CASCADE

A year in Computer Science and Automation and the people who keep it moving forward.

Issue 03 AUGUST 2026

Contents

Makers of *CSA*

A long-form conversation with a lecturer who packed rooms and opened minds

C. Pandu Rangan

— *a conversation*

PROF. C. PANDU RANGAN
CS / IIT MADRAS

An IISc alumnus who arrived to study mathematics and left a computer scientist, C. Pandu Rangan went on to shape theoretical computer science at IIT Madras for four decades to the point where nearly every CS department in the world has someone who learned from him. In this conversation with Prof. Pandu Rangan, Prof. Bhavana Kanukurthi traces the whole arc: a small-town boyhood, a chance encounter in the Main Building, a career that began with four books and a punched-card "compiler".

INTERESTS

- graph algorithms
- automata & graph grammars
- computational complexity
- computational geometry
- randomized & approximation algorithms
- cryptography

A warm welcome to Cascade, Professor. You are an alumnus of IISc yourself, and your journey by your own admission, hard to trace — really began here. Can you take us back to the start?

What drew you to the field in the first place?

Let's begin at the beginning, as they say. I was quite taken with mathematics in my tenth standard. Around then I came up with a way to evaluate cube roots by hand — extending the pen-and-paper method we all use for square roots. It's a fairly trivial extension, but it was novel, and in a mid-sized town like Erode people were excited that this boy had found a new way of doing mathematics. We sent it as a proposal to the National Science Talent Scheme; they approved it, encouraged it, even offered a fellowship.

But mathematics was my passion. My father is a professor of English, a language expert, no sense of science or technology and on some probing he learned that IISc is the mother of all science and technology institutes in India. So he brought me here simply to show me the campus. We walked around and everything stumped me. Then he wandered into a room in the Main Building, and we bumped into, I now know who it was Narayan Mahishi. He received two total strangers who had walked in without appointment, a plainly dressed father and son from a small town, and very patiently explained what a PhD here could look like after a Master's in mathematics. That was the first person I met at IISc.

And how did mathematics turn into computer science?

I did my Master's in mathematics at Vivekananda College in Chennai, a top-class department. Then I wrote the all-India entrance examination. After a massive filtering, thirty-two of us were called for interviews that ran over two days, close to ninety minutes each. Four were finally selected, and I happened to be number one on the list. So they made me a unique offer: "Would you like to do computer science?" Professor E.V. Krishnamurthy (**EVK**) was then chairman of both the School of Automation and the Mathematics Department. A crystallographer by training who had turned into a computer scientist. I had never seen a computer, never heard of one. All I wanted was mathematics. I couldn't even interpret the implications of the offer. I said yes.

Then everyone told me I was making a stupid decision. This was 1977. After the Emergency, the new government had kicked out IBM, Coca-Cola, all foreign investment — George Fernandes sent IBM packing. We were left with three or four mainframes and no maintenance. The communist parties opposed computerization because it would cost jobs; no computers for banks, for anything. "Where is the future for a computer scientist? Take fluid dynamics, become a mathematics faculty somewhere. But don't go for computer science."

"When humans need something, they will find a way to get it done. Computers are going to rule the world tomorrow — and it is a need. Don't worry about the opportunity."

So I went back to EVK and told him what people were saying. His reply I still remember. "When humans need something, they will figure out a way to get it done. Computers are going to rule the world tomorrow, and it is a need. Whether the government supports it or not, whether there is opposition or not, computers will take over. It is five years down the line, by the time you graduate, you will have a phenomenal future. All this is temporary. Be part of the big picture." I was enthused. I said yes, I will join computer science.

You've said IISc made you a computer scientist almost from nothing.

One sentence I want on record: IISc is a place that can convert clay into a computer scientist. And I was clay. Absolute clay, nothing more. It converted me because of the ambience, the research, the freedom, the ability to nurture questioning without anyone monitoring or time-bounding you. Complete freedom. That conversion started then, thanks to EVK's advice at the turning point. Though EVK himself was a globetrotter over five years of PhD, I think I had about three hours of interaction with him in total. He would catch me in the corridor: "What happened? Did you submit something? Go ahead and do that." He gave approval and support, but couldn't interfere.

Two amusing things. We went for our first computer class — chemists, MBA people, all of us completely ignorant. Fortran, compiler translates to machine language, the machine executes everything flew five thousand feet above my head. Then they took us to the computer centre. Behind glass walls sat a mainframe like a deity in a garbhagriha, and we looked at it reverently. A man hurried in, pulled a bunch of punched cards, put them into the card reader and I thought, this man must be the compiler. I didn't even know a compiler was software! So you are talking to someone who once believed a person loading punched cards was the compiler. That was our level of ignorance.

"When I reached IISc, there were only four books on computer science in all of India."

The next day the librarian sent me to a row in the maths department for computer science books. I found nothing — rows of linear algebra, partial differential equations, ten rows of mathematics. I went back, a little annoyed, and finally he walked me there himself: the tenth row, the very last shelf. Four books. When I reached IISc, there were only four books on computer science in all of India. A manual, a xeroxed handwritten copy of David Gries on compilers, and the promise that Aho, Hopcroft and Ullman was arriving next week by voucher. With that meagre resource, and whatever seniors passed down, I had to begin research.

How did the thesis itself take shape?

EVK believed the best way to learn is to teach, so he had me lecture the School of Automation students on algorithms and automata theory, an MSc mathematics man, out of his comfort zone, reading a book and then teaching it. Vijaya Ramachandran, now at Urbana-Champaign, was a student in that class. Then EVK returned from CMU with a technical report on graph replacement systems. Rather than string replacement, they wanted substructure replacement to work at a higher level of abstraction, extending the grammars used for compilers into graph grammars. That report became the trigger for my thesis: graph grammars, the classes of graphs they generate, membership questions all the automata-theoretic questions, but on graphs.

"He created the Tamil fonts himself and set my name in Volume 2. Seeing it was a thrilling experience — you're just a student here."

Meanwhile, at TIFR, EVK pointed me to Knuth's books the only ones available. Knuth marks problems by difficulty; anything above 48, if you solve it, becomes a paper. I took one ranked 48, worked on it, and solved it. I wrote to Knuth, he has no email even now, only hard mail and he replied that it was "very cute" and forwarded it to the new Journal of Algorithms that he, Wilf and others were starting. It appeared in 1982. Then he wrote asking to set my name in my own language. I wrote "Chandrasekaran Pandu Rangan" in Tamil on a sheet of paper; he created the Tamil fonts himself and put it in Volume 2, referring to my solution. Even now, my name in Tamil is in Knuth's book.

And then the country's stance on computing began to change.

Around 1981, another political storm — then Indira Gandhi returned, and after her, Rajiv Gandhi. A pilot, tech-savvy, keen on the computer revolution. Microprocessors had just hit the world; the market was flooded with "IBM compatible" machines from Taiwan, everyone starting a PC company and chasing government orders. Floppy disks were shrinking into compact disks. He asked why India had no undergraduate computer science programme — there were only ad-hoc arrangements run by computer centres and cobbled-together electrical and maths faculty. So he simply said: start one.

Narasimha Rao, then the minister, called the IITs and gave each a seed grant of fifteen lakhs: "Start B.Tech in computer science and engineering. Do whatever it needs; if more is required, we'll figure it out." So every IIT advertised for faculty with a PhD in computer science. But where was such a PhD in 1981? A couple of us from the School of Automation would keep bumping into each other — even on the interview boards. I was called by IIT Bombay, Delhi, Kanpur and Madras, and got offers from all four. I had the privilege of choosing an IIT, rather than an IIT choosing me — purely a matter of timing, not because I was some Nobel laureate. I chose Madras: I am the only son, my parents are orthodox and rooted in Tamil culture, and I had to take care of them.

What was the interview like at IIT Madras?

Professor Indiresan was the director. He asked me to sign a sheet of paper, looked at it and said, "Your signature is very artistic. If I offer you a job, will you join?" I said Chennai is new to me — if you give me housing, I'll join. "If I give you housing, will you join?" Yes. "Then I'll give you housing. Your interview is over." Two minutes, entirely a negotiation over housing. And he added, "I'm going to treat you in a very privileged manner."

"Everyone was given a basic pay of 700 rupees. I was given 740. That 40 rupees made me feel like a Maharaja."

I didn't understand until the offer came. They had recruited forty-eight lecturers, all on a basic pay of 700 rupees. The director had given me 740, one special increment because computer science had practically no PhDs available. Forty rupees! But in a government system, seniority is decided by your joining date and your basic pay, so when the first quarters fell vacant, I was called to occupy them ahead of everyone. That forty rupees made an enormous difference and more than the money, the signal made me feel like a Maharaja.

Years later, on selection and promotion committees, I always fought for that same signal. When you promote someone, don't fix them at the lowest basic — give a head start, an extra increment or two. It says: this isn't a routine promotion; here is our recognition of what you have done. It's not monetarily large, but the message is amazing. Directors would tell me afterwards that candidates received it wonderfully. Because I remembered how one increment excited me when I was young.

Over the years, your work at IIT Madras has influenced generations of students, to the point where nearly every computer science department in the world has someone who learned from you. How did that journey unfold?

Walk us through the IITM journey

I joined on June 3rd, 1982. That evening I went for a walk on the beach and was pushed by the police into a concert at the University Centenary auditorium — compulsory audience. It was Chief Minister Karunanidhi's birthday, the concert in his praise. We simply walked in and sat, and he came and sat two seats away, said "Vanakkam," asked what I do. "I joined IIT today." "Appadiya? Nalla irunga." I still remember it. I began teaching on September 15th — which happened to be Annadurai's birthday. A strange coincidence of Tamil Nadu Chief Ministers bracketing my beginnings. That "nalla irunga" has kept me going all these years.

But there was no research resource. IIT Madras was a coursework-intensive institute; research culture wasn't there, and the CS department wasn't even formed yet I joined the computer centre. There's a famous slogan about our PhD pipeline: advertise, and eight apply; call for interview, four appear; then two; you select one, and finally he doesn't join. Eight, four, two, one, zero. Coming from IISc, surrounded by PhD scholars, I arrived to a department with essentially none. So if I was to do research at all, the only resource was the undergraduates.

So you built a framework around undergraduates.

The students told me plainly: the mathematics here is pathetic after JEE, too elementary, no challenge. So I said theoretical computer science has enough mathematics and enough challenge; let's do that. But their window is small: five or six courses, two labs, tiny project slots. High potential, small interval. So I needed compact problems that gave quick success. I built a matrix restricted classes of graphs down one axis, NP-complete problems across the other and started ticking boxes. Interval graphs, circular-arc, comparability, permutation graphs. "This is polynomial on interval graphs; can you push it to a superclass?" Suddenly it might turn NP-complete. It felt tangible to them, and they loved it.

"A journal paper from a sixth-semester undergraduate — at a conference, a colleague simply couldn't believe it wasn't a professor."

The results came. Kiran Kumar, from the first batch, found a new linear-space algorithm for the longest common subsequence problem; it went into Acta Informatica. At a conference, a neighbour asked where "Professor Kiran Kumar" was now — he couldn't believe it was a sixth-semester undergraduate. Atul Saini — "Tulli" — designed a VLSI algorithm accepted at an IEEE conference in Singapore, and later wrote the citation-classic book on the C++ Standard Template Library. Ramalingam's interval-graph numbering became a paradigm; taught abroad as the "Ramalingam–Pandu Rangan numbering," it seeded a whole family of algorithms. Batch after batch.

Each new batch would say, "Sir, don't give the problem my seniors left open — if Ramalingam couldn't solve it, I can't." So I had to keep reinventing myself: computational geometry, randomized algorithms, approximation algorithms, always the recent and compact. I became a rolling stone. People say a rolling stone gathers no moss — but I wasn't worried about mass. I was gathering momentum.

When we last spoke, you described using teaching as a tool and building a framework of compact problems for undergraduates and how you kept having to expand your own horizons. Let's pick up there.

> *How did that framework of problems actually work — and did it carry over to cryptography?*

Undergraduates have very little time for intense research, so the problems must be compact — low-hanging fruit they can pluck without going too deep. So I built a big map of research problems and plugged different students into different coordinates of it. The simplest illustration: graph problems are generally NP-complete, but on a restricted class — planar graphs, interval graphs — the same problem can become polynomial. So I kept a matrix: which NP-complete problem becomes polynomial on which special class of graphs. If a subclass is already investigated, that gives them a handle; they move to a superclass, catch the intuition, do a little more, and they have a new result.

I did exactly the same in cryptography. There we have various models of security and various paradigms; in secure distributed computing and multiparty computation there is a "holy grail" problem, and around it a huge matrix of variations and specialities. Eventually we could even answer that holy grail problem — a celebrated JACM paper by Arpita and Ashish. A classical result.

> *And the shift from undergraduate research to a full research group?*

IIT Madras was not known as a destination for research scholars. The joke about our admissions: you advertise and eight apply; four appear for the interview; two are selected; one offer goes out — and he doesn't join. Eight, four, two, one, zero. For much of my early career we had no scholars, so any research had to be done through the undergraduates. Things changed slowly. Mathematically inclined people in the IT industry grew unhappy and returned for higher studies — Chaya Ganesh was one outstanding such student.

So I extended the framework to the open-problem level. The research scholars provided the deep foundation by staying much longer, and when the undergrads came in they took a small part of a big ongoing effort and became co-authors — catalysts for the deep work. Kannan Srinathan's thesis won IBM's global best-thesis award. Kamakoti, now the director of IIT Madras, was my 1990 PhD student; Sudarshan, now deputy director at IIT Delhi, was another of those early scholars.

"For the first fifteen to twenty years, it was the undergraduates who did the major part of the research."

> *So this became less about your own research and more about building an ecosystem.*

Very much. Since scholars weren't there, I had to make teaching the tool to build research, and I did two things. First, a coherent stream of courses discrete mathematics, data structures and algorithms, advanced algorithms, then cryptography across the second to fifth semesters, with advanced assignments. Solve the starred problems and you get a straight A; find a novel solution, even at the same complexity, and you get the top grade. IIT students are bright and competitive, so these challenges gravitated them toward research. Once they had a taste of solving advanced problems, I would hand them a slightly open one their solution might be an n-to-the-sixth polynomial, itself a breakthrough, and then I'd ask them to improve it.

Second, I went on the road. I gave seminars where strong undergraduate teaching was happening - NIT Trichy and other top schools and invited the top rank-holders from Chennai's twenty-odd colleges for a Saturday of talks and lunch: what research is, what you can do at an IIT Madras. Applications for theory and cryptography rose immediately. As head of department I also asked leading industries IBM among them for special fellowships well above the standard stipend, which drew excellent scholars. When I took the headship we had about seven PhD students; when I finished, seventy-two and MS students went from a handful to eighty. The department had 150 research scholars.

"They call it the lab with the maximum IQ per square foot in India."

We call it the TCS lab, people confuse it with the company, but it is the Theoretical Computer Science lab. It is buzzing with gold medalists and award winners: STOC and FOCS best-paper winners, the best undergraduate publication at those venues, even a student hand-picked to work directly with Bill Gates. Its graduates are faculty at Princeton, MIT, Berkeley — Raghavendra, known as the "Theory Oracle," is at Berkeley. My lab has produced twenty-seven President of India Gold Medalists; colleagues who guided two or three in an entire career find that number astonishing. Motivate students in their coursework, and they knock on your door on their own.

You always seem to think of yourself as a teacher first. Where do you draw your inspiration — teaching, or research? *So which comes first for you?*

Always teaching. But my research instinct itself was sharpened by problem-solving. As a PhD student in the mathematics department at IISc I was lucky to have colleagues like Praneshachar; every day the American Mathematical Monthly posed challenging problems, and we Venkatachala, and Lobo, who was almost a chess grandmaster would pick the combinatorial ones and solve them on the board, quite apart from our research. That sharpened how I think. You learn to think by wrestling a variety of hard problems, not only by reading the relevant papers.

At IIT Madras teaching is the primary activity and research an allied one — in a senate meeting eighty per cent of the time went to the B.Tech programme, and research got ten minutes. So I had to use teaching to promote research: give students the input, let them immerse themselves in a topic, hand them challenging problems, make them think intensely. What luck had given me through Praneshachar, I could now build deliberately into a course.

Is it a myth that teaching and research trade off?

It is a myth. In my experience, absolutely no compromise on either front. Teaching in a class like IIT — where students bring their own solutions and questions — is continuous learning that feeds research.

"I don't distinguish. As long as you create something, we will find a good venue for it — so be a proud author of a good paper."

You had a memorable way of describing that — the blade paradigm.

The Malhotra company made shaving blades on a single machine, priced from ten paise to over a rupee. A visitor was amazed there was just one machine for all the brands. Malhotra explained: the machine makes blades; we test a sample, and if it is a little blunt we call it Panama and sell it for ten paise, if it is ultra-sharp we call it Prince and price it at a rupee. I do the same with results. A routine one goes to Information Processing Letters; an excellent one to SODA or JACM. I never set a threshold, everybody feels they have produced a good result and published a paper. Venkatesan and I published only IPL papers; he went on to win the ACM best-thesis award at MIT and now directs the Simons Institute. Like Rama and the squirrel, you honour the small contribution too, rather than dismissing it. That is what enables the research instinct.

You energised more than your own group. For a generation, people across the country first heard the word "cryptography" because of what you were doing at IIT.

> *I was in the audience when you once taught the BGW protocol with no notes — the equations simply flowed onto the board. How?*

I always use the board, because then the delivery rate matches the grasp rate. If I flip slides, by the fifth slide the first is already lost. Slides suit a knowledgeable audience you are summarising for — the novel point in telegraphic mode. But when students meet a definition for the first time and study an example, writing everything on the board is best. And when you write, the idea gets into your bloodstream; it becomes a part of you. I read the source carefully before the lecture, then reconstruct it from memory on the board, hyperlinking ideas, building the proof of correctness live.

The first couple of offerings there is some fumbling — "wait, I should explain this before that." But by the third or fourth it is a natural flow. From head to hand, hand to chalk, chalk to board. I interact — write, turn to the class, check whether they have understood, invent an example on the spot when someone asks. Do that two or three times and the content is completely internalised.

"From head to hand, hand to chalk, chalk to board. Farm to table, translated to teaching."

> *And making mistakes in front of the class?*

Very important. When I am wrong, I apologise in class, go back, work it out, and clear the mistake at the start of the next class before continuing. I did it as a natural habit; only later did I learn that many teachers won't — ego, or they simply don't bother. Students appreciate it enormously.

And there is a surprising effect. In IITs students fight bitterly over half a mark. But because they have seen me accept my own mistakes, the moment I point out theirs, they accept it and stop arguing. Colleagues tell me this never happens to them. When the teacher owns his errors, the students quietly adopt the habit too — it held right up to the last batch I taught Discrete Mathematics here at IISc.

In this post-pandemic world, where writing is disappearing, what is your message for students?

How should students learn in the AI era?

First: stay away from the machines and read the book. Any problem you type into ChatGPT gives you a solution you can reproduce without thinking — and that takes you nowhere. You get the marks, not the competence. In the exam hall there is no ChatGPT; you have to think. Mathematics is creative, abstract thinking, and if you lean on a tool you miss the very opportunity to build it. To learn to paint, seeing a thousand AI images will not make your hand work. Creativity has to go through the struggle — you break your head and you think.

"It's not cheating the teacher. It's cheating themselves — of the one chance they have to struggle and learn."

So I run a mock session. Computers away. I write a problem on the board — is the minimum-cost edge always part of a spanning tree? — and say: you have three minutes, you and your pen. The whole class falls into intense silence, and I tell them, this silence is what I want; you, your paper, and the problem. Someone will react, and that shows everyone can — a JEE class is homogeneous, there is no real difference between rank five and rank five thousand. Once, teaching a class of seven hundred, I ended every lecture with a puzzle: take 4444 to the power 4444, sum its digits to get A, sum A's digits for B, then C — find C, with nothing but common sense. You should see the fire it kindles.

Post-COVID I do three things: insist on handwritten submissions, because writing internalises; randomly call on students to explain their steps, so copying without understanding shows immediately; and keep telling them that being on their own is best. Leaning on a source was always there — a friend who knew the answer, an oracle — ChatGPT has only made it democratic. I cannot police it. But if a student understands the logic — his own, a friend's, or ChatGPT's — I am happy to give the marks. Beyond that it is up to them. At IISc my post-dinner problem-solving sessions run past midnight, with scholars from mechanical, chemistry, biology — the participation is amazing.

At a national level — how should computer science be taught, at school and undergraduate level, especially in the age of AI?

What should India do to strengthen computer science education?

Start by asking: what is computer science? — independent of the job market. One high-level answer is the design and use of computers. There is the system side — architecture, networks, parallel and distributed systems — and the use of computers to solve problems. So a computer is a problem-solving tool, and the thinking we must build is computational: explicit, computable steps. Rolle's theorem tells you a theta exists, but not how to find it. Computational thinking is constructive — I want the value, because I want to build. And constructive is not enough: the determinant has a constructive definition that takes exponential time, so we need algorithmic thinking too, tied to the system's resources — cache, processors — because that is how real apps are built.

The ACM 2023 curriculum articulates this well, with eight streams of essential knowledge; at undergraduate level we give a basic introduction to all of them, and specialisation comes later. At school level, even more basic — computational thinking. School students only know formula evaluation: the volume of a sphere is four-thirds pi r cubed, but ask why four-thirds, or give a radius that will not cancel neatly, and they are lost. Teach them logic and step-by-step solution-building — sort the array, don't merely sum it — not the syntax of a for-loop. Logical thinking to build a solution is what they can grasp at that age.

"Don't let the job market blur the goal. Whether it gets you a job today is a different question from what computer science is."

And we must not drop programming because "AI does our job." To use an AI tool, fine — you needn't know complexity or Turing machines. But if you are learning computer science, you must learn automata theory and the models of computation; the impossibility results in distributed computing are exactly what blockchain is built on. Don't confuse using AI as a tool with learning AI as a subject.

One more thing I do: at the start of any course I ask, what are the five greatest ideas in this subject? The FFT is a great idea, reaching back to Gauss; seen that way, students look at it differently. No textbook says depth-first search won Hopcroft and Tarjan a Turing Award — when I show what it unlocks and ask, now do you see why?, I have had the class break into applause. Shannon's perfect security, the one-way function with a trapdoor, multiparty computation — the millionaires' problem, or five schools that must find the total number of drug cases without revealing their own — history and real application make the great ideas come alive.

You radiate positivity. Where does it come from?

My family — above all my father, S.R. Chandrasekharan, known as SRC. He was a professor of English at a rural college, Chikkaiah Naicker College in Erode, teaching English to students who spoke only Tamil. He animated everything — teaching "The Lady of Shalott," you could see the web, the lady, the frustration come alive. He would split the class in two and set them hunting ten difficult words each, with meanings and sentences, idioms and phrases; the last fifteen minutes of every class were for general knowledge and English, and he made a feared, boring subject exciting. He insisted they read the English newspaper — go to the library, write down what you read, read it out even in broken English.

"My extra problem-solving in class is my father's extra language-learning. I simply customised it in my own way."

Once the college erupted in a strike — a mob breaking furniture, and no way to call the police from three miles out. The principal sent for my father. He simply said, "Students, please be silent; let four or five of you come, and we will understand your problem." You should have seen how the mob went quiet. I was a student in that same college and watched it happen. The positivity comes to me from him — and from my mother, who anchored me in values.

I am the only son; I never knew shortage, though we lived humbly — I grew up on a cycle, never a scooter, but any book I wanted, he bought. He trained me in public speaking from a young age — every speech, poetry and essay competition, in Tamil. My first public speech was in the fourth standard, and I won first prize. I knew the Vishnu Sahasranamam by heart at six; the Jesuit Father at St. Joseph's in Trichy had me chant it at a Sunday Mass as an example, then lifted me in his arms and gave me a Bhagavad Gita. I did mathematics with chalk on the floor of the house — big multiplications, square roots, the whole floor covered in numbers. Seeing that, my father brought me to IISc in my tenth standard and introduced me to Narayana Mahishi, then the registrar, to ask about a future in mathematics. The confidence and the positivity are completely his.

— End of conversation

A LONG-FORM CONVERSATION, RECORDED ACROSS SEVERAL SITTINGS · EDITED FOR LENGTH AND CLARITY

New *Faces*

A lighthearted hello to the newest member of the department.

Sanchari Sen

— a warm welcome

DR. SANCHARI SEN
ASSISTANT PROFESSOR · CSA

The department welcomes Sanchari Sen, who joins CSA as an Assistant Professor. Her research builds efficient, scalable and sustainable AI systems through techniques that reach clear across the hardware–software stack — spanning accelerator, compiler and algorithm design — with the goal of enabling high-performance and energy-efficient AI.

INTERESTS

- efficient systems for ML & AI
- hardware–software co-design
- architecture for AI accelerators
- compiler design for accelerators
- performance & energy modelling

Rather than a formal bio, we asked Sanchari a handful of unserious questions — the kind that actually tell you something about a person.

How has your experience been so far — and have you found a favourite Bangalore restaurant yet?

Everything has been great so far, except for the traffic — which I guess is no surprise to anyone. The Bangalore food scene is even better than I expected, and we try to explore somewhere new every time we go out. My current favourite is Khmer Kitchen in J.P. Nagar, which serves amazing South-East Asian dishes.

Fuel of choice — coffee, tea, or something else entirely?

Primarily a coffee person, though I don't mind tea occasionally. I was mostly used to espresso or drip coffee before, and only got introduced to filter coffee after moving here. I must admit it's my go-to form of coffee right now.

Pop-culture check — the best book or show you've enjoyed recently?

I'm watching Spider-Noir on Amazon Prime and really enjoying it. I'm not a binge-watching kind of person, but it's hard to stop between episodes. I'd never have imagined Nicolas Cage as Spider-Man, but he has really embraced the role.

Travel bucket list — the number-one destination you're hoping to visit next?

I haven't managed a road trip anywhere around Bangalore yet, and I'd love to go somewhere in the Western Ghats next. I've heard they're beautifully lush and green during the monsoon — though with the current dry spell, I'm worried I won't catch them in their full glory.

Instant mastery — if you could instantly learn any new skill, what would it be?

I'd love to learn a stringed instrument and play along with songs. I've tried in the past but clearly lack the patience to sit through the initial cacophony of weirdly sounding notes.

Department *Activities*

Workshops, programmes and partnerships across the department's year.

The year in *events*.

The department's flagship gathering this year — a look at who came together and what was discussed.

WORKSHOP · 2026

Around 200 participants — including 75 external attendees — from IITs, IISERs, IIIT Bangalore, ISI Kolkata, NIT Jamshedpur and IIEST Shibpur, alongside professionals from Walmart Global Tech, TCS Research, Microsoft Research, DRDO, Amazon, PayPal, BMW TechWorks, Texas Instruments, Verizon and Dell. Talks spanned MDPs and Bellman operators through to risk-aware learning, world-model agents and large-scale e-commerce systems.

csa.iisc.ac.in/rl-workshop-2026

Research & *Awards*

Papers, prizes, conferences — the year's quieter, longer victories.

Honours & *papers*.

Highlights from the department's year in research — selected awards, accepted papers, and the conferences that took our students and faculty around the world.

R · 01

Vishali S

PHD STUDENT · MATHEMATICS × CIVIL

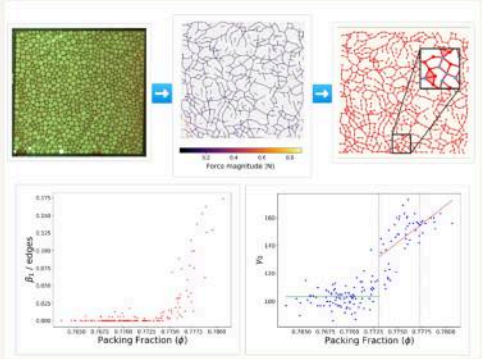
Advisor: Prof. Vijay Natarajan & Prof. Tejas G Murthy

Best Poster Award — Powders & Grains 2025

Topological Signatures of Jamming in Granular Force Networks via Persistent Homology

Vishali S, Abrar Naseer, Vijay Natarajan, Tejas G Murthy

POWDERS & GRAINS 2025 · GOA, INDIA



From photoelastic force networks to persistent-homology signatures across packing fractions — the jamming transition made visible.

R · 02

Sumit Kumar Mandal

FACULTY

INAE Young Engineer Award 2025

Recognised by the Indian National Academy of Engineering for sustained contributions in energy-efficient computing systems and machine-learning hardware — including in-memory and 2.5D ML accelerators with optimised on-package interconnects.

Citation, INAE Young Engineer Award 2025

INAE ANNUAL CONVENTION 2025 · INDIAN NATIONAL ACADEMY OF ENGINEERING

Dr. Sumit Kumar Mandal receiving the INAE Young Engineer Award 2025 at the INAE Annual Convention.

R · 03

Debajyoti Kar

PHD STUDENT

Advisor: Prof. Arindam Khan

Microsoft Research India PhD Award 2025

Improved Approximation Algorithms for Three-Dimensional Knapsack — presented at SoCG 2025, Kanazawa. A companion paper on Three-Dimensional Bin Packing was presented at ICALP 2025, Aarhus University.

with Klaus Jansen, Arindam Khan, K.V.N. Sreenivas, Malte Tutas (SoCG '25); Arindam Khan & Malin Rau (ICALP '25)

MSR INDIA · SOCG '25 · ICALP '25

R · 04

Aditya Subramanian

PHD STUDENT

Advisor: Prof. Arindam Khan

Accepted at SODA 2026

Online Connectivity Augmentation

Mohit Garg, Aditya Subramanian

ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS 2026

FIND US · WRITE TO US

We'd love to *hear* from you.

EMAIL · CHAIRPERSON

chair.csa@iisc.ac.in

WEB

csa.iisc.ac.in

ADDRESS

CSA, IISc
Bengaluru — 560 012

EDITOR

Paul K. Lhungdim

CONTRIBUTORS

The faculty & students of CSA

PRODUCTION

CSA Office