth and color perception. They are additionally able to target objects more quickly based on environmental context.

The robots are also learning to grasp transparent and other tricky objects through the use of vision models like GlassLoc. < VIDEO >

Jenkins' lab aims to discover methods for computational reasoning and perception that will enable robots to effectively assist people in common human environments. | Photo: Joseph Xu

>CREDITS

EDITOR: Steven Crang

DESIGNER: Zach Champion

WRITERS: Zach Champion, Gabriel Cherry, Steven Crang

PHOTOGRAPHY/ILLUSTRATION: Steve Alvey, Zach Champion, Joseph Xu

Cover Illustration by Karen Parker Moeller, MOEdesign. ©2021

Regents of the University of Michigan

Jordan B. Acker
Michael J. Behm
Mark J. Bernstein
Paul W. Brown
Sarah Hubbard
Denise Ilitch
Ron Weiser
Katherine E. White

Computer Science and Engineering Division, College of Engineering

Bob and Betty Beyster Building
2260 Hayward Street
Ann Arbor, MI 48109-2121

cse.engin.umich.edu

>THE FEED

AFTER FIVE YEARS, LET'S ENCRYPT, A NON-PROFIT BASED ON TECH DEVELOPED AT MICHIGAN, HAS HELPED TO SECURE THE INTERNET

Just five years ago, most websites relied on unencrypted HTTP, the aging and inherently insecure protocol that provides no protection to sites or visitors from threats that range from surveillance through phishing to identity theft.

Today, the internet is a much more secure place, with over 80% of websites protected by HTTPS secure encryption.

That dramatic transformation — to a secure web — is due in large part to the activities of Let's Encrypt, a non-profit certificate authority (CA) founded five years ago by professor J. Alex Halderman and his collaborators.

Let's Encrypt has driven adoption of the digital certificates needed to enable secure sites by making them free, easy to install and manage, and readily available through hosting providers.

This approach has been a radical break from traditional practice, where implementing HTTPS has required website operators to engage in a number of costly and labor-intensive actions.

"As a non-profit," Halderman says, "we give people the digital certificates they need in order to enable HTTPS websites, for free, in the most user-friendly way we can. We do this because we want to create a more secure and privacy-respecting web."

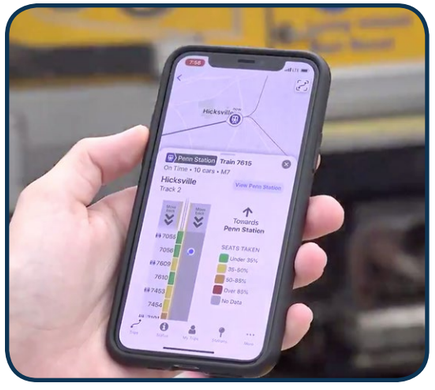
As a result of their unique approach, Let's Encrypt is today the world's largest CA, and over 225 million websites are protected by certificates issued by the organization.

Let's Encrypt's origins go back to 2012, when a research group led by Halderman and Peter Eckersley at the Electronic Frontier Foundation was developing a protocol for automatically issuing and renewing certificates. Simultaneously, a team at Mozilla led by Josh Aas and Eric Rescorla was working on creating a free and automated certificate authority. The groups learned of each other's efforts and joined forces in May 2013.

Let's Encrypt was publicly announced on November 18, 2014, issued its first browser-trusted certificate on September 14, 2015, and began providing service to the public on December 3, 2015.

< READ MORE >

Images link to the full story!



NEW U-M ROBOTICS BUILDING OPENS FOR WINTER 2021

The 140,000-square-foot, four-story complex collocates Ford Motor Company engineers with U-M researchers from all parts of the robotics enterprise for enhanced collaboration opportunities and faster real-world impact. CSE faculty specializing in robotics are occupying expanded lab space in the facility as Winter semester 2021 opens.

< READ MORE >

STUDENT DEVELOPER BUILDS OFFICIAL LONG ISLAND RAILROAD COMMUTER APP

A CS undergraduate led front-end development on a major update to Long Island Rail Road's TrainTime app, using more than half a dozen different sources of information to communicate train positions and car capacity and link it all in real time with a trip search feature.

< READ MORE >

BEFORE WE PUT $100 BILLION INTO AI...

America is poised to invest billions of dollars to retain its leadership position in the fields of artificial intelligence and quantum computing.

This investment is critically needed to reinvigorate the science that will shape our future. But to get the most from this investment, argues professor and Associate Director of the U-M Robotics Institute Chad Jenkins, we must create an environment that will produce innovations which uplift everyone in our society.

Writing for VentureBeat, Jenkins has discussed the need for investment in fixing the systemic inequalities that have sidelined Black people from contributing to AI and from having a hand in the products that will undoubtedly impact everyone. Black scholars, engineers, and entrepreneurs currently have little to no voice in AI.

Recently, legislation has passed greatly boosting America's investment in AI and quantum computing technology.

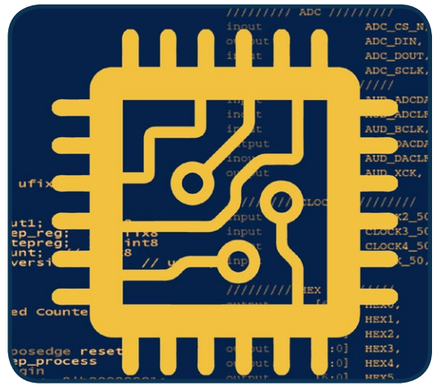
"As a Black American, I am deeply concerned about the outcomes and ill-effects that this surge of funding could produce if we do not have diversity in our development teams, our research labs, our classrooms, our boardrooms, and our executive suites," says Jenkins.

< READ MORE >

HARDWARE MODEL CHECKER TAKES GOLD AT INTERNATIONAL COMPETITION

The 11th Hardware Model Checking Competition had nine categories with 639 total benchmarks; the U-M system earned gold in seven categories, silver and bronze in the remaining two, and solved the most benchmarks, including 23 not solved by any other competitor.

< READ MORE >



ALUM-FOUNDED COMPANY RELEASES SOLUTION FOR AI-INTENSIVE APPLICATIONS

SambaNova, co-founded by CSE alumnus Kunle Olukotun, the Cadence Design Systems Professor at Stanford, has built a new platform from scratch that is optimized for AI applications. The company emerged from stealth mode to announce its first product, a system-level AI accelerator for hyperscale and enterprise data centers.

< READ MORE >

STUDENTS DESIGN NEXT-GEN AUTONOMOUS DELIVERY VEHICLES

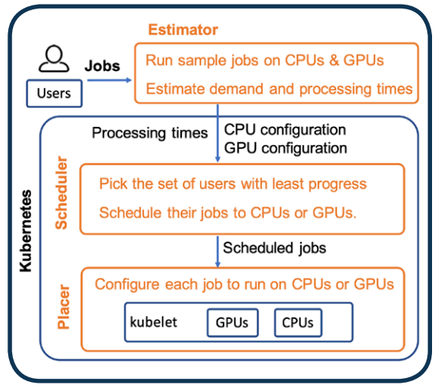
A team of undergraduate students envisions an autonomous delivery service that can deliver items to users that are inaccessible without a personal vehicle, such as groceries and medicine.

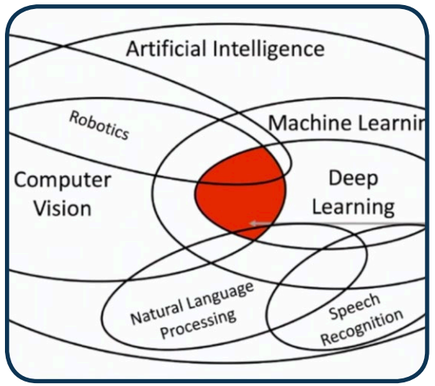
< READ MORE >

ENABLING FAIRER DATA CLUSTERS FOR MACHINE LEARNING

Researchers have designed a new smart scheduler that enables data centers with a blend of different computing hardware, such as CPUs and hardware accelerators, to share their resources more fairly.

< READ MORE >





GIFT ENDOWS CSE CHAIR AND FUNDS TARGETED ACTIVITIES

A gift of $2.5M from the Daniel E. Offutt III Charitable Trust has been made to endow the CSE Chair and fund targeted activities. Michael P. Wellman, the Lynn A. Conway Collegiate Professor of Computer Science and Engineering, has been named the first Richard H. Orenstein Division Chair of Computer Science and Engineering.

< READ MORE >

DEEP LEARNING FOR COMPUTER VISION COURSE AVAILABLE FREE ON YOUTUBE

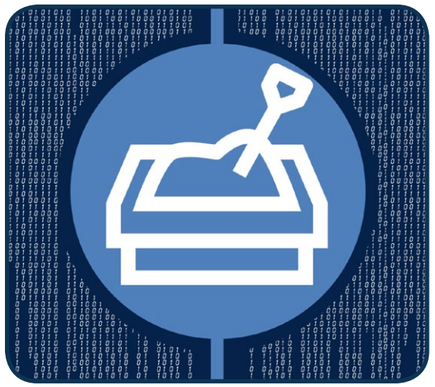
A full semester of 22 lectures in deep learning for computer vision is available as open courseware. Students can learn to implement, train, and debug their own neural networks and gain a detailed understanding of cutting-edge research in computer vision.

< READ MORE >

PREDICTING HOW HARDWARE WILL SPEED UP DATA CENTERS

A new analytical model can be applied in the early stages of a hardware accelerator's design to predict its effectiveness in speeding up a data center before ever being installed. Testing at Facebook has shown results with an error rate of less than 3.7%.

< READ MORE >



TOOL TO AUTOMATE POPULAR SECURITY TECHNIQUE EARNS DISTINGUISHED PAPER

A distinguished paper-winning project introduces a new technique that automatically constructs sand-boxing policies for applications that keep them from compromising other programs.

< READ MORE >

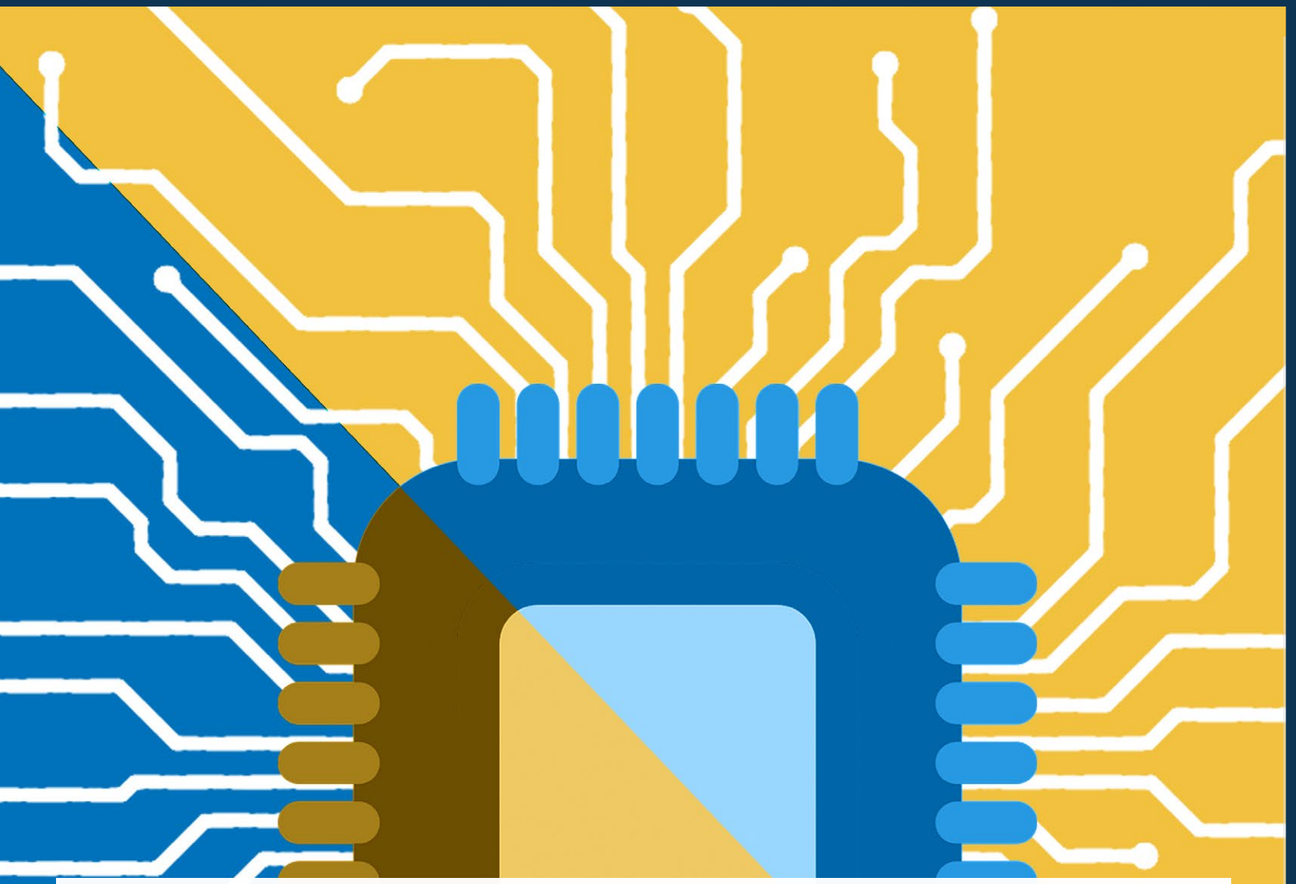
AFTER 52 YEARS, IBM APOLOGIZES FOR FIRING OF TRANSGENDER PIONEER

At a public event celebrating LGBTQ+ inclusion, IBM presented emeritus professor Lynn Conway with a rare lifetime achievement award. The award accompanied IBM's apology to Conway, which came 52 years after the company fired her for coming out as transgender.

Though not a household name like Thomas Edison, Nikola Tesla, or Alan Turing, Conway appears alongside them in Electronic Design's "Hall of Fame" for revolutionizing the microchip. It was at IBM that she made foundational contributions to superscalar computer architecture in the mid-1960s, including the innovation of multiple-issue dynamic instruction scheduling (DIS).

Later, at Xerox Palo Alto Research Center, Conway innovated scalable MOS design rules and highly simplified methods for silicon chip design, co-authoring the famous "Mead-Conway" text and pioneering the new form of university course that taught these methods – thereby launching a worldwide revolution in VLSI system design in the late 1970s.

< READ MORE >



AUTOMATIC CODE TRANSLATION FOR HARDWARE ACCELERATORS

A new technique developed in professor Westley Weimer's lab could enable broader adoption of post-Moore's Law computing components through automatic code translation. The system, called AutomataSynth, allows software engineers to tap into the power of hardware accelerators like FPGAs without specialized programming knowledge or needing to rewrite old, CPU-centric code.

With Moore's Law nearing its end, companies and designers rely on a number of hardware techniques to circumvent the diminishing returns provided by new CPUs. Among the most viable short-term candidates have been hardware accelerators like field-programmable gate arrays (FPGAs), which can be dedicated to rapidly executing particular common functions and eliminating bottlenecks in larger applications.

While their adoption by companies like Microsoft and Amazon Web Services is already well underway, FPGAs are limited in their use by programming requirements that are foreign to many software developers. These requirements also limit their use on pre-existing legacy software, which was typically written to work specifically with CPUs.

Most programs in use today have to be completely rewritten at a very low level to reap the benefits of hardware acceleration. Because of this, the components are being installed more rapidly than they're actually being utilized.

"Companies are taking steps to try to make [FPGAs] more approachable for people," says Kevin Angstadt, a PhD candidate leading the project, "but when it comes to writing new programs, the process is still very primitive."

< READ MORE >

SOLVING EMOTION RECOGNITION'S PRIVACY PROBLEM

Through the use of adversarial machine learning, researchers in CSE have demonstrated the ability to "unlearn" sensitive identifiers from audio, instead using stripped-down representations of a person speaking to train emotion recognition models.

< READ MORE >

RESEARCHERS REPORT >$11M IN RESEARCH GRANTS IN Q1 FY2021

The awards were distributed to 18 different primary investigators for a range of investigations into subjects including automated verification of distributed systems, lattice cryptography, AI education, and tracking of mental health symptoms.

< READ MORE >

HELPING ROBOTS THINK LIKE HUMANS WHEN SEARCHING

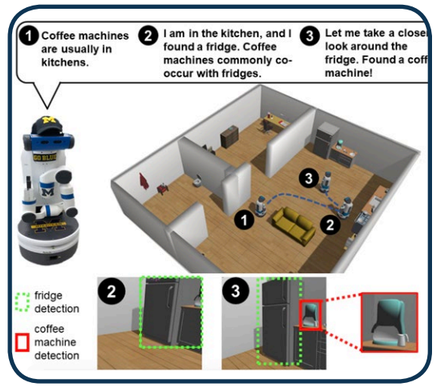
A research team has shown that robots can learn how to find things faster by learning how objects around the house are related, like looking for a coffee pot nearby if they see a refrigerator.

< READ MORE >

TEACHING AI HOW PEOPLE MOVE WITH INTERNET VIDEOS

A research team has demonstrated a simple but highly effective method for enabling neural networks to self-train on unlabeled videos with only partial views of human poses.

< READ MORE >



BUILDING A *Testing-Free* FUTURE

CSE researchers are assembling the puzzle pieces to make formal verification a central stage of software development at every scale. The powerful technique uses mathematical proofs to demonstrate that software meets specifications for safe, correct execution without the need for traditional testing. Thanks to a slew of recent findings, their labs' work now runs the gamut of leading-edge applications in the field — automating the use of formal verification, making the methods compatible with distributed and parallel systems at every level of abstraction, and even applying its logic directly to compiled binary code. The crew aims to build out libraries of the most complex and widely used pieces of enterprise software that are airtight, secure, and ready for deployment.

It's common sense: when you write software, you check your work. Testing is a carryover solution from all the other tools we've engineered throughout history, and it's a cornerstone of quality software design. And yet, as a standalone approach, testing is lacking: it's time consuming, it's labor and resource intensive, and most importantly it's extremely difficult to do exhaustively, leaving deployed software susceptible to unexpected inputs and behaviors.

In fact, most software in use today is so complex that testing it exhaustively is practically impossible.

"It would take billions and billions of years to run perfect test cases that cover all the behavior of reasonably large programs," explains assistant professor Baris Kasikci, "and if you leave certain things out, bugs can creep in."

But many critical systems rely in large part on testing to ensure that they're safe and reliable, leaving the question of bugs not an "if" but a "when."

"We're playing with bad software," says professor Karem Sakallah. "Right now, most of the software that we use is brittle and breaks at some point."

Researchers like Kasikci and Sakallah envision a smarter alternative. Rather than having humans, with all our imaginative limitations, come up with test cases, why not let math do the heavy lifting? Enter formal verification, a means to demonstrate that a program or algorithm is correct and reliable with all the elegance of a logical proof.

"Formal verification is an alternative here," assistant professor Manos Kapritsos explains. "I'm not going to try all the possible inputs and outputs; I'm going to prove, by converting my program into a series of logical formulas, that a property I specify holds at the end of my program."

That property, called a specification, is designed to describe how the program is allowed to behave. Essentially, a programmer has to be able to describe what the different good outputs of a program look like. The goal then is to prove that it's impossible for the program to behave otherwise.

"Having a foolproof system that says: you develop it, you check it automatically, and you get a certificate of correctness," Sakallah says, "that's what gives you confidence that you can deploy a program without issue."

Making it worth your while

A common question about formal verification is why it isn't just used on everything, if it's supposed to be so airtight. Unfortunately, while testing is certainly time consuming, writing proofs about a piece of code or an algorithm is usually even worse on that front — so far, anyway.

"Verification is associated with a great deal of time and energy," says Upamanyu Sharma, an undergraduate alum who did research with Kapritsos, "and for less critical applications, program testing and some static analysis is usually deemed sufficient."

That's why a primary early motivation for Kapritsos, Kasikci, and Sakallah, working with assistant professor Jean-Baptiste Jeannin in Aerospace Engineering, was automating the process.

To understand the difficulty, you have to look at what's involved in verifying a program. As mentioned above, the first step is to write a specification or a description of what the program's correct behavior should look like. It should be clear whether any specific program state, or the values a program has stored following each instruction, follows the spec or not.

>>

The goal is to demonstrate that the spec is true for all program states that can be reached in all possible executions. For example, to prevent collisions at an intersection, a program for a traffic light controller must never end up in a state with a green light going all directions. Logically, a property that holds in all cases like this is called an invariant. Showing that this invariant holds without exhaustive testing requires a proof by induction. Coming up with this inductive invariant is the key step in these proofs.

"You prove properties based on the structure of the code, and you make an inductive proof," Kasikci explains. Induction is a core concept in mathematical and computational proofs. You demonstrate two things: first, that the property holds in the program's starting state; and then, that the property is maintained each time the program's state changes.

If the property holds for every possible state, you've got an invariant, proven by induction.

"Manually finding the inductive invariant is a tedious and error-prone process and is, in fact, one of the biggest challenges in formal verification," Sakallah explains.

That's what makes automation of this process so powerful — the researchers maintain that it will be a key to the practice's broader adoption.

Automatically deriving inductive invariants is achieved in most cases with tools called model checkers that construct and reason about mathematical formulas describing the code and the spec. Model checkers have been very successful in the hardware space for verifying the safety and correctness of chip designs; a model checker designed by Sakallah called AVR received the top award at last year's Hardware Model Checking Competition.

TWO WAYS TO VERIFY THAT YOUR CODE IS BUG-FREE

Interactive theorem prover

Helps simplify some steps of writing a formal code proof, which are often more rigorous than the proofs mathematicians write. They're much more expressive than model checking tools, but require huge human manual effort to guide the interactive system. Early theorem provers (which originated in the 1970s) simply helped humans check and verify proofs, while more sophisticated ones prove parts of the theorem on their own.

Model-checker

Achieves complete coverage of a program's state space without actually enumerating the states individually, by using symbolic analysis to reason about sets of states, and performing fast algorithmic checks with so-called satisfiability solvers, to determine that no good state is ever one step away from a bad state. Coupled with aggressive abstractions that eliminate irrelevant details, it has been shown to scale to fairly large hardware verification problems and is increasingly making impact in the verification of software programs.

The lessons learned from these hardware verification problems have been quite helpful in designing software verifiers. Until very recently, however, the use of automation on software was limited to rudimentary problems.

"Adoption that we're looking at is mostly for very well-defined implementations," Kasikci says. "You're talking about quicksort algorithms that you can write in 20 lines of code. We have tools that can automatically prove that there is no bug in that code."

Basic data structures, basic algorithms — that was the level of complexity possible in discussions about automated verification when Kapritsos and Kasikci began their work. "They're still very important, because you rely on these things all the time when building software."

But the researchers are scaling things up. Their work over recent years has focused on adapting these tools to more complex classes of software that were previously out of reach.

An illusion of complexity

The first series of breakthroughs for the group was focused on the problem of automatically verifying the protocols that define how large, distributed computing systems communicate and cooperate. This problem brought together the challenging components of automatic verification and complex distributed systems that are very difficult to analyze.

To solve it, the researchers first made a key separation. Rather than package the abstract distributed protocol together with its actual implementation in code, they attacked the protocols alone.

"A protocol is pseudocode, or an algorithm," Kasikci explains. "It's a high-level description of how a system should behave." Taking that pseudocode and writing it into a real implementation introduces a lot of additional complexity.

By removing the complexity of implementation, they can formally verify a protocol and demonstrate its correctness in general before it's called upon for a particular use case.

"Even if you don't go into the implementation, reasoning about the protocol is very important," Kapritsos says. "If the protocol has a flaw, then of course the implementation would have a flaw. Before you pay developers to spend time implementing a protocol, you make sure that the protocol is actually correct."

A good example to understand what these distributed protocols do is the Paxos consensus protocol. This system is implemented widely to give the illusion of a single correct machine executing user requests, while in reality they're being handled by a large, complex system made up of many machines that can fail. Think of a bank with multiple branches and ATMs. For the user making a transaction, there might as well be a single bank server that gives them the same results no matter which ATM they use. But behind the scenes, there are many machines on that bank's network that have to remain in agreement about the transaction. Further, this illusion has to stay in place even if one or more of the bank's servers is out of commission at the time.

The issue with formally verifying protocols like this one is pretty straightforward: they're huge. The networks under consideration are widespread and can involve a theoretically limitless number of servers.

The group's solution came from a unique feature of these protocols inspired in part by Sakallah's background in hardware analysis. That feature was symmetry, and the insight paved the way for the group's most important breakthrough — what if distributed protocols aren't actually as complex as they look?

"Human-created artifacts seem to have some structure to them that make them tractable," Sakallah explains. "People do not create random things, they create very complex, but highly-structured, things: an airliner, chips, computers, software programs."

But something is lost between the period of creation and analysis, Sakallah continues. Making sense of a complex finished product seems overwhelming, but people had to actually make it somehow. "If you bring the insight that went into creating the artifact, you can use it to make the analysis of it tractable."

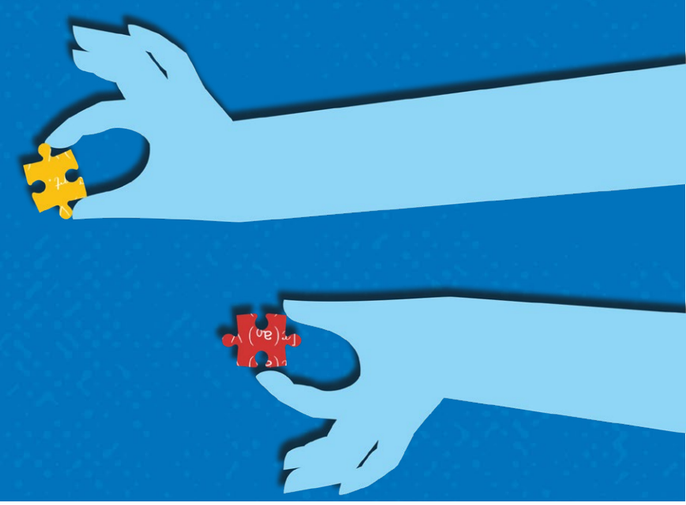
Formally verifying a distributed protocol like Paxos may involve a theoretically limitless number of "nodes" (individual servers or other entities connected by the protocol), but, the researchers argue, there isn't anything special about each one. They're all essentially the same, and all play the same role in the protocol. They back up data, agree on all of the transactions taking place, handle things a certain way in the event of a node failure — and they all do this uniformly.

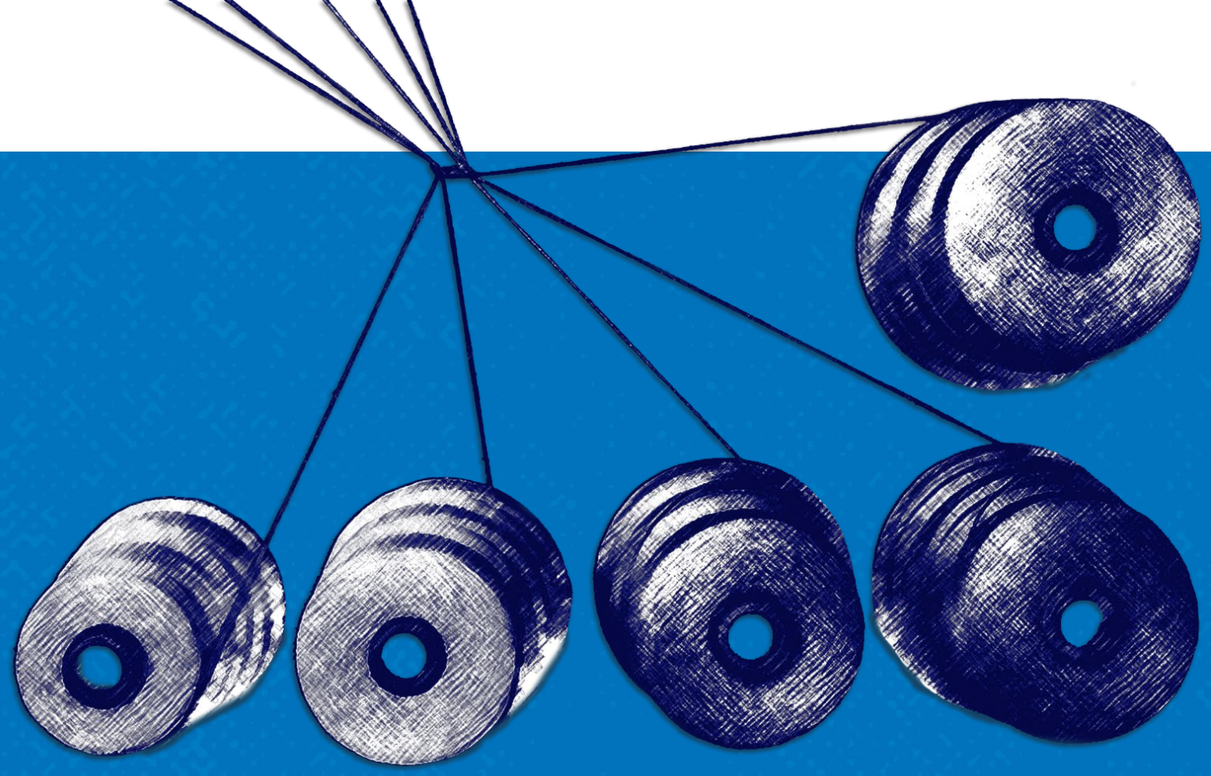
As a consequence, solving the problem for a small number of nodes could be seamlessly extended to a huge number of nodes without meaningfully changing the analysis. This was the key to automation, and became the basis for the group's protocol verification framework called I4. Their method made use of the Averroes or AVR verification system originally designed for hardware in Sakallah's lab.

"The essence of our idea is simple," they wrote in their paper, presented to the ACM Symposium on Operating Systems Principles (SOSP) in October 2019, "the inductive invariant of a finite instance of the protocol can be used to infer a general inductive invariant for the infinite distributed protocol."

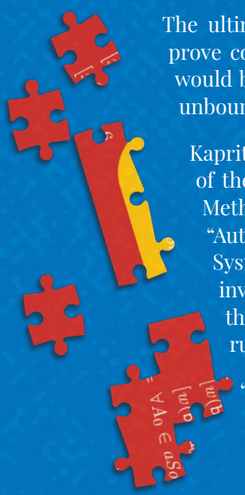
In the end, they were able to simplify the problem enough that they could use a model-checking tool to verify several distributed protocols with little to no human effort.

"Our goal in that project was to see if we could take the human out of the loop," says Kapritsos. With further development of I4, the researchers want any developer to be able to prove a distributed protocol that they don't necessarily understand very well themselves. >>





Sakallah and Aman Goel, a PhD student in Sakallah's lab, undertook one such further development in 2020 with their project IC3PO. This tool, like I4 before it, is an extension of the AVR hardware verifier that takes fuller advantage of the symmetry and regularity observed in distributed protocols. It uses that symmetry to significantly scale their verification, make it more automated, and produce inductive invariants that are compact enough for a developer to read and understand afterwards.



The ultimate goal of IC3PO is to automatically prove complex protocols, such as Paxos, which would be a major advance in the verification of unbounded, or infinite, systems.

Kapritsos and Kasikci are also taking this arm of the effort further with a new NSF Formal Methods in the Field grant for their project "Automating the Verification of Distributed Systems." The proposed research will investigate a new approach for automating the verification of complex software running on multiple machines.

"This project is meant to push the idea of I4 further, into more complex protocols and implementations," Kapritsos says.

The nitty gritty

With the work on automatically verifying distributed protocols off to an ambitious start, that still left the actual messy implementations of these and other systems in need of exploration. Kasikci and Kapritsos took several steps in this direction throughout 2020.

Kapritsos and a team of collaborators published a tool called Armada at the 2020 ACM Conference on Programming Language Design and Implementation (PLDI) that targeted the semi-automatic verification of another complex class of programs, concurrent programs.

Concurrency has been a vital tool for increasing performance after processor speeds began to hit a plateau. Through a variety of different methods, the technique boils down to running multiple instructions in a program simultaneously. A common example of this is making use of multiple cores of a CPU at once.

Formal verification on these programs is notoriously difficult, even lagging by a decade behind other verification work, according to Kapritsos and undergraduate alumnus Upamanyu Sharma, both co-authors on the paper.

"The main challenge in concurrent programs comes from the need to coordinate many different threads of code together," Sharma says. "To verify that multi-threaded programs are correct, we have to reason about the huge number of possible interleavings that are possible when multiple methods run at the same time."

To date, a variety of proof methods have been designed to deal with different types of concurrency. In this project, the researchers set out to design a single framework that allows a user to apply many of these techniques to verify a single program.

With Armada, programmers use a C-like language to select a proof technique and provide some annotation describing how it should be applied to the program they're trying to verify. From there, the proof itself is generated automatically and ready to be run through a prover for verification. In the event the proof fails, the user changes their annotation or working code and generates a new one.

In the world of verifying concurrent programs, this is to date the most low-effort technique available. The authors hope that this shorter pipeline will encourage the broader use of verification outside of the most critical systems where the technique is already justified.

Kapritsos is also in the early stages of expanding the work from bug-proofing to another major pain point in software testing, performance. This as-yet untread territory could remove a major hurdle between eliminating all testing from the software development pipeline.

On another front, Kasikci was awarded a grant from DARPA to adapt formal methods to a unique new setting — small patches to compiled binary code. This project, called Ironpatch, will target complex systems already in deployment, like cars, ships, and rockets, that are so dense and heterogeneous that they're difficult to patch through traditional means.

"It's a little bit of a mess," said Kasikci. "Traditionally, you fix the bug in the source code, you rebuild the software and you redeploy it. But these moving environments are really hostile to that model because there's a lot of different software and lots of different kinds of computers."

Ironpatch takes a different approach, bypassing the source code and instead making tiny modifications called micropatches directly to the binary heart of the running software. This eliminates the need to recompile the software and because the changes it makes are so minute, they can fix problems without causing others. Ironpatch is designed to be self-contained—once a vulnerability is identified, the system will automatically generate, formally verify and apply a micropatch to eliminate it. This also eliminates the need to upload software patches to a remotely located system, a particularly handy feature when that system might be located on a spacecraft millions of miles away. Kasikci will be working on Ironpatch jointly with Kapritsos and professor Westley Weimer.

Verifying the future

These projects are just the beginning; the long-term ambitions of the researchers are much more thorough.

"For me there is a broader theme here — it's bringing formal verification closer to practice," Kapritsos says, "bringing it closer to real developers by making it more useful and easier to perform."

That means more than speeding up a testing alternative, it means a whole new paradigm in security and reliability. With formal verification as a universal stage in the development pipeline, it will enable entire libraries of complex, reusable code that is certified safe and ready to deploy.

"The key idea is composability," Kasikci says. "It's one of the fundamental ideas in computer science. We have little things, we put them together, and we build bigger things. We have functions, we put them together, we have programs. The beauty of formal verification is that when you prove a property about a function, you prove it for all input and output combinations. It doesn't matter what you compose with it, the property's still going to hold. If you have a bunch of these things where the property you're after is proven, then you can compose them to make something larger for which the property will hold automatically." //

PROJECTS REFERENCED IN THIS ARTICLE

- "Armada: Low-Effort Verification of High-Performance Concurrent Programs" < PAPER >
- "Automating the Verification of Distributed Systems"
- "I4: Incremental Inference of Inductive Invariants for Verification of Distributed Protocols" < PAPER >
- "Ironpatch: Automatic Generation of Assured Micropatches"
- "One-Click Verification of Distributed Protocols with Symmetry-Aware IC3"

PRECISION HEALTH

IN THE PALM OF YOUR HAND

Precision health is an approach to wellness that takes into account variability in genes, environment, and lifestyle for each person. And thanks to advancements in technology, it's here today.

Huge amounts of data are being collected and analyzed to manage our care, with data sources including laboratory tests, biometric sensors, patient records, hospital data, and more. But results can be slow in coming, and the wait between testing and diagnosis can be days or weeks.

However, recent breakthrough developments in technologies for real-time genome sequencing, analysis, and diagnosis are poised to deliver a new standard of personalized care.

Imagine a case in which a patient is admitted to a clinic and a simple blood or saliva test is administered. Before the visit is over, a complete diagnosis and personalized treatment plan is available. In another scenario, a surgeon who is seeking to remove a tumor with minimal impact to healthy tissue could confirm decisions through real-time tissue sample analysis. Finally, picture a portable pathogen detector that could alert a user to dangerous exposure during a pandemic or disease outbreak.

The key to making these and other visions real would be a handheld device that provides real-time genomic sequencing and analysis of patient DNA or pathogen DNA or RNA.

>>

Handheld sequencer is technology driver

The Oxford Nanopore MinION is a real-time, handheld genomic sequencing unit. Analyzing and parsing the data produced by the MinION is the next challenge. | Photo: Oxford Nanopore Technologies

Advances in genomic sequencing

It cost nearly $3 billion to sequence the first human genome in 2001. Today, the cost to sequence a whole human genome is under $1000 and expected to reach about $100 soon. In addition, first- and second-generation sequencing systems were large, expensive, and designed for batch operation. Results would become available days or more after samples were taken. But new, lower-cost third-generation sequencing systems now exist, such as the Oxford Nanopore MinION, which can rapidly sequence individual samples and fit in the palm of your hand.

The human genome is made up of over three billion base pairs of DNA. To sequence a genome, the MinION employs small nanopores to divide a collected sample into billions of strands, called "reads."

"The MinION is a great handheld sequencing tool and is capable of rapidly sequencing biological data," says Reetuparna Das, an associate professor in CSE. "It takes the chemical sample, divides the DNA or RNA into strands, and sequences those strands into electrical signals, known as 'squiggles.' However, it does not have the compute capability to analyze raw data in the field and quickly produce actionable results."

All that stands between us and real-time diagnosis is a computing system that can analyze the sequenced data and provide treatment and therapy recommendations before the patient even leaves the office.

The computing challenges

In what is known as secondary analysis, it is the job of a computing system to interpret squiggles as base pairs of DNA, a process which is known as basecalling. A base pair is essentially one rung on a DNA or RNA structure's ladder. Following that, the system must align the read data to genome reference data and then identify variants between the sample and the reference. The variant data of human genomes is used to identify a genetic disease marker. Sequencing is also used to identify pathogens by aligning DNA or RNA strands to a reference pathogen database and using metagenomic classification tools.

And although this sounds straightforward, sequencing produces about GBs to TBs of data and the processing challenges are steep because of the precision, complexity, and scale of the task. Two multidisciplinary teams of researchers at U-M are working on approaches to overcome this hurdle.

Associate professor Reetuparna Das and professor Satish Narayanasamy, along with professor David Blaauw in Electrical and Computer Engineering, are leading a team funded by the National Science Foundation and the Kahn Foundation that is developing a hardware/software platform to accelerate next-generation genomic sequencing with a focus on pathogen detection and early cancer detection. In this effort, they are collaborating with associate professor of internal medicine and of microbiology and immunology Robert Dickson and assistant professor Carl Koschmann of pediatrics, as well as with associate professor Jenna Wiens, who is also a part of the second research team.

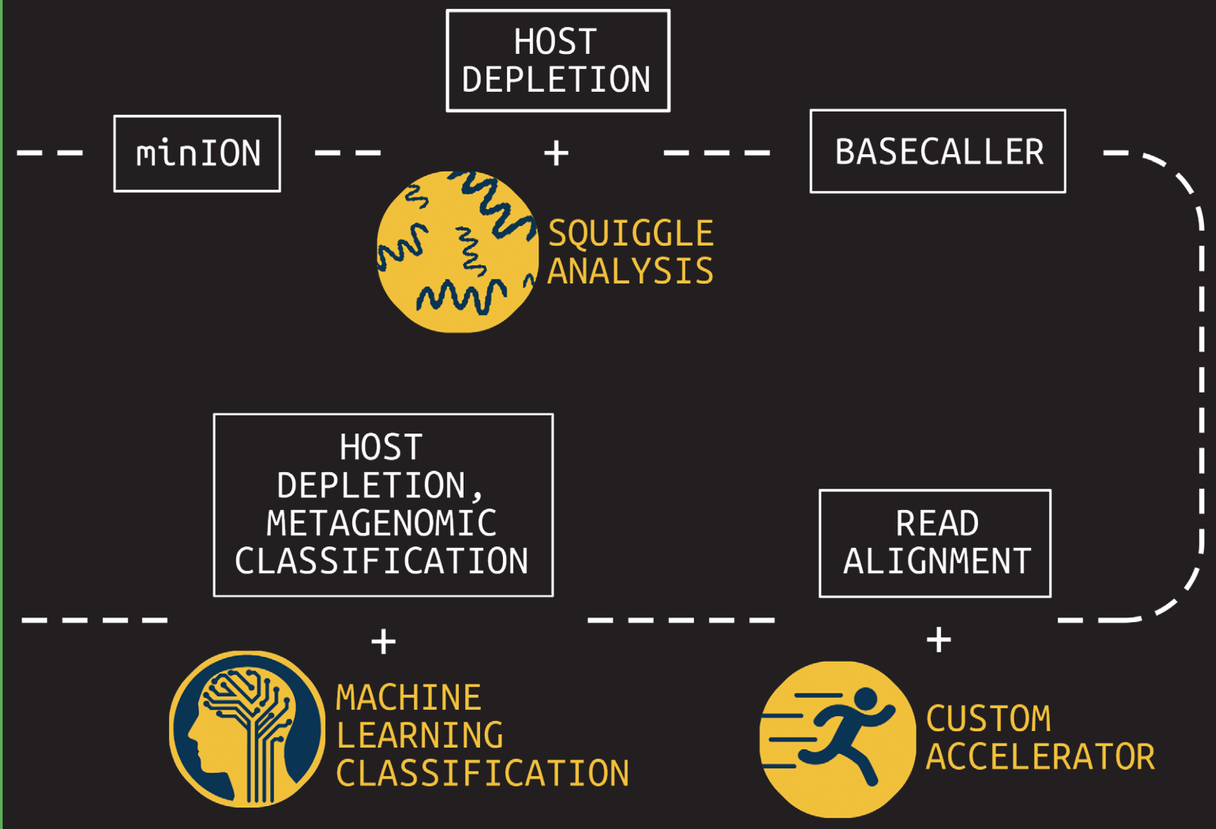
The second team, funded by the Kahn Foundation, is developing data acquisition and machine learning techniques to dramatically improve the prediction, treatment, and management of disease in aging populations. A key component of this effort is the use of machine learning to speed metagenomic analyses.

This large-scale interdisciplinary effort is a collaboration between researchers at Technion – Israel Institute of Technology, the Weizmann Institute, and U-M. The U-M researchers are led by Betsy Foxman, professor of epidemiology at the School of Public Health. Wiens, who is also a Co-Director of U-M Precision Health, is a Co-PI for the U-M research group.

>>

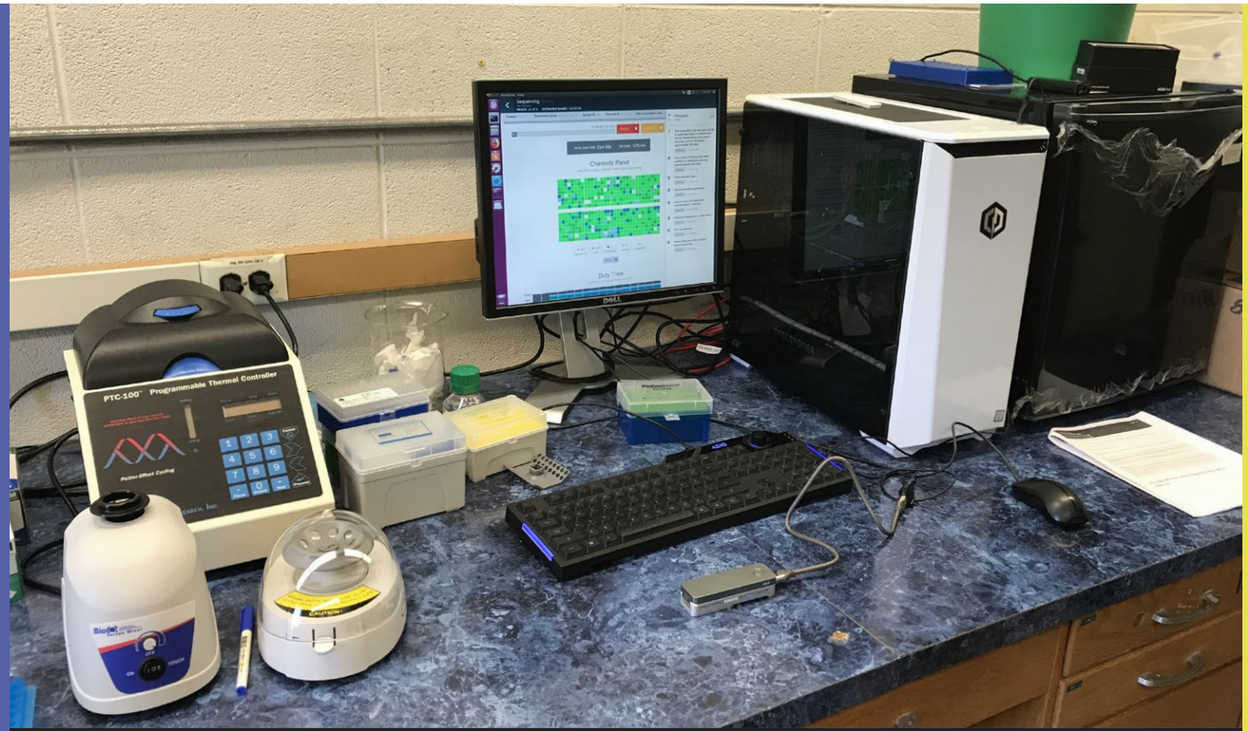
Creating a pipeline to accelerate analysis

Researchers at U-M are working to develop a pipeline that will accelerate the ability to analyze sequencing data and provide actionable insights. The goal is to provide these insights within a short time of the sample being collected. The pipeline includes squiggle analysis before basecalling, custom hardware to accelerate read alignment, and machine learning for host depletion for purposes of microbiome analysis.



A faster pipeline for analyzing sequenced data

An Oxford Nanopore MinION in the researchers' lab enables rapid, mobile genome sequencing. U-M researchers are working to accelerate the efficiency of downstream genome and microbiome analysis. | Photo: Reetuparna Das



An accelerated computing platform for genomic sequencing

Blaauw, Das, and Narayanasamy are focused on dramatically accelerating and optimizing the pipeline to process data from the MinION. The goal, say the researchers, is to reduce the time required to analyze a sequenced genome from hundreds of CPU hours to a matter of minutes.

"To realize the full potential of genomic sequencing," says Das, "computing power needs to increase by orders of magnitude."

The problem is, that's not possible under traditional processor roadmaps, where additional transistors and cores are packed ever more tightly into a processor for incremental processing gains. Added additional programming cores won't solve the problem either.

"Sustainable growth of processor performance is only possible with custom layers including hardware, software, and algorithms," says Das.

There are a number of areas of inefficiency that occur during secondary analysis which the team is addressing.

First, says Das, is the read alignment process, during which read data is aligned to genome reference data. Read alignment is composed of two steps: seeding and seed extension.

Seeding finds a set of candidate locations in the reference genome where a read can align. Possible matches are known as hits in the reference. In seed extension, for a read, the reference strings at the hit positions are matched at the read. With current technology, this takes hundreds of CPU hours for a whole genome.

For seeding, the researchers discovered a huge memory bandwidth bottleneck. They did hardware/software codesign and developed a new algorithm, data structure, and index that trades off memory capacity for memory bandwidth. They then built a custom accelerator that traverses the new index efficiently to find hits and seeds. The seeding algorithm has been released as open source software and is planned to be integrated with state of art alignment software from Broad Institute and Intel.

For seed extension, they built a systolic array that would in a few hundred cycles use approximate string matching to match read and reference data.

The researchers have developed a custom ASIC to eliminate the throughput bottleneck by using a pruning algorithm to optimize the alignment of DNA reads and candidate mutations and by reducing floating point computation by 43x when tested on real human data.

These enhancements and others have been mapped to custom hardware. This includes an accelerator for seed extension which achieves 2.46M reads/second, a ~1800x performance improvement, and a 27x smaller silicon footprint compared to a Xeon E5420 processor.

According to the researchers, when run on a high-end 56-core server in the Amazon cloud, their secondary analysis tools will take about six hours for whole genome sequencing. On an Amazon FPGA server, this reduces to about 20 minutes. When run on the researchers' custom hardware, processing time is about a minute.

The team has also developed techniques to optimize the read process for pathogen detection. One is to quickly analyze the beginning of a read to determine if it is host or pathogen material. If it is host, the remainder of the read can be skipped since it is only the pathogen material that is of interest. In addition, the researchers are often able to accomplish this host vs. pathogen differentiation using machine learning on squiggle data, without the need for resource-intensive basecalling.

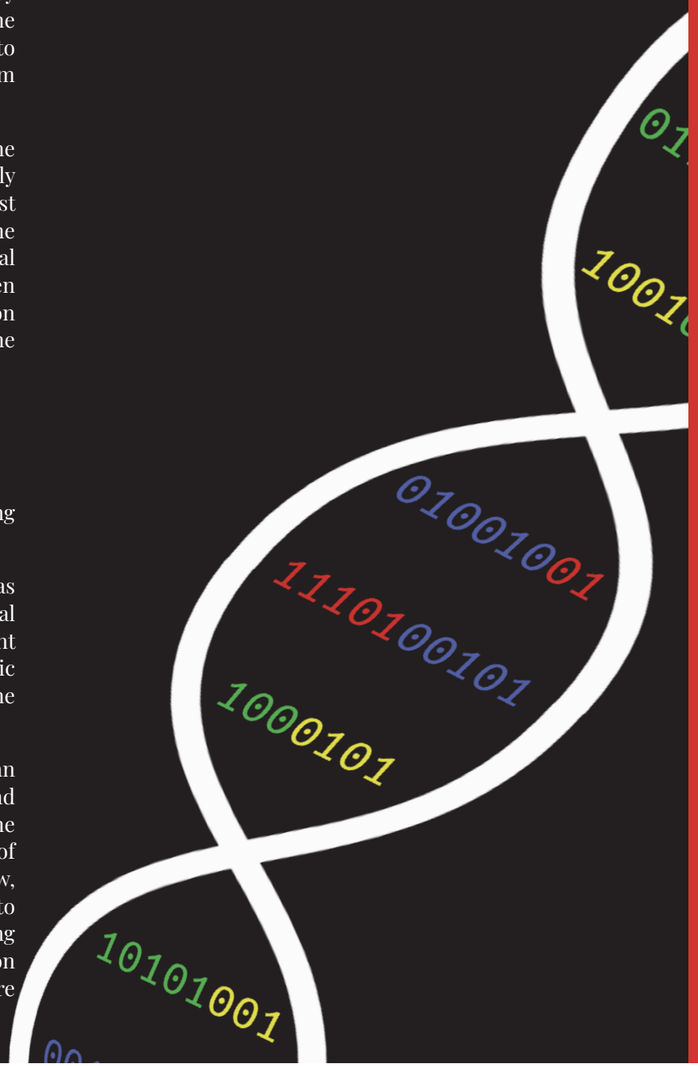
Microbiome analysis to provide faster insights

When processing clinical samples, a fast data processing pipeline is key to the delivery of actionable insights.

"In clinical samples, much of the data — sometimes as much as 90% — can be host DNA, rather than microbial DNA," says Meera Krishnamoorthy, a PhD student working with Wiens. "As a result, existing metagenomic classification tools store a lot of information about the host, and this can get computationally inefficient."

In collaboration with a team of researchers in Michigan Medicine and the School of Public Health, Wiens and Krishnamoorthy are working on in-silico machine learning approaches to host depletion, or the removal of host data reads, which will become a part of Das, Blaauw, and Narayanasamy's custom hardware. Their goal is to remove all of that host data before classification allowing downstream microbiome analyses to focus solely on microbial data. Existing host depletion methods are laboratory based and can be resource intensive to perform.

In contrast, Krishnamoorthy and Wiens' approach is computational and does not rely on large reference databases, but instead is based on a convolutional neural network. It takes as input read output by the basecaller and then after a series of convolutions and pooling steps outputs a prediction regarding whether or not the read pertains to the host. The proposed approach proposes to increase the efficiency of downstream analyses, enabling microbiome research that has the potential to transform future medical care.



LEARNING TO BUILD ACCESSIBLE SOFTWARE SYSTEMS OVER ZOOM

Roommates taking EECS 495, Software Development for Accessibility, work together off campus in Ann Arbor, MI on August 13, 2020.

The course is taught by Dr. David Chesney, and emphasizes team-based development of large, complex software systems using established development methodology. Students typically build accessible software systems for individuals with a disability or illness. < VIDEO >

Because of the coronavirus pandemic, Chesney moved the course online and refocused it to tackle issues around COVID-19. By doing so, he provided his students with the opportunity to build and find themselves during an uncertain semester.

How CS is changing EDUCATION

From K-5 to high school and college, CS is reshaping how our students learn. Michigan researchers are taking on the big challenges to integrating computing into everyone's education.

Professor Mark Guzdial wants more people to use computing in their work and lives.

"Programming is super powerful, and I want a lot of people to have access to that power," he says. "Computational literacy is critically important in our world today."

Guzdial, himself a U-M alumnus, joined the faculty at Michigan in 2018 after 25 years at Georgia Tech. Over the years, he's established himself as a leading voice in the field of computing education research.

"I'm interested in both the infrastructure that allows us to teach programming to people and to make things accessible in order to broaden participation in computing," says Guzdial. "But I'm also interested in building the tools and doing the laboratory experiments to come up with deeper insight into what's going on when students are learning about computing."

Computational media connects with students

While he was at Georgia Tech, Guzdial and his then-PhD student Andrea Forte (now on the Information faculty at Drexel) were evaluating how Georgia Tech students performed in a required CS literacy course. They recognized that before many students could succeed at computing, they had to see it as useful and connected to their lives. Forte made the observation that for these students, computing was not a matter of calculation but of communication: these students cared about digital media.

So they, along with Georgia Tech researcher Barbara Ericson (who is a Michigan alumna and is now an assistant professor at the U-M School of Information) developed tools to teach programming in terms of what Guzdial calls media computation: how to manipulate the pixels in a picture, how to manipulate samples in sound recordings, and how to manipulate the frames in a video.

It's relevant, says Guzdial, in the same way a biology class is. "Why do you take a biology class? Because you live in a world full of living things. Why do you take a class in computing? Because you interact in a world increasingly driven by digital media. You should know something about how it is created and structured and how it can be manipulated. That's what the course became."

>>

Transforming education

A second grade class in Kent City, Michigan works with collaborative, exploratory tools developed at U-M. | Photo: Billie Freeland

CS enables richer history lessons

An interactive tool allows students to develop and edit scripts to explore and visualize historical trends and events. < Explore the tool >

Not like the old books

eBooks include mixed up code exercises, in which students learn to identify and sort code blocks, in addition to other interactive and measurable features. < Explore the eBook >

And it worked. "That's when I realized," states Guzdial, "that we can change how and what we teach in computing. We can connect by making the tools useful and usable, for both teachers and students."

Fast-forward to 2020, and Guzdial is creating task-specific programming tools as a complement to media computation. In one project, he's creating a new programming language that strips away programming syntax and allows users to make a chatbot. He has created chatbots that act as a baby, a toddler, Alexa, and Lady Macbeth.

Teaching history with CS

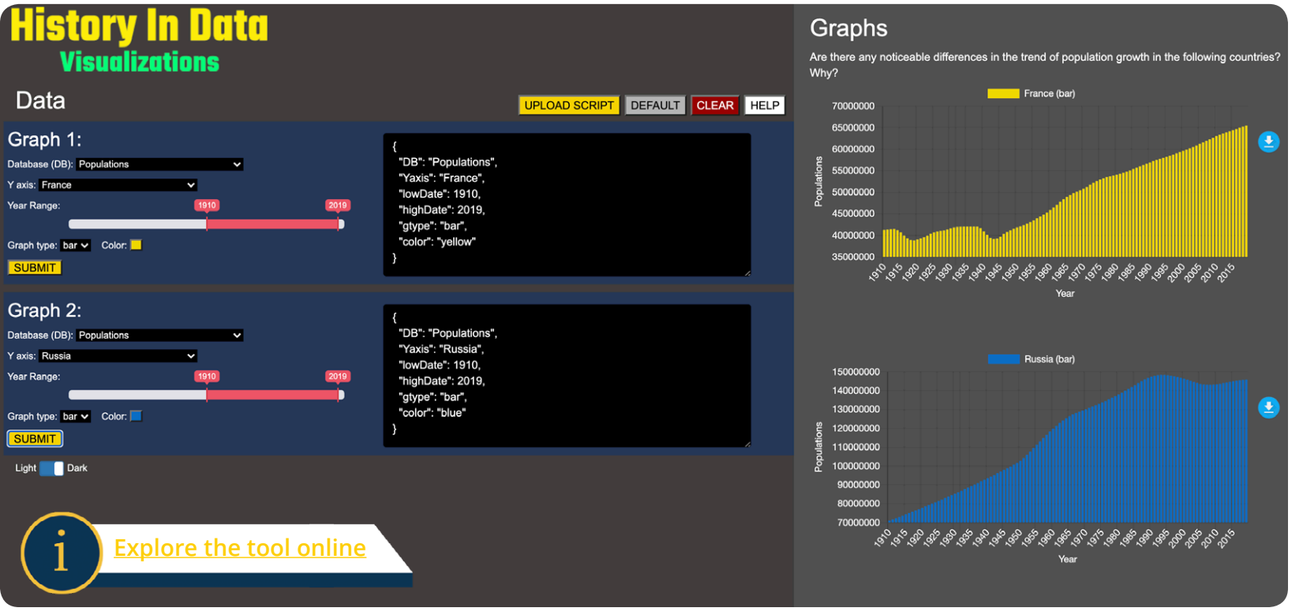
Under a grant from the National Science Foundation, Guzdial and Engineering Education Research PhD student Bahare Naimipour have proposed a new way to integrate the use of task-specific computer science tools into history courses. In this project,they are collaborating with Michigan alumnae Prof. Tamara Shreiner at Grand Valley State, who teaches data literacy to future social studies educators. Guzdial, Naimipour, and Shreiner involve Shreiner's new teachers in the process to develop curricula that meet teachers' perceptions of usefulness and usability. High school students who take their courses will learn data manipulation skills as a part of completing their history assignments.

"This project is being built into new, interactive course materials that will allow high school students to build data visualizations in history classes as part of an inquiry process. We use programs to capture the students' process as they investigate historical questions," explains Naimipour.

"By doing this,' adds Guzdial, "CS becomes a tool that students can get comfortable with as a part of their learning. A larger and more diverse range of students will get CS experience, and the use of data manipulation tools will allow them to explore history in a new way. We hope that this type of engagement will lead to a greater level of comfort with and participation in CS courses."

Over a three-year period, the project will track teachers from Shreiner's pre-service data literacy course at Grand Valley State University, into their field experience, and on into their in-service placement.

"For the project to be successful, teachers need support to feel comfortable with the new tools and to adopt them in their classes," notes Guzdial. "For this reason, at the end of the three year project, we will be able to describe the factors that influence adoption and non-adoption, so that we can iterate and improve."



eBooks integrate coding and learning techniques into lessons

"One of the things I've found is that high school teachers can only adopt new books about every seven years. So even if I taught them interesting new ways to teach computer science, they couldn't buy the books for some time," says Barbara Ericson, who's on the faculty at the School of Information. "So, that's part of why I've been working to develop free eBooks that anyone can use."

Ericson has collaborated with Guzdial on numerous occasions, including on a long-term project to improve computing education, from teacher certification to standards – and on media computation. For the media computation project, she wrote the Java book, the visualizer for sound, and other tools. She also wrote many of the exercises. She did a lot of professional development with teachers and got feedback on the tools.

Another project that they began working on together, but with which Ericson took the lead and ran with, is Ericson's eBooks project. It's a key part of her focus on increasing participation in CS – a subject she is passionate about.

Students like using interactive ebooks and studies have shown that they have better learning gains than with traditional practice.

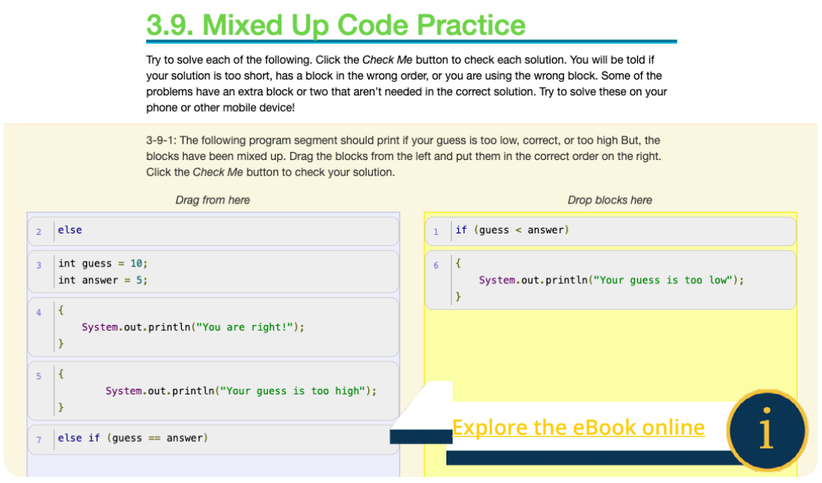
"We designed our eBooks based on educational psychology, where more people perform better if they have a worked example or an expert solution followed by similar practice," states Ericson. "So rather than just providing practice, we have worked examples of code that you can run, we explain the code with textual comments as well as through audio, and then it's followed by practice, including multiple choice problems, mixed up code (Parsons) problems, and clickable code.

"One of the things we've learned in our research is that people – even the teachers – don't like to read. They immediately skip to the problems and then go back to the text if they need to. So we don't have much text: we have some bullet points and examples, and then we go to the interactive examples."

Ericson can monitor how people use the eBooks through log-file analysis. She can look at which questions people are able to answer correctly, and where they struggle. This allows for continuous improvement.

"Interactive ebooks are going to be the future. They're much easier to change and improve," says Ericson. "And they could increase success in CS."

Ericson has developed three eBooks to date: two for high school AP CS, and a new one that is debuting at U-M this Fall in SI 206 (Data-Oriented Programming). James Juett, a Lecturer in CSE who is teaching ENG 101 (Thriving in a Digital World), has also created an eBook on MATLAB, following Ericson's approach, which he is using in his class this Fall. There are over 25,000 registered users for CS Awesome, Ericson's AP A eBook, and over 18,00 teachers in a users group.



A collaborative, digital platform for K-5 learning

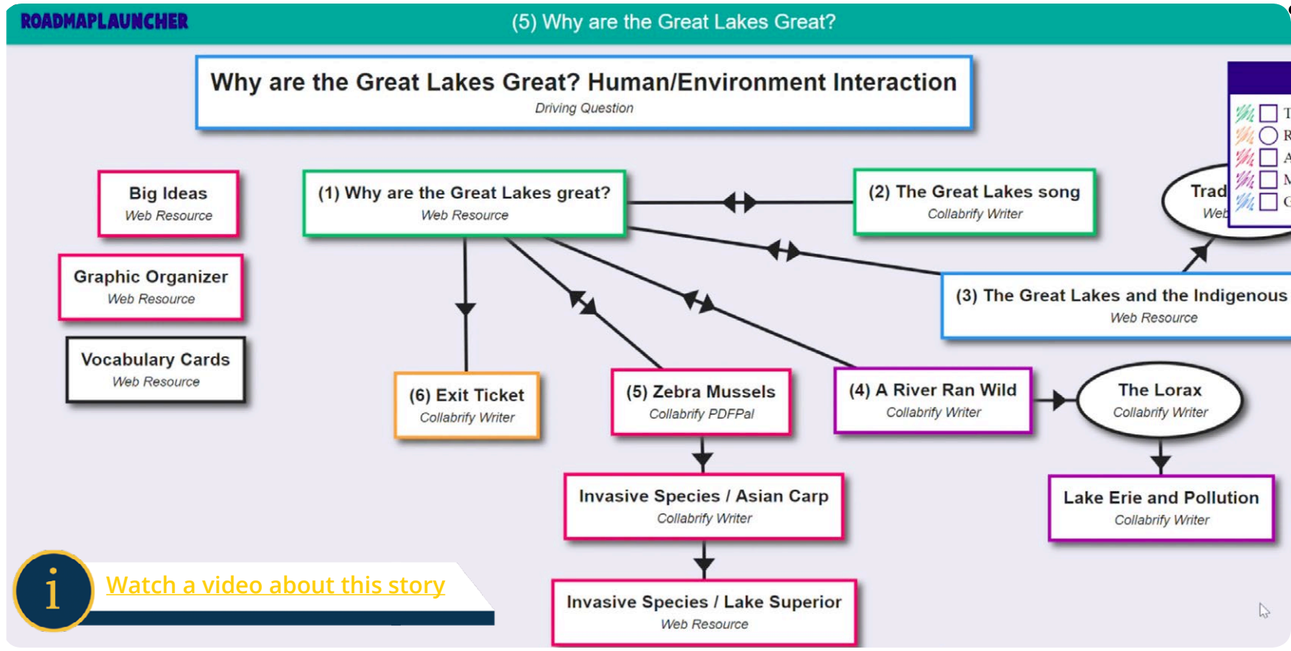
For many years, Arthur F. Thurnau Professor Elliot Soloway has been hard at work to transform K-12 education with collaborative digital tools. Now that we're faced with the COVID-19 pandemic, it turns out that his prescriptions are just what the doctor ordered. >>

Technology that inspires collaboration

Third-grade students in Mrs. Monique Coleman's class at Haas Elementary School, Genesee County, Michigan work together on a project. With the new voice over IP feature, the students can continue to collaborate remotely. | Photo: Liat Copel

Follow the roadmap

The Collabrify Roadmaps platform allows teachers to customize and share interactive lessons that students can explore. < Video >



Soloway has long advocated a more exploratory and collaborative approach to learning, enabled by technology. As far back as 1990, he was merging education and tech in a project at Ann Arbor's Community High School, where students conducted on-the-street interviews and used Soloway's tech to create full-color multimedia lab reports.

An AI researcher by training, Soloway recognized early on the need for a collaborator in the education space, and in 2011 he teamed up with Cathie Norris, Regents Professor in the Department of Learning Technologies at University of North Texas in Denton. The two have published, experimented, and advocated together since.

In 2012, Soloway and Norris began work on a digital learning toolset, called Collabrify, that has now emerged as a web-based platform they see for transforming how students and teachers interact to explore, learn, and guide.

Today, the Collabrify Roadmap Platform is the vehicle used to deliver curricula through Soloway's and Norris' new Center for Digital Curricula (CDC) at U-M, which launched in the summer of 2020 to provide free, deeply-digital standards-aligned curricula for K-5 (the area of greatest need, according to Soloway). And although the vision for collaborative online learning and its benefits pre-date the COVID-19 outbreak, Collabrify and the CDC have arrived at an opportune time.

The Collabrify Roadmap Platform is a set of free, customizable digital learning tools that allows students to work individually or in groups using mobile devices or laptop computers. Roadmaps provide teachers with scheduling templates that can be customized to include all the activities that would normally take place in their classrooms. The system guides students through the day, points them to the resources they need to complete their work and enables them to collaborate with teachers and each other. The platform also provides a searchable repository of online lessons developed and vetted by teachers, which is also distributed for no cost.

Wendy Skinner, who teaches second grade at Brandywine Community Schools in Niles, Michigan, says it's a significant improvement over other attempts at K-12 online platforms, which aren't designed to be as comprehensive, intuitive, or engaging.

"Roadmaps are the only thing I've seen where I can plug in my skills for my kids in the way that I'd do it in the classroom," Skinner says. "The kids have a schedule, and there's a nice visual finish and sequence to it. It's all in one place and I can monitor it and look at what they're doing. I can make sure that it includes the things that I value and make my connections with the kids meaningful."

While tools like Google Classroom can help streamline classroom logistics like grading and file sharing, Soloway explains that aside from Collabrify Roadmaps, there is no single, reliable source for managing digital curricula. "One of the important takeaways is that our center is providing not only the platform – we're also providing the standards-aligned deeply digital curricula that teachers can customize to their needs."

"A lot of schools have attempted to arm their kids with some type of digital device to assist in learning – a Chromebook or a tablet," says Don Manfredi, a mentor-in-residence at the U-M Office of Technology Transfer who helped launch the CDC. "But the content to actually take advantage of these devices was very far behind. So the idea was, let's provide teachers with a deeply digital curricula to make these devices more valuable to the kids. And with COVID-19, it's just gotten pushed to the forefront."

And what does "deeply digital" mean? It's not just a pdf of a textbook. It's the ability to use an integrated suite of technologies as part of a platform to plan, explore, interact, collaborate, and create.

The CDC relies on a core group of Michigan teachers who create and vet digital content in addition to their teaching duties. Skinner and over 100 other Michigan teachers are now using those materials, and some of them have stepped up to create new digital content which is now available on the repository.

The CDC recently received a grant from Twilio, which has enabled the ability for kids to quickly talk to each other. In a classroom, they can "Turn and talk" with a classmate. The new feature provides voice over IP to allow for the same type of spontaneous communication. "Some students could be in class, some could be at home, or they could all be remote," says Soloway. "The kids don't need phones – they can just talk to classmates through Collabrify."

Collabrify also provides the ability for teachers to monitor in real time what students are doing, and now with the talk feature they can also speak to students at the same time. Instead of handing in material for grading, teachers can interact with students to correct or reinforce in real-time, while the students are working on the assignment. Formative assessment like this has been shown to improve performance by 2-3X, according to Soloway.

Among the many challenges related to K-5 education this year is the ability to deliver curriculum electronically to students in a format that they can master," says Pamela Thomas, Elementary Principal at Kent City Community Schools. "Roadmaps has proven to be a powerful tool, yet simple enough for our youngest students to access learning. Teachers are able to create their own lessons, or access curriculum and share it with students in a way that is both engaging and easy to navigate.

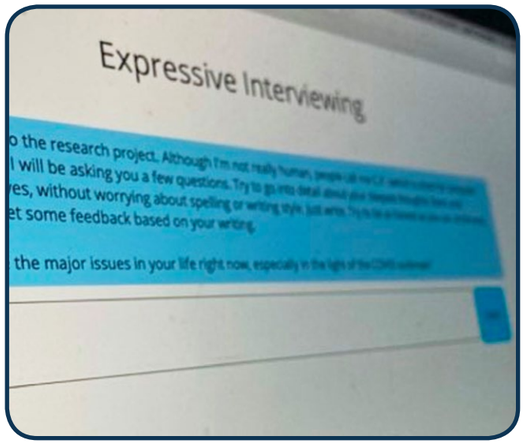
Students have responded well to the system, including seventh grader Jillian Biewer, who attends Marysville Middle School in Marysville, Michigan. She particularly likes the BrainVentures exercises, which offer an opportunity to learn across multiple disciplines and collaborate with friends even though she's isolating at home.

"When I'm doing a BrainVenture and I'm talking to people, it makes me feel closer in a way, like they're there right next to me doing it with me," she said. "It's almost therapeutic because when I'm doing it, I'm not worrying about what's going on around me, I'm not worrying about the news, I'm just learning in a fun way." //

>RESEARCHERS RESPOND: COVID-19

AI-POWERED INTERVIEWER PROVIDES SOCIALLY DISTANCED GUIDED REFLECTION EXERCISES

A virtual interviewer powered by natural language processing offers socially-distanced support for people facing trying times. The dialogue system takes inspiration from counseling strategies like motivational interviewing and expressive writing to guide users through written self-reflection.



"Unlike other systems that serve similar purposes, which frequently use multiple choice questions to progress in the dialogue, our aim is to have a natural dialogue where users can respond using free text," says Rada Mihalcea, Janice M. Jenkins Collegiate Professor of Computer Science and Engineering at U-M and co-lead on the project. "The goal of our system is to encourage individuals to talk about any difficulties they may have encountered due to the pandemic."

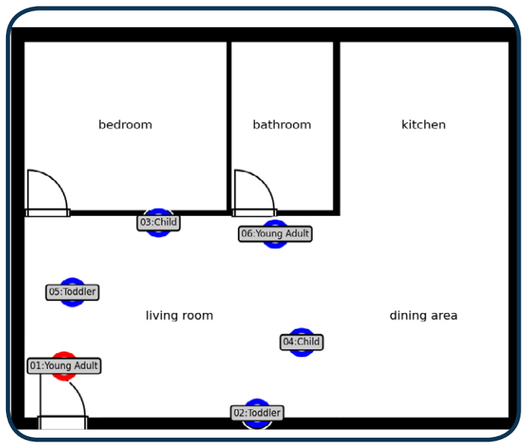
< READ MORE >

PREDICTIVE MODELING TO HELP US REOPEN MORE SAFELY

How many people can safely shop in the same grocery store? Are masks really important? Is six feet far enough? Increasingly specific social distancing questions are weighing on states and municipalities as they inch toward relaxing COVID-19 restrictions. Now, a team of computer science and medical researchers is working on a tool that could provide more precise answers.

The team is mashing up census data, virus transmission rates and decades of social science research. They've created a prototype that can visually model a household, enabling users to customize it with the size and type of dwelling, number of occupants and whether any of them is infected with the virus. They hope to make an open-source version available soon, enabling municipalities and other decision makers to evaluate how proposed policies would affect virus transmission rates in different types of households.

"We want to give policymakers the power to ask 'what-if' questions about the effects of specific policies and actions, before they're implemented," said assistant professor Nikola Banovic. "By trying out dozens or hundreds of possible interventions in a simulation, we could much more quickly find policies that would keep people safe while minimizing disruption to daily life."



Such a tool could inform more targeted guidelines or help governments target assistance to the types of households that are likely to need it most.

"A lot of guidelines that have been put in place tend to treat all households the same, but in reality, they're not the same," said Banovic. "A family of four that's living in a studio apartment, for example, can't social distance in the same way as a family that lives in a three-bedroom house."

< READ MORE >

BUILDING BETTER QUALITY CORONAVIRUS DATABASES

Amid a growing coronavirus crisis, experts in all fields have begun compiling massive datasets to track the impact of the contagion. These datasets capture everything from society-wide virus response information to medical needs data, available medical resources across the country, and buyer interest for medical equipment that could drive financing for new production.

To make constructing these datasets as accurate and timely as possible, associate professor Michael Cafarella is leading an NSF-funded project that will build high-quality auxiliary datasets to enable automatic quality checking and fraud detection of the new data. These safeguards are imperative to making sure coronavirus decision-making is driven by clean, accurate data.

Rapid analytical efforts by policymakers, scientists, and journalists rely on coronavirus data being complete and accurate. But like all dataset construction projects, those chronicling the coronavirus are prone to shortcomings that limit their effectiveness if left unaddressed. These issues include messy or unusable data, fraudulent data, and data that lacks necessary context.

Automatically checking coronavirus datasets against the pertinent, related datasets provided by Cafarella's team can make them more effective and insightful. For example, an auxiliary database about hospitals might contain data about the hospital's staff count, so a hospital resource allocator can test whether resources requested for coronavirus treatment are consistent with the level of staffing.

The team will build two large auxiliary databases. The "unified medical institution auxiliary database" will be a database of all known United States medical institutions, and will include rich background information for quality-checking, as well as an easy method for data integration. The "unified government office auxiliary database" will be a database of all known government offices in the United States, such as city halls, courts, or licensing offices, at any level of government.

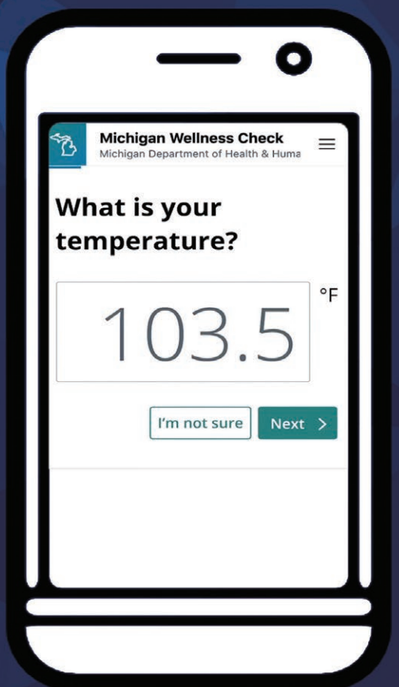
< READ MORE >

STUDENTS LEAD WEB APPLICATION TO HELP STATE CURB THE SPREAD OF COVID-19

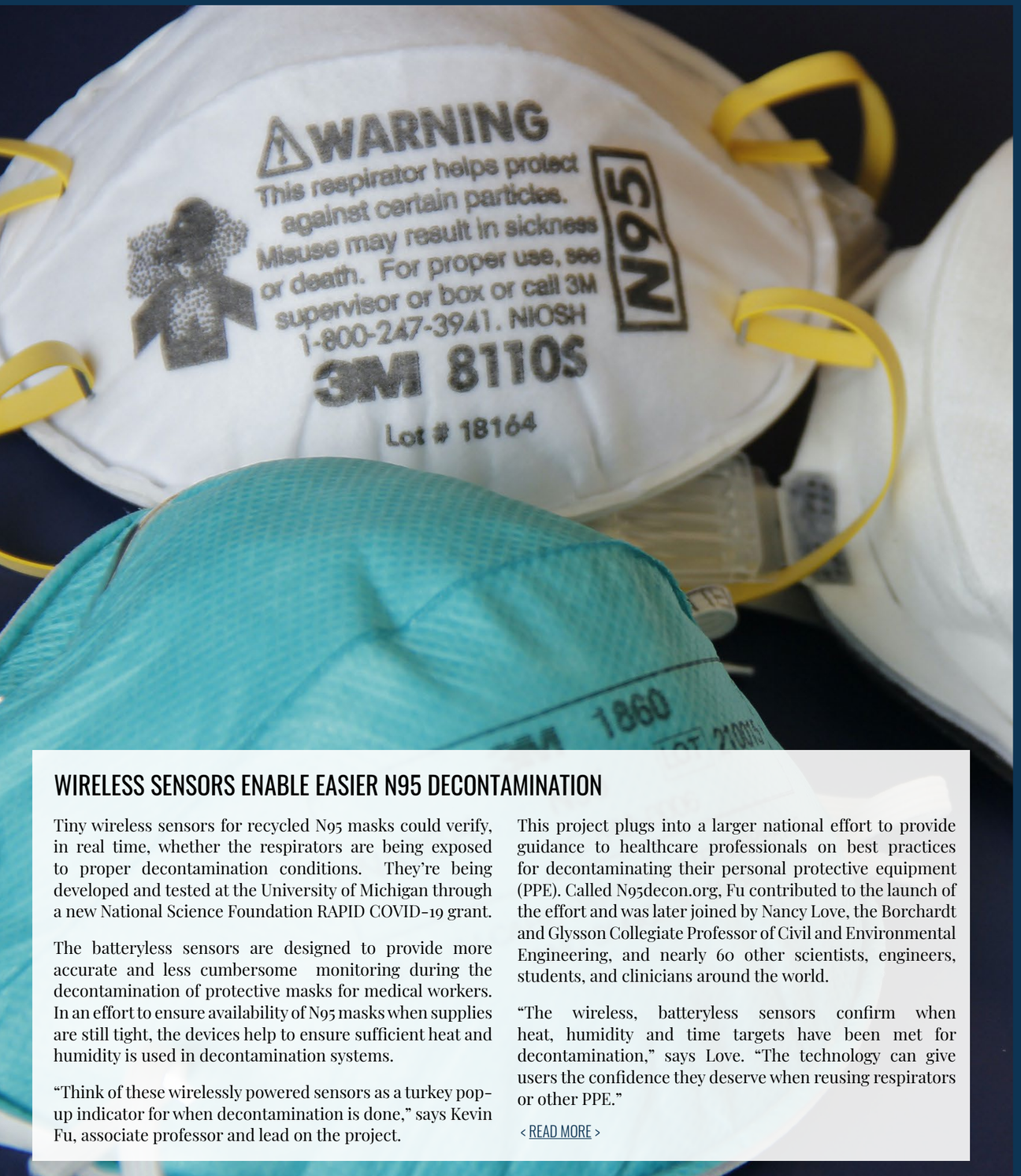
A team of students have developed an online tool to help Michigan residents track potential COVID-19 symptoms, ultimately enabling state officials and employers to make the right call about reopening workplaces during the pandemic. Called MI Symptoms, the web application plugs in to a broader effort at U-M to help the state safely ramp its economy up by surveying employees with a standard, recurring wellness check.

The effort was spearheaded by students from several classes at U-M, including EECS 441: Mobile Development for Entrepreneurs and EECS 497: Human-Centered Software Design and Development, with guidance from course instructors Sugih Jamin, Jeff Ringenberg, and Elliot Soloway as well as College of Engineering IT staff. The site was launched for public use on May 29, and more than 2,500 employers around the state have since put it to use, including the state of Michigan.

< READ MORE >



>RESEARCHERS RESPOND: COVID-19



WIRELESS SENSORS ENABLE EASIER N95 DECONTAMINATION

Tiny wireless sensors for recycled N95 masks could verify, in real time, whether the respirators are being exposed to proper decontamination conditions. They're being developed and tested at the University of Michigan through a new National Science Foundation RAPID COVID-19 grant.

The batteryless sensors are designed to provide more accurate and less cumbersome monitoring during the decontamination of protective masks for medical workers. In an effort to ensure availability of N95 masks when supplies are still tight, the devices help to ensure sufficient heat and humidity is used in decontamination systems.

"Think of these wirelessly powered sensors as a turkey pop-up indicator for when decontamination is done," says Kevin Fu, associate professor and lead on the project.

This project plugs into a larger national effort to provide guidance to healthcare professionals on best practices for decontaminating their personal protective equipment (PPE). Called N95decon.org, Fu contributed to the launch of the effort and was later joined by Nancy Love, the Borchardt and Glysson Collegiate Professor of Civil and Environmental Engineering, and nearly 60 other scientists, engineers, students, and clinicians around the world.

"The wireless, batteryless sensors confirm when heat, humidity and time targets have been met for decontamination," says Love. "The technology can give users the confidence they deserve when reusing respirators or other PPE."

< READ MORE >

FIGHTING PANDEMIC-RELATED FOOD INSECURITY IN DETROIT

As COVID-19 swept across the city of Detroit, it brought with it a wave of food insecurity, particularly among low-income residents and seniors who rely on public transportation and can only afford to buy small amounts of food at a time. Now, a U-M research team has stepped in to help identify solutions.

Funded by a National Science Foundation RAPID grant, they aim to help the city identify the most seriously affected areas and provide policy and technical recommendations. These recommendations might include redesigning bus routes and schedules, using scooter or bike share services to improve food access, repurposing existing city assets like shuttle vehicles and improving school lunch delivery programs.

The research team includes HV Jagadish, Bernard A. Galler Professor of Electrical Engineering and Computer Science and Director of the Michigan Institute for Data Science (MIDAS); Robert Hampshire, an associate professor at the U-M Transportation Research Institute, the Department of Industrial and Operations Engineering, and the Ford School of Public Policy; and Aditi Misra and Tayo Fabusuyi, both assistant research scientists in the U-M Transportation Research Institute (UMTRI).

Jagadish is contributing technical expertise to the project, including a data analysis technique that makes it possible to combine sets of geographical data that are organized in different ways—for example, comparing school lunch data that's organized by school district with transit data that's organized by ZIP code.

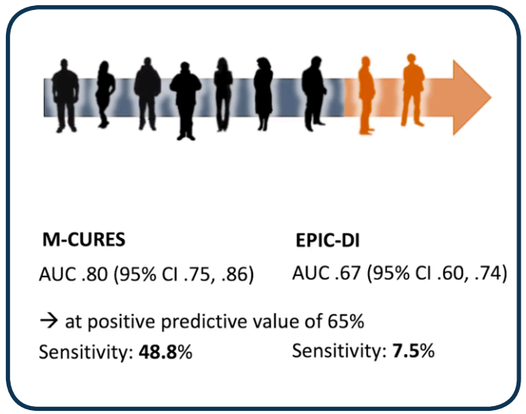
< READ MORE >

A COMPUTER MODEL THAT PREDICTS COVID'S NEXT MOVE

A computational model could hospitals and care providers a leg up on COVID-19 by predicting which patients are likely to quickly deteriorate upon admission. The model can help hospitals anticipate fast-changing patient needs while keeping care providers safe.

Called M-CURES and developed by an interdisciplinary team, the model uses a machine learning algorithm to crunch more than 200 health and demographic variables of individual COVID-19 patients. The researchers found that some of the most predictive variables include age, underlying health conditions and current medications. The model then outputs a numerical score, updated every four hours, that predicts the patient's likelihood of requiring ICU-level care. Preliminary validation of M-CURES showed it to be effective in predicting the progression of the disease.

The M-CURES model correctly identified nearly half of high-risk cases, significantly outperforming a proprietary deterioriation index. Likewise, it flagged far more low-risk patients than the proprietary model, or patients who could safely be transferred to a field hospital.



Widespread use of models like these could help healthcare providers better triage admitted patients.

"M-CURES could help the hospital get better answers to questions like who is likely to need ICU care and how many ICU beds it will need within a given time frame," said Jenna Wiens, associate professor and co-director of Precision Health at U-M. "It could also help the families of severely ill COVID patients by giving them more time to evaluate treatment options."

< READ MORE >

H>CKING RE>LITY

Microphones that “hear” light; microprocessors that “tell” us secrets; self-driving cars that “see” fake objects; sensors that “feel” the wrong temperature. Our devices are under attack in new, increasingly sophisticated ways. Security researchers at CSE are exploring the limits of hardware and finding new, sobering vulnerabilities in our computers and homes.

Where does “computer” end, and “real world” begin? This line, separated so firmly in our minds by apps and user interfaces, is finer than it appears.

Our computers aren’t bounded by programs, but by physics: the electrons flowing between a microprocessor and memory; device ports and wireless receivers; microphones, wires, speakers, screens, and other channels along which signals travel or cross from one medium into another. While programmers often categorize hardware as externalities that they don’t have to worry about, their programs are ultimately running on real-world machines with real-world imperfections.

With the widespread adoption of smart speakers, security cameras and vision systems, and embedded systems, the distinction gets even hairier — computers are suddenly a lot more perceptive than we’re used to, and the edges between computer and reality become viable channels for input. In tandem with this, microprocessors themselves have turned increasingly to clever tricks to achieve better performance without considering the observable side effects that start to manifest in the real world.

With this complexity in function come new avenues for hacking attacks. In a hypothetical closed computer, walled off from real-world physics, the weak points are code and human behavior. Trick someone >>

Microphones that hear light

A physics quirk discovered in the labs of Profs. Kevin Fu and Daniel Genkin led to a unique new hack: remotely controlling smart speakers with a laser. | Photo: Joseph Xu

An unexplained quirk of physics

PhD student Benjamin Cyr rigs up a laser and telescope in the Ann and Robert H. Lurie Bell Tower on North Campus as Prof. Daniel Genkin looks on. The two successfully injected commands into an Amazon smart home speaker from the bell tower and through a window 75 meters away. | Photo: Joseph Xu

into downloading harmful script or sneak it through a safe application unnoticed: hack accomplished.

Now, consider a hacker who can: open your garage door by aiming a laser through your window; find out your passwords, bank info, and more just by measuring the time it takes to access memory; slam a self-driving car's brakes with a laser; or turn the temperature up on an incubator with electromagnetic waves.

These scenarios and more have all been demonstrated by security researchers at the University of Michigan. The hacks are well beyond eventualities — they're possible right now. And as we rely ever more deeply on pervasive autonomous systems, internet of things devices, and elaborate methods to speed up microprocessors, we'll start to experience these hacks without some much-needed intervention.

"The security mechanisms we use were designed in the 60s," says assistant professor Daniel Genkin. "So now, we need to catch up."

Alexa and the laser pointer

In the realm of physical devices, smart home speakers like the Amazon Echo or Google Home present a classical example of that bridge between computer and reality: their only interface works by passively accepting audio from the environment. The shortcomings of this modality become apparent the moment a child decides to order something on Amazon without permission, needing only their voice and a parent's PIN.

But even here there's room for craftier work. Genkin, associate professor Kevin Fu, and research scientist Sara Rampazzi worked with a team led by PhD student Ben Cyr to inject unwanted commands into smart speakers and smartphones from 110 meters away. How did they do it? Not with sound, but with light.

In an as-yet unexplained quirk of physics, the team discovered that lasers could be encoded with messages that are interpreted by MEMS microphones as if they were sound. The team can record which command they want to inject, such as unlocking the front door, modulate the intensity of a laser with the audio signal, and then aim and fire.

As Genkin points out, this is a very dramatic example of a semantic gap.

"Microphones were designed to hear sound. Why can they hear light?"

The range of the attack is limited only by the intensity of an attacker's laser and their line of sight. In one compelling demonstration, the team used a laser from 75 meters away, at a 21° downward angle, and through a glass window to force a Google Home to open a garage door; they could make the device say what time it was from 110 meters away.

As for why the microphones are "hearing" the light, that remains an open question for future investigations.

"It's actually an unsolved problem in physics," Fu says. "There are plenty of hypotheses, but there's no conclusion yet. So this work goes well beyond computing."

For now, human-focused approaches to mitigation like extra layers of authentication between sensitive commands and execution may be the safest bet at the consumer level.

"The cyber-physical space is the worst of both worlds." Genkin says. "The devices are small, they're critical, and they're cheap. And you have a race to the bottom of who can make it cheaper."

Getting a sense for things

Direction, temperature, pressure, chemical composition — an array of sensors now exist to make specific measurements, enabling our machines to become more interactive and capable.

But tricking this range of functions doesn't actually require a range of hacks.

"Even though the sensor or attack modalities may be different," explains Connor Bolton, a PhD student in Fu's lab, "when you start looking at how each different component is exploited you end up seeing that the methods used are similar. When you start defining everything by the methods, you can see that a lot of attacks exploit the same kinds of principles."

The common weak point is generally the sensor's transducer, that component which converts physical signals into electrical. These are the components that take the physical world and all of its complexity most directly into account.

"Reality for sensors is only what they're programmed to see," Fu explains. "We're effectively creating optical illusions by using radio waves and sound waves that fool the computer."

Rampazzi tricked temperature sensors with electromagnetic waves of a certain frequency, for example. They've fooled other devices with sound, radio, and light waves. Sensors' vulnerabilities are two-fold: many are unable to distinguish correct from incorrect inputs, and unable to weed out interference from the environment. This combo of unreliable devices communicating in unreliable environments proves to be a nightmare for security.

"The moment you can no longer trust your environment," Fu says, "it becomes much harder to program a system. It's like programming Satan's computer at that point."

Unfortunately, with such a broad base of application areas, perceptual imperfections leave innumerable systems with diverse weaknesses. Take the temperature sensor for example — several different types of analog temperature sensors have been found to be susceptible to the same method of adversarial control, including devices used in critical path applications such as hospital incubators and chemical manufacturing plants.

"From meters away, or even in an adjacent room, an attacker could trick the internal control system of an infant incubator to heat up or cool down the cabin to unsafe temperatures," says Rampazzi.

The uniform characteristics of these attacks leave the field with both a trove of opportunities and unique challenges — almost everything is a viable hack to be fixed, but coming up with broad, systematic solutions is more difficult.

Bolton has focused his recent research on standardizing the scattered efforts in hardware security. The fairly new field draws expertise from assorted backgrounds, ranging from computer science to physics, leading to a

>>

number of redundant and mismatched solutions for overlapping problems. With a standardized model, the field can start to talk about their findings in a unified way that's more suitable to tackling fundamental issues.

"I hope people start using the same language and coming together," he continues, "and things become more about researching the methods versus a 'new attack on a new sensor' that really uses the same methods as others."

The CPUs that showed their hand

You don't need microphones or sensors to penetrate a computer's physical boundaries; chips, too, face hardware hacks that manipulate physical phenomena.

Thanks to a group of researchers including Genkin, standard practices in computer architecture have become mired in controversy. From the start of his career in 2013 to the present, he's helped to demonstrate a slew of so-called side-channel attacks. These rely on the non-traditional byproducts of code's execution, like a computer's power usage and the time it takes to access memory, to find pointers to off-limits data. In other words, they target places where computing crosses into the real world.

One such weak point his group has honed in on is the use of speculative execution. The result of a series of innovations that help a CPU process instructions faster, this practice involves predictively accessing data before it's needed by a program.

Unfortunately, this technique circumvents standard steps meant to ensure sensitive data is accessed properly, and that data fetched predictively can be leaked. Two of the group's most famous developments on this front are the Spectre and Meltdown exploits, which demonstrated how speculative execution could be abused to allow one application to steal data from another.

Because these attacks exploit weaknesses in the underlying hardware, defenses against them lie beyond the scope of traditional software safeguards. Solutions range from fundamentally changing how instructions are executed (which would mean a big hit for performance) to simply patching specific exploitation techniques through hardware and firmware as they become public. To this point, the latter approach has been preferred.

"The first attack variants were all using the processor's cache to leak information, so then you started seeing research papers proposing a redesign of the cache to make it impossible to leak information that way," says Ofir Weisse, a PhD alum who works on side-channel security. He says that this approach doesn't address the fundamental problem, since the cache is only one of many ways to transmit a secret.

"The fundamental problem is that attackers can speculatively access secrets, and then can try to transmit them in numerous ways."

In fact, Genkin's work in 2020 identified that specific patches adopted to address the issue were themselves prone to exploitation – it turns out that by simply "flushing" data out of the cache, it could be just as readily leaked elsewhere. It was a matter of identifying the initial pattern, says Genkin, and then finding the weak points in system after system.

Presently Genkin and other researchers at U-M are still burrowing deeper into the rabbithole, in search of both more vulnerabilities and viable defenses. But the question of where it ends can only be answered by the decisions of industry.

"I think at the end of the day there are going to be provably secure ways to mitigate all of these attacks," says Kevin Loughlin, a PhD student in this area, "but I think people will make a conscious security-performance tradeoff to leave themselves open to certain things that they view as not feasible."

Green doesn't always mean "go"

If what we've learned above is to be any indication, giving your car a pair of "eyes" is asking for trouble. A team at U-M were the first to show a very glaring reason why this is so, and it's as intuitive as pointing a laser at a light sensor.

Self-driving cars rely on a suite of sensors and cameras for their vision, and all of it is additionally wrapped in a protective machine learning system. This system both makes sense of the heterogeneous inputs the vision package receives, and makes it more difficult for incorrect or malicious signals to have an effect on the car's overall perception. To date, those malicious signals had been mainly targeted at the car's radar and cameras.

"Programming Satan's computer"
L-R: PhD student Conner Bolton, undergrad alum Fang Han, and research scientist Sara Rampazzi investigate sensors in Prof. Kevin Fu's lab. The electronic devices on which we depend are not as secure and trustworthy as we thought, according to the group. | Photo: Mike Hensel

A team led by PhD student Yulong Cao, professor Z. Morley Mao, Kevin Fu, and research scientist Sara Rampazzi broke new ground by designing an attack on vehicular LiDAR (Light Detection and Ranging) sensors. LiDAR calculates the distance to objects in its surroundings by emitting a light signal and measuring how long it takes to bounce off something and return to the sensor.

A LiDAR unit sends out tens of thousands of light signals per second, while the machine learning model uses the returned pulses to paint a fuller picture of the world around the vehicle.

"It is similar to how a bat uses echolocation to know where obstacles are at night," says Cao.

The problem the team discovered is that these pulses can be spoofed, meaning that fake pulses can be inserted into the genuine signal. To fool the sensor, all an attacker needs to do is shine their own light at it.

The team used this technique to dramatic effect. By spoofing a fake object just in front of the car, they were able to both keep it stopped at a green light and trick it into abruptly slamming its brakes. The fake object could be crafted to meet the expectations of the machine learning model, effectively fooling the whole system. The laser could be set up at an intersection or placed on a vehicle driving in front of the target.

"Research into new types of security problems in autonomous driving systems is just beginning," Cao says, "and we hope to uncover more possible problems before they can be exploited out on the road by bad actors."

>>

Seeing fake obstacles

Prof. Z. Morley Mao and her research team have demonstrated how to fool a self-driving car into slamming its brakes or sitting at a green light by aiming a laser at its LiDAR sensors. | Photos: (L) Graham Murdock, *Popular Science*; (R) Robert Coelius

Hacking — more than just code

With computers that talk, phones that hear, cars that see, and a world of devices that feel, we've entered bold new territory for the realm of cyber defense.

"You can't get away from physics," says Fu. "Programmers are taught to constantly ignore the details. This is the underbelly of computing — those abstractions lead to security problems."

And that only scratches the surface of the physical anomalies being discovered. While philosophers debate about the nature of human reality, computers are beginning to face an existential crisis of their own.

"It got to the point where it's all rotten," says Genkin. "It's rotten to the core, and to get the rot out you have to start digging." //

PROJECTS REFERENCED IN THIS ARTICLE

- "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving" < PAPER >
- "Light Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems" < PAPER >
- "Meltdown: Reading Kernel Memory from User Space" < PAPER >
- "SoK: A Minimalist Approach to Formalizing Analog Sensor Security" < PAPER >
- "Spectre Attacks: Exploiting Speculative Execution" < PAPER >
- "Trick or Heat? Manipulating Critical Temperature-Based Control Systems Using Rectification Aacks" < PAPER >

SECURING DRONES AND UAVS FROM CYBERATTACKS

Drones and UAVs are vulnerable to hackers that might try to take control of the craft or access data stored on-board. A group led by professor Westley Weimer built a suite of software to keep drones secure.

The suite is called Trusted and Resilient Mission Operations (TRMO). The U-M team is focused on integrating the different applications into a holistic system that can prevent and combat attacks in real time. < VIDEO >

CENSORED PLANET

Tracking internet censorship without on-the-ground participation

Internet censorship is a growing problem, and is not limited to closed societies. Censored Planet shows how even the freest countries are not safe from quietly encroaching interference.

The largest collection of public internet censorship data ever compiled shows that even citizens of the world's freest countries are not safe from internet censorship.

A University of Michigan team used Censored Planet, an automated censorship tracking system launched in 2018 by assistant professor of electrical engineering and computer science Roya Ensafi, to collect more than 21 billion measurements over 20 months in 221 countries. They presented the findings Nov. 10 at the 2020 ACM Conference on Computer and Communications Security.

"We hope that the continued publication of Censored Planet data will enable researchers to continuously monitor the deployment of network interference technologies, track policy changes in censoring nations, and better understand the targets of interference," Ensafi said. "While Censored Planet does not attribute censorship to a particular entity, it provides technical details on which network, where and when censorship occurs. We hope that the massive data we've collected can help political and legal scholars determine intent."

'News websites blocked in networks in Poland, social networking sites in Sudan'

Ensafi's team found that censorship is statistically increasing in 103 of the countries studied. Censored Planet found statistically significant, previously undetected censorship events between August of 2018 and April of 2020 in multiple countries, including unexpected places like Norway, Japan, Italy, India, Israel and Poland—countries which the paper notes are rated as some of the freest in the world by advocacy group Freedom House. Previously undetected events were also identified in Cameroon, Ecuador and Sudan.

>>

Internet censorship spreads in the world's democracies

Countries that are ranked "free" by advocacy group Freedom House and have increasing internet censorship according to a U-M study. The bars on this map show the rate of the increase of censorship in each country between August 2018 and April 2020. | Illustration: Steve Alvey

While the study observed an increase in blocking activity in these countries, most were driven by organizations or internet service providers filtering content. The study did not observe any nationwide censorship policies such as those in China. While the United States saw a smaller uptick in blocking activity, Ensafi points out that the groundwork for such blocking has been put in place in the United States.

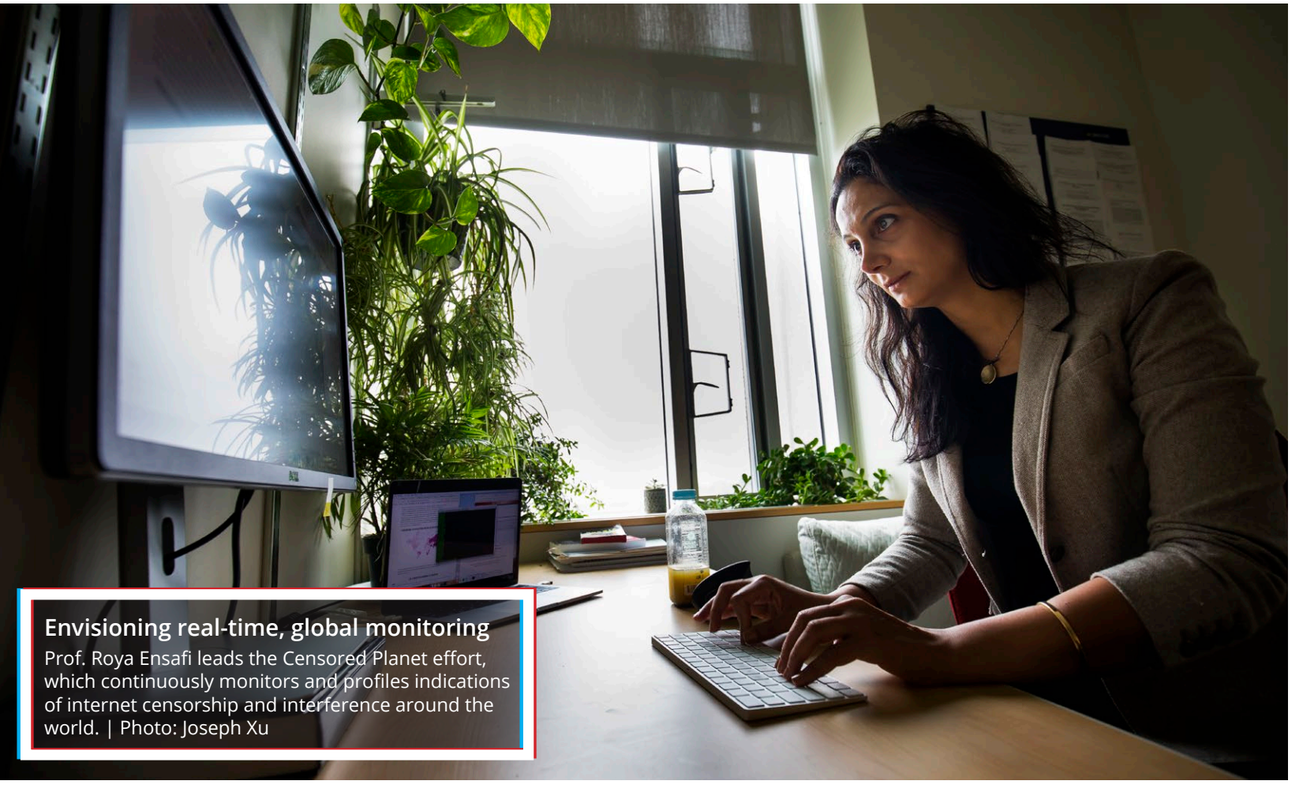
"When the United States repealed net neutrality, they created an environment in which it would be easy, from a technical standpoint, for internet service providers to interfere with or block internet traffic," Ensafi said. "The architecture for greater censorship is already in place and we should all be concerned about heading down a slippery slope."

It's already happening abroad, the study shows.

"What we see from our study is that no country is completely free," said Ram Sundara Raman, a PhD candidate in computer science and engineering at U-M and the first author on the paper. "Today many countries start with legislation that compels internet service providers to block something that's obviously bad like child sex abuse material or pirated content. But once that blocking infrastructure is in place, authorities can block any websites they choose, and it's usually a very opaque process. That's why censorship measurement is crucial, particularly continuous measurements that show trends over time."

Norway, for example—tied with Finland and Sweden as the world's freest country according to Freedom House—passed a series of laws requiring internet service providers to block some gambling and pornography content, beginning in early 2018. But in Norway, Censored Planet's measurements also identified network inconsistencies across a broader range of content.

Similar tactics show up in other countries, often in the wake of large political events, social unrest or new laws. While Censored Planet can detect increases in censorship and provide valuable technical details on the targets and ways of blocking, it cannot identify any direct connection to political events. It's also important to note that it's not always government-demanded network censorship that leads to websites being unreachable.



Envisioning real-time, global monitoring
Prof. Roya Ensafi leads the Censored Planet effort, which continuously monitors and profiles indications of internet censorship and interference around the world. | Photo: Joseph Xu

Censored Planet releases technical details for researchers, activists

The researchers say the findings show the effectiveness of Censored Planet's approach, which turns public internet servers across the globe into automated sentries that can monitor and report when access to websites is being blocked. Running continuously, it takes billions of automated measurements and then uses a series of tools and filters to analyze the data, removing noise and teasing out trends.

The paper also makes public technical details about the workings of Censored Planet that Sundara Raman says will make it easier for other researchers to draw insights from the project's data. It will also help activists make more informed decisions about where to focus their efforts.

"It's very important for people who work on circumvention to know exactly what's being censored on which network and what method is being used," Ensafi said. "That's data that Censored Planet can provide, and tech experts can use it to devise circumventions for censorship efforts."

Censored Planet's constant, automated monitoring is a departure from traditional approaches that rely on volunteers to collect data manually from inside the countries being monitored. Manual monitoring can be dangerous for volunteers, who may face reprisals from governments. The limited scope of these approaches also means that efforts are often focused on countries already known for censorship, enabling nations that are perceived as freer to fly under the radar. While censorship efforts generally start small, Sundara Raman says they could have big implications in a world that is increasingly dependent on the internet for essential communication needs.

"We imagine the internet as a global medium where anyone can access any resource, and it's supposed to make communication easier, especially across international borders," he said. "We find that if this upward trend of increasing censorship continues, that won't be true anymore. We fear this could lead to a future where every country has a completely different view of the internet."

The paper is titled "Censored Planet: An Internet-wide, Longitudinal Censorship Observatory." The research team also included former U-M computer science and engineering student Prerana Shenoy and Katharina Kohls, an assistant professor at Radboud University in Nijmegen, Netherlands. The research was supported in part by the U.S. National Science Foundation, Award CNS-1755841. //

PROJECTS REFERENCED IN THIS ARTICLE

- < Navigate the 3D world censorship map > seen on the previous page.
- "Censored Planet: An Internet-wide, Longitudinal Censorship Observatory" < PAPER >

>STATS

NEW FACULTY
Hired in 2020

Gred Bodwin
Asst. Professor
PhD Massachusetts Institute of Technology, 2018

Laura Burdick
Lecturer III
PhD University of Michigan, 2020

Mithun Chakraborty
Asst. Research Scientist
PhD Washington University, 2017

Xin He
Asst. Research Scientist
PhD University of Chinese Academy of Sciences, 2017

Yatin Manerkar
Asst. Professor
PhD Princeton University, 2020

Lu Wang
Asst. Professor
PhD Cornell University, 2015

Xu Wang
Asst. Professor
PhD Carnegie Mellon University, 2020

Paul Grubbs
Asst. Professor
PhD Cornell University, 2020

Anhong Guo
Asst. Professor
PhD Carnegie Mellon University, 2020

Max New
Asst. Professor
PhD Northeastern University, 2020

Thatchapol Saranurak
Asst. Professor
PhD KTH Institute of Technology, 2018

2567
UNDERGRADUATE MAJORS
FALL 2020

1167
UNDERGRADUATE DEGREES GRANTED
AY 2019-2020

371
GRADUATE STUDENTS
FALL 2020

150
MS+PhD DEGREES GRANTED
AY 2019-2020

90 FACULTY
62 TENURE/TT
24 TEACHING
4 RESEARCH

9 GRAD & UNDERGRAD PROGRAMS

$31M+
IN RESEARCH EXPENDITURES
FY2020

PROFESSORSHIPS

Mark Ackerman
George Herbert Mead Collegiate Professor of Human-Computer Interaction

Todd Austin
S. Jack Hu Collegiate Professor of Computer Science and Engineering

Satinder Singh Baveja
Toyota Professor of Artificial Intelligence

John P. Hayes
Claude E. Shannon Endowed Professor of Engineering Science

H.V. Jagadish
Bernard A. Galler Collegiate Professor of Electrical Engineering and Computer Science

John Laird
John L. Tishman Professor of Engineering

Rada Mihalcea
Janice M. Jenkins Collegiate Professor of Computer Science and Engineering

Emily Mower Provost
Toyota Faculty Scholar

Trevor Mudge
Bredt Family Professor of Engineering

Christopher Peikert
Patrick C. Fischer Development Professor of Theoretical Computer Science

Kang G. Shin
Kevin and Nancy O'Connor Professor of Computer Science

Michael Wellman
Lynn A. Conway Professor of Computer Science and Engineering

IEEE FELLOWS

Todd Austin
Valeria Bertacco
Peter M. Chen
Lynn Conway*
Edward Davidson*
Edmund Durfee
Kevin Fu
John P. Hayes
Benjamin Kuipers
Scott Mahlke
Pinaki Mazumder
John Meyer*
Trevor Mudge
Karem Sakallah
Kang G. Shin

ACM FELLOWS

Mark Ackerman
Peter M. Chen
Yuri Gurevich*
Mark Guzdial
John P. Hayes
H.V. Jagadish
John Laird
Rada Mihalcea
Trevor Mudge
Karem Sakallah
Kang G. Shin
Michael Wellman

AAAS FELLOWS

Lynn Conway*
Yuri Gurevich*
H.V. Jagadish
Chad Jenkins
Benjamin Kuipers
John Laird
Pinaki Mazumder

SLOAN FELLOWS

Todd Austin
Michael Cafarella
Reetuparna Das
Kevin Fu
J. Alex Halderman
Sugih Jamin
Chad Jenkins
Honglak Lee
Z. Morley Mao
Christopher Peikert
Jenna Wiens

U-M LEADERSHIP POSITIONS

Valeria Bertacco
Director, Applications Driving Architectures Center
Vice Provost for Engaged Learning, University of Michigan

Edmund Durfee
Co-Director, Technology Increasing Knowledge: Technology Optimizing Choice, U-M Rehabilitation Engineering Research Center

H.V. Jagadish
Director, Michigan Institute for Data Science

Chad Jenkins
Associate Director, Michigan Robotics Institute

Elliot Soloway
Co-Director, Center for Digital Curricula

Quentin Stout
Co-Director, Center for Space Environment Modeling

Gregory Wakefield
Director, ArtsEngine, College of Engineering

Jenna Wiens
Co-Director, University of Michigan Precision Health

*Indicates emeritus faculty

FACULTY 2020-2021

Mark S. Ackerman
George Herbert Mead Collegiate Professor of Human-Computer Interaction

Michael Adams
Assistant Research Scientist

Raed Almomani
Lecturer I

William Arthur
Lecturer IV

Todd Austin
S. Jack Hu Collegiate Professor of Computer Science and Engineering

Nikola Banovic
Assistant Professor

Satinder Singh Baveja
Toyota Professor of Artificial Intelligence

Jonathan Beaumont
Lecturer III

Valeria Bertacco
Arthur F. Thurnau Professor

Greg Bodwin
Assistant Professor

Mark Brehob
Kurt Metzger Collegiate Lecturer

Laura Burdick
Lecturer III

Michael J. Cafarella
Associate Professor

Joyce Chai
Professor

Mithun Chakraborty
Assistant Research Scientist

Peter M. Chen
Arthur F. Thurnau Professor

Mahdi Cheraghchi
Assistant Professor

David Chesney
Toby Teorey Collegiate Lecturer

Mosharaf Chowdhury
Assistant Professor

Marcus Darden
Lecturer IV

Reetuparna Das
Associate Professor

Andrew W. DeOrio
Lecturer IV

Kimberly Khalsa Diaz
Lecturer III

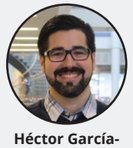
Ronald Dreslinski
Assistant Professor

Edmund H. Durfee
Professor

Roya Ensafi
Assistant Professor

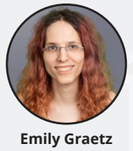
David Fouhey
Assistant Professor

Kevin Fu
Associate Professor



Héctor García-Ramírez
Lecturer II

Daniel Genkin
Assistant Professor



Emily Graetz
Lecturer II

Anhong Guo
Assistant Professor

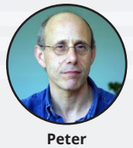
Mark Guzdial
Professor

J. Alex Halderman
Professor

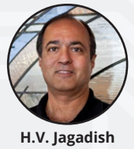
Nicole Hamilton
Lecturer III

John P. Hayes
Claude E. Shannon Professor of Engineering Science

Xin He
Assistant Research Scientist



Peter Honeyman
Lecturer I



H.V. Jagadish
Bernard A. Galler Collegiate Professor of Electrical Engineering and Computer Science

Sugih Jamin
Associate Professor

Odest Chadwicke Jenkins
Professor

Justin Johnson
Assistant Professor

James Juett
Lecturer III

Amir Kamil
Lecturer IV

Manos Kapritsos
Assistant Professor

Baris Kasikci
Assistant Professor

John Kloosterman
Lecturer III

Danai Koutra
Assistant Professor

Benjamin Kuipers
Professor

Sindhu Kutty
Lecturer III

John E. Laird
John L. Tishman Professor of Engineering

Walter Lasecki
Assistant Professor

Kevin Leach
Lecturer I

Euiwoong Lee
Assistant Professor

Honglak Lee
Associate Professor

Harsha Madhyastha
Associate Professor

Scott Mahlke
Professor

Z. Morley Mao
Professor

Jason Mars
Associate Professor

Pinaki Mazumder
Professor

Rada Mihalcea
Janice M. Jenkins Collegiate Professor of Computer Science and Engineering

Andrew Morgan
Lecturer II

Emily Mower Provost
Associate Professor; Toyota Faculty Scholar

Barzan Mozafari
Associate Professor

Trevor Mudge
Bredt Family Professor of Engineering

Satish Narayanasamy
Professor

Brian Noble
Professor

Edwin Olson
Professor

Cyrus Omar
Assistant Professor

David R. Paoletti
Lecturer IV

Christopher Peikert
Patrick C. Fischer Development Professor of Theoretical Computer Science

Veronica Perez-Rosas
Assistant Research Scientist

Seth Pettie
Professor

Atul Prakash
Professor; Associate Chair, Computer Science and Engineering

Sara Rampazzi
Research Investigator

Jeffrey S. Ringenberg
Lecturer IV

Karem A. Sakallah
Professor

Sofia Saleem
Lecturer I

Alanson Sample
Associate Professor

Thatchaphol Saranurak
Assistant Professor

Kang G. Shin
Kevin and Nancy O'Connor Professor of Computer Science

Elliot Soloway
Arthur F. Thurnau Professor

Quentin F. Stout
Professor

Lingjia Tang
Associate Professor

Ilya Volkovich
Lecturer III

Gregory H. Wakefield
Associate Professor

Lu Wang
Assistant Professor

Xinyu Wang
Assistant Professor

Xu Wang
Assistant Professor

Westley Weimer
Professor

Michael P. Wellman
Lynn A. Conway Professor of Computer Science and Engineering; Richard H. Orenstein Chair of Computer Science and Engineering

Thomas F. Wenisch
Professor

Jenna Wiens
Associate Professor

Austin Yarger
Lecturer I

COURTESY FACULTY

Steven Abney
Associate Professor
Linguistics

Eytan Adar
Associate Professor
School of Information

Ella Atkins
Professor
Aerospace Engineering

Robin Brewer
Assistant Professor
School of Information

Ceren Budak
Assistant Professor
School of Information

Kevyn Collins-Thompson
Associate Professor
School of Information

Paramveer Dhillon
Assistant Professor
School of Information

Tawanna Dillahunt
Associate Professor
School of Information

Barbara Ericson
Assistant Professor
School of Information

Ryan Eustice
Professor
Naval Architecture and Marine Engineering, Mechanical Engineering

Eric Gilbert
Associate Professor
John Derby Evans Professor of Information
School of Information

Jean-Baptiste Jeannin
Assistant Professor
Aerospace Engineering

Matthew Johnson-Roberson
Associate Professor
Naval Architecture and Marine Engineering

David Jurgens
Assistant Professor
School of Information

Vineet Kamat
Professor
Civil and Environmental Engineering

Jie Liu
Assistant Professor
Computational Medicine and Bioinformatics, Medical School

Qiaozhu Mei
Professor
School of Information

Viswanath Nagarajan
Assistant Professor
Industrial & Operations Engineering

Kayvan Najarian
Professor
Computational Medicine and Bioinformatics, Emergency Medicine, Medical School

Michael Nebeling
Assistant Professor
School of Information

Mark W Newman
Professor
School of Information

Long Nguyen
Associate Professor
Statistics

Stephen Oney
Assistant Professor
School of Information

Ravi Pendse
VP for Information Technology and CIO
Office of the President

Daniel Romero
Assistant Professor
School of Information, Complex Systems

Florian Schaub
Assistant Professor
School of Information

Ambuj Tewari
Assistant Professor
Statistics

Richmond Thomason
Professor
Linguistics, Philosophy

Joshua Welch
Assistant Professor
Computational Medicine and Bioinformatics, Medical School

Jieping Ye
Professor
Computational Medicine and Bioinformatics, Medical School

CSE.ENGIN.UMICH.EDU

Computer Science and Engineering at Michigan

@UMichCSE

Computer Science and Engineering at Michigan

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN