IEEE Conference on Artificial Intelligence
Santa Clara, California, June 5–6, 2023

Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso
El Paso, Texas 79968, USA, vladik@utep.edu

1 General information about the conference

Who organized it. In June 2023, IEEE organized its inaugural Annual IEEE Conference on Artificial Intelligence IEEE CAI 2023. Specifically, this conference was jointly organized by the IEEE Computational Intelligence Society (CIS, lead organizer), Computer Society, Systems, Man, and Cybernetics Society, and Signal Processing Society.

What were the main objectives. The main objectives of this conference were to:

- promote discussions between industry and academia,
- help AI-related policy making,
- inspire startup growth in AI-related domains, and

Figure 1: Gary Fogel and Piero Bonissone, Conference Chairs

- help connect students with future employers.

For these purposes, the conference enabled participants to:

- access leaders from the biggest influencers and innovators in AI;
- learn technologies and solutions for practical and immediate applications;
- gain lessons learned and leading practices for the latest AI technologies;
- discover exhibits, posters, and demos by leading companies and emerging start-ups;
- hear plenary keynotes each day, plus two invited speakers and panels for each AI segment;
- network with job seekers/recruiters, VCs, and start-ups to learn and connect.

General statistics. Out of 292 submissions, 154 were accepted and presented, which makes the acceptance rate 53%.

Selected papers were presented orally, the majority as posters.

Overall, 375 people registered for this conference, of which only 38% were authors. This fits well with one of the main objectives of this conference – to help people learn more about AI.

Of all the participants, 31% were from outside the US, representing about 20 countries.

To attract student participants, the conference offered 17 travel grants to graduate students; 9 of these grants went to US folks, 3 to students from Canada, and 1 each to students from Chile, China, Germany, India, and Vietnam. At the end, only 21% of the participants were student. The organizers plan to try better for next year's conference.

2 What was presented

Keynote talks, regular talks, etc. The conference had four keynote talks:

- "Multi-million core clusters co-designed for the future of generative AI" by Sean Lie;
- "AI and education" by Peter Norvig;
- "Computing in the foundation model era" by Kunle Olukotun; and
- "Deep Learning in genomics and proteomics" by Serafim Batzoglou.

Regular talks were divided into six groups called "verticals":

- AI in Healthcare/Life Sciences,
- AI in Transportation/Aerospace – with an emphasis on close-to-autonomous devices in all the environments: on the road, in the air, underwater, etc.,
- AI for Energy,
- Industrial AI – prognostic and health management, digital twins, etc.,
- AI for Earth Systems Decision Support, and
- Social Implications of AI.

Each vertical, in addition to regular talks, also contained invited talks and panels.

The conference also included IEEE Entrepreneurship Workshop and IEEE CIS Women in Computational Intelligence Reception.

What we will describe. In a short report, it is impossible to describe the contents of all the 154 accepted papers. What we will describe is mostly ideas mentioned in the invited talks. These ideas will be organized as follows: what are current successes of AI, what are remaining problems, and what ideas can we use to solve these problems.

3 What are the current successes of AI

These successes are well known. Current AI-based systems are helpful in many area of human activity; for example:

- *in agriculture:* they help detect the presence of bugs in the winery field;

- *in art:* they generate beautiful images in reply to our tasks;
- *in biology:* they predict where the DNA will be spliced and how the resulting proteins will fold;
- *in computing:* they help write code and – what is even more complicated – they can detect and fix bugs in the existing code;
- *in engineering:* they provide almost autonomous driving and flying, they help detect defects in engineering systems ranging from pavements to buildings and turbines;
- *in medicine:* they diagnose many diseases better than most human doctors, and they help design new drugs to cure these diseases; etc.

It is difficult to find an area where AI has not yet led to drastic changes.

4 What are the remaining problems

These are successes, but there are also many challenges. Let us start listing them.

4.1 Short-term problems related to machine learning

Let us first list problems related to the current approach – machine learning. Let us start with short-term problems.

Need for explainability and trustworthiness. Modern machine learning-based system are very good, but they are not perfect. Sometimes, AI systems give wrong advice. A problem is that most modern machine learning systems are, in effect, black boxes. They do not provide any explanations. We need to make AI results explainable – or at least more explainable. This will help increase our trust in solutions provided by AI systems, this will help to make sure that these systems treat everyone fairly and equally.

We also need to counter unjustified fears – such as ChatGPT taking over the world – and we also need to deal with justified fears, such as job displacement and other possible negative societal implications.

Need for privacy. The more data we have, the better decisions we can make – whether it is about curing diseases or teaching students. On the other hand, we do not want people to have access to details about our illnesses or our grades. In the past, a simple anonymization of the records was enough to guarantee such privacy. However, nowadays, Large Language Models like ChatGPT are so good that they can often identify a person based on partial anonymized information. How can we restore privacy?

Need for faster computations. The spectacular successes of modern AI became possible largely because of the 40,000 times increase in AI-related compute over the last 5 years. This shows that:

- while Moore's law – that predicted doubling of computational speed every 2 year or so – is no longer valid for personal computers,
- from the machine learning prospective, Moore's law is very much alive.

While AI has been very successful, there are still many problems for which the current computation speed is not sufficient. We need to continue drastically increasing computation speed, probably by the same amount in the next 5 years. However, the currently used speed-up techniques are not scalable, we are approaching their limits. For example:

- in 5 years, we decreased the gate size from 16 nm to 7 nm; chips with 2 nm gate are currently being developed, but this is probably the limit – and this will not provide us with the desired drastic speedup;
- another reason for the current speedup is that we moved to 8-bit numbers – but this is also probably the limit.

How can we achieve the desired future speedup? This is known as the *Grand Machine Learning Demand Challenge.*

Need to decrease energy consumption. In the modern world, computations take 10% of all the generated energy – and this amount grows exponentially. This is not sustainable, we need to decrease energy consumption.

Need to democratize large language models. At present, spectacular Large Language Models like ChatGPT require so much computer resources that only a few huge companies like Google or Microsoft can afford them. We need to make sure that smaller players can design such systems as well.

4.2 Need to go beyond (traditional) machine learning

What can we do if we do not have enough data? Current machine learning algorithms work well, but they require a large amount of information for training. In contrast, we humans can learn from a few examples – and learn fast. How can we make computer systems learn fast and from few examples?

4.3 Long-term problems

General problem: what next? Current machine learning techniques are on a roll, new successes arrive every day, and for a few years, as we increase the available amount of computations, we expect new and new spectacular successes. But, as with other techniques and ideas, eventually, we will reach a plateau. We have not reached this plateau, but it is inevitable. So, we need to be prepared, we need to start thinking about what next. Predicting the future is a difficult problem, but maybe some visionaries can guide us – after all:

- many AI ideas were already envisioned of Alan Turing way before actual computers appeared, and

- many ideas of quantum computing were promoted by Richard Feynman a few decades before technology enabled us to take them seriously.

Specific problem: how to deal with economy and finances. Finances are the lifeblood of modern economy. When properly maintained and regulated, financial decisions boost economy. On the other hand, mistakes can bring (and have brought) world economy into recession. Practitioners have tried to apply, to economy and finance, many computational techniques – including AI techniques. However, so far, successes of AI techniques in economics and finance have not been as spectacular as in many other application areas. AI can help predict weather, it can help predict the effect of newly synthesized medicines, but it is still not as good in predicting how the economy will react to different measures.

When we decide which engine to put on a plane, we use computational models. However, when we decide what minimal wage to set, what tax level would work best, we use political arguments instead of computations.

The current lack of success is a clear indication that the problems of economics and finance are very difficult – but many AI researchers believe that they are solvable. Solving them is thus an important challenge. It would be great to avoid political accusations – which are a lousy way to make decisions – and instead make decisions based on a clear understanding of what are the consequences of different alternatives.

Specific problem: how to best use AI in education. In spite of all the technical progress, in spite of all the promises, education is still largely similar to how it was in the first universities in the middle ages: a professor teaches, students learn. Lectures were at first only oral, then printed, now often online – but the process is largely the same. How can we use AI to boost education as successfully as AI have boosted many other application areas? And how to best avoid negative consequences of AI in education – such as possibility to use it for cheating?

5 What ideas can we use to solve these problems

5.1 How to increase our trust in AI systems

Educate people about the limitations of current AI systems. While AI system have achieved great successes, these successes have been over-hyped in the media – and this exaggerated picture of AI systems leads to unjustified fears. We researchers know the limitations of the current AI systems – everyone who played with ChatGPT knows that while it can provide a smooth essay on Columbus, it is so as good if you ask it a technical question; for such questions, answers are often wrong. We need to educate people about these limitations.

Can AI systems predict human behavior. One of the fears is that since we can now get the full person's genome for about \$300 (in comparison to billions of dollars that it cost when it started), knowing this genome will enable us to predict a person's behavior. This fear is caused by the fact that people make too much of genetics. Numerous studies of raised-apart identical twins convincing show that social interactions are almost as important than the genes is determining people's behavior. Again, we need to publicize this information.

5.2 How to maintain privacy

How to maintain privacy. Machine learning can design privacy-preserving systems the same way it designs algorithms that are less vulnerable to adversarial intervention: we simultaneously train two related systems:

- the first system that limits information (thus making it more private) while trying to make the remaining information as useful as possible, and
- the second system that tries to identify people based on the available limited information generated by the first system.

There are already successes in this direction: e.g., there is a system called Incognitus that automatically anonymizes medical notes.

In medical applications, privacy is not a big obstacle. While at present, AI systems do not provide perfect privacy, good news is that in medical applications, many patients are willing to provide as much information as possible – even if this may violate their privacy – because they hope and understanding that sharing all this information will help AI systems to come up with cures for their diseases.

5.3 How can we achieve the desired speedup?

Need for co-design. In the past, the speed up was mostly achieved by independently changing the computation precision, chips, and connections between the chips. The possibilities of such independent changes are, at present, almost exhausted. Thus, the only way to further speed up AI computations is to *co-design* all the components of the AI system: algorithms, chips, connections between the chips, etc. In other words, instead of using general-purpose chips, general-purpose processors, we need to have processors specialized for solving machine learning problems – since specialized software and hardware are always more effective and efficient than general-purpose ones.

Co-designing chips and communication between chips. To co-design chips and communications between the chips, it is important to take into account that communications inside the chip are much faster than between the chips. So, a natural conclusion is that we need to decrease the amount of communication between the chips. There are two main reasons why we still need such communication:

- the first reason is that with the large amount of data, we cannot fit everything on a single chip;

- the second reason is that at present, processing units get their data from memory – which is often located on a different chip.

To decrease the effect of the first reason, it is desirable to make chips as large as possible, so that each chip will be able to store as much information as possible. At present, there are two main problems why the chip size is limited:

- first, computers generate a large amount of memory, they need constant cooling, and larger chips are difficult to cool, and

- second, larger chips have a high probability of defects, which may make them useless.

There already are ideas that help to overcome these two problems:

- we can use the 3rd dimension to cool the chip, and

- we can add duplicate connections between in-chip processors, so that if one of the processors is not working, its neighbor can be used to replace it (this is similar to how defects in computer memory are usually handled).

These are not just theoretical ideas: there are already chips on the market – such as CS-2 – that implement these ideas to reach more than 50 times speedup.

To decrease the effect of the second reason, a natural idea is to place memory and processing units close to each other on the same chip so that, e.g., for matrix operations – that take most time of neural network training – matrix elements are located on the same chip close to the processing elements.

An important general idea is to make hardware reconfigurable – this is known as *plasticine architecture* – so that when new ideas and new designs appear, we will be able to use the current hardware without having to spend time and efforts on designing a new one.

Co-designing software and hardware: idea of sparsity. A priori, we do not know which data is useful and which is not, so we feed all this data into the neural network – and processing this large amount of data requires a lot of data processing. In reality, we do not need that much information to make useful decisions. For example, to recognize a cat, we do not to know the values of all the pixels on a 4000 by 4000 image: a few pixels sketching a cat are usually enough, and most details are irrelevant. So, while we start with the possibility that all inputs x_i are potentially relevant and that all intermediate results are potentially relevant, once the system learns, the results depends only on few inputs. In other words, in a general description of possible dependence, most coefficients are 0s. In such cases:

- for linear terms $\sum_i a_i \cdot x_i$, we say that a linear function is sparse;

- for quadratic terms $\sum_{i,j} a_{ij} \cdot x_i \cdot x_j$, we say that the corresponding matrix a_{ij} is sparse, etc.

This sparsity is very prominent on modern neural networks. Indeed, each neuron – the main processing element of such a network – transforms its inputs $x_1, \ldots, x_n$ into the signal $s(\sum w_i \cdot x_i + w_0, 0)$ for some *activation function* $s(z)$. Signals processed by neurons get processed by other neurons, etc. The objective of training is to find the weights for which, for all known examples, the difference between:

- what the network produces for this example and
- what output was actually observed,

is as small as possible. To minimize the corresponding objective function J – describing this difference – neural networks use *gradient descent*, when on the next step, the weights w_i are updated to $w_i - \alpha \cdot \dfrac{\partial J}{\partial w_i}$ for some constant α.

Modern neural networks use the activation function $s(z) = \max(0, z)$ which is known as *Rectified Linear Unit* (ReLU, for short). Fot this function, its derivative is 0 for half of the values. Thus, the derivative of J – which contains the composition of ReLUs – it also often 0. In other words, the resulting neural network is *sparse*. Moreover, many of the weights are very small, so ignoring them – and thus, making the corresponding matrices even more sparse, with up to 90% zeros – does not affect the effectiveness of the neural network. We can take this sparsity into account by only communicating non-zero values – this decreases the communication time by an order of magnitude.

We can go further and start analyzing what kind of sparsity do we have. Namely, the objective function J is a composition, mostly the composition of of ReLU expressions. For functions of one variable, the derivative $F'(x)$ of the composition $F(x) = f(g(x))$ is equal to the product $F'(x) = f'(g(x) \cdot g'(x)$ of the derivatives. For functions transforming tuples into tuples, the gradient is, similarly, the matrix product of the corresponding matrices $\dfrac{\partial y_i}{\partial x_j}$ – which are known as *Jacobians*.

To describe the corresponding sparsity, it is therefore reasonable to look for classes of sparse matrices that are closes under matrix multiplication – i.e., for which the product of two matrices from this class also belongs to this class. For the product $c_{ik} = \sum_j a_{ij} \cdot b_{jk}$, if this class has matrices with $a_{ij} \neq 0$ and matrices with $b_{jk} \neq 0$, then their product will contain a non-zero element c_{ik}. So, the relation $i \sim j$ – corresponding to the existence of elements with $a_{ij} \neq 0$ – is an equivalence relation. Thus, this equivalence relation divides all the indices into non-intersecting classes so that when i and j belong to different classes, then a_{ij} is 0. Such matrices are known as *block matrices*, since after an appropriate permutations, they only have blocks of non-zero elements close to the diagonal, while all other elements are 0s. Block matrices – and compositions of permutation and block matrix – are not enough to describe sparsity of neural networks; however, interestingly, compositions of a few such matrices – known as *butterfly matrices* – already provide a good description of a generic sparsity. To describe such compositions, we need fewer parameters that to describe a generic matrix and thus, processing matrices can be done faster.

Another general idea: two-stage training. Huge models are needed to get close-to-perfect training results. However, in the beginning, such models start from zero, and for some time, until they are fully trained, produce not-so-perfect results. But to reach the intermediate not-so-perfect results, we do not need these huge networks, we can reach the same not-so-perfect results with smaller networks, and thus, with fewer computations. This leads to the idea of 2-stage training:

- first, a smaller network provide initial training, and
- then the huge model is switched on, to transform the not-so-perfect results into close-to-perfect ones.

In this arrangement, the first stage does not require that much resources; as a result, we get an almost double speed-up.

Yet another general idea: using other optimization techniques. Gradient descent is the simplest – and not very effective – methods of numerical optimization. To speed up computations, it is therefore desirable to use more complex and more effective optimization techniques, e.g., evolutionary algorithms. In this, there are some promising preliminary results, but overall, this promising idea needs more research.

Another promising idea – that also already led to interesting results – is using machine learning techniques to come up with new faster algorithms.

Borrowing from biology: general idea. Many of the success computational ideas – neural networks, evolutionary algorithms – come from simulating nature. That nature is effective and efficient is not surprising: modern living creatures are the result of billions of years of improving evolution. It is therefore reasonable to see what other biology-inspired ideas we can use.

For example, current simple artificial neurons are a very crude approximation to much more complex biological neurons. So maybe some features of biological neurons can made our artificial neural networks more effective and more efficient? Several researchers tried, some of these results look promising – e.g., spiking neurons had some successful applications. So far, this idea has not yet led to drastic improvement, but the advantages of biological systems give us hope: e.g., the fact that a human brain with billions of neurons requires only about 20 Watt of power, several orders of magnitude smaller than an artificial neural network of comparable size.

Using domain-specific information. In specific application domains, there are often specific features that can speed up computations. For example, in medical application, it is important to know which DAN mutations are harmful and which are not. We all have many mutations, most of them are harmless.

In deciding which DNA mutations are harmful and which are not, we can use the fact that most DNAs have a high degree of similarity: even with dogs, we have 85% in common, and with primates, it is 98.5%. So, if a mutation is common in animals, it unlikely to be harmful – otherwise, evolution would have eliminated it long ago.

5.4 How can we decrease computers' energy consumption?

The fewer computations steps we require, the smaller amount of energy we consume. Thus, ideas that speed up computations automatically decrease energy consumption.

5.5 How can we democratize AI?

Similarly, when we drastically decrease the amount of computations needed to train a large language model, we thus allow smaller companies to design and train such models – and thus, to break the current alarming dominance of the few major players.

Another way to decrease dependence on large companies is not to use proprietary software, only use open-source software. At this moment, open-source AI systems are not yet as efficient as the company-provided ones, but open-source AI systems are catching up and hopefully, in a few years they will be as effective.

5.6 How to apply these ideas to other computational domains

While AI-based models are very important, there are also many other important practical problems that require a lot of computations – for example, problems of computational fluid dynamics (CFD) whose applications range from meteorology to medicine. In many of these problems, we have sparsity – for the same reason as in machine learning. In many of these problems, once we know an approximate solution which is close to the desired one, we can linearize the problem – which means that we need to deal with matrices. AI-related hardware and software help speed up computations related to sparsity and to matrix multiplication. It is therefore natural to apply the resulting hardware and software solutions to such problems. Some such applications have already led to a speed-up of these non-AI computations. The main challenge is that:

- in AI applications, 8-but accuracy – corresponding to relative accuracy of $2^{-8} \approx 0.4\%$ – is more than sufficient, while
- in many computational problems, approximation errors accumulate as we perform dynamic simulations, so to acheieve even a smaller accuracy like 10-20% in the final result, we often need to have 64-bit (or even higher) accuracy of each step of going from each moment of time to the next one.

It is, in principle, possible to perform 64-bit computations on an 8-bit machine, but this drastically slows down computations. It is desirable to decrease this slow-down.

5.7 What can we do if we do not have enough data?

One of the main reasons why we are able to learn from few examples is that we gave a large amount of prior knowledge. It is therefore desirable to utilize this knowledge. In the past, special AI techniques called *expert systems* were developed to deal with such knowledge. These techniques also included techniques for dealing with uncertainty: probabilistic, fuzzy (to deal with uncertainty described by natural language words like "small"), rough sets (to describe yes-no-unknown-type uncertainty), etc. The hope is that incorporating these techniques into machine learning will enable machine learning techniques to learn from few examples just like we do – or at least to learn from fewer examples than they need now.

5.8 How to best use AI in education?

General idea. The best way to study is when:

- a student gets the best professor in the world, and
- the attention of this professor is fully focused on this particular student.

How can we use this observation to improve education?

How to make sure that classes are given by the best professors. Some instructors are better, some are not so good. In the ideal world, everyone should take classes from the best professors. These best professors should teach them the material and provide feedback on their assignments. The problem is that there are only a few best professors, and, with the current technology, a professor can meaningfully provide feedback to a limited number of students – and the more students in the class, the more limited the feedback.

A solution used by many successful MOOCs is to use an AI-based system: this system classifies students answers into clusters. For the Stanford Intro to AI class, 90% of 100 000 students were divided into about 50 clusters – so that the same comment can be given to all the students from this cluster. The remaining 10% were encouraged to joint discussion forums with other students.

How to make sure that classes are as individualized as possible? Every student has his/her own pace: for each part of the material, some students learn is faster, some learn it slower. It is desirable to adjust this pace for each student. This is already possible, many system allow asynchronous as-you-learn learning schedule.

When a student asks a question, in most cases, we do not want to give a full answer – as many computer-based systems do. We want to provide prompts that will help students find the answer – this is how good human instructors answer student questions.

Different students also have different learning styles: some learn better from a smooth text, some from pictures and/or bullet points. A good textbook must provide different options.

Another possibility of individualization is related to the fact that while some core material is a must for every students, other topics may differ. It is therefore desirable to adjust these topics to the interests of each individual student – after all, research has shown that the main factor in student success is the student interest and enthusiasm. Again, a good textbook must allow such a choice. There are already textbooks like that – e.g., the most widely used textbook on Algorithms has core chapters and many additional chapters from which a professor and/or student(s) can select a few.

And we need to have a good notetaking system, that would help the student learn – and also provide feedback to the instructor, indicating which topics need adjustment.

How can we achieve all this?

- At present, AI systems are largely trained on question-answer pairs.
- For systems aimed at education, we need to train them on the *process* of reaching the answer.

What about cheating? On take-home assignments, students can, potentially, ask their friends to help; now they can also ask ChatGPT. From this viewpoint, nothing really changed. How can we avoid the use of ChatGPT for teaching? The same way we avoud the use of friends' help: maintain honor code, and have in-class test and quizzes.

For take-home assignments, just like now instructors usually ask students to explicitly explain what outside sources they used and how, it is reasonable to similarly ask students to indicate whether they used ChapGPT and is yes, how.

Remaining challenges.

- Human-to-human communication – whether oral or written – is a big problem for many CS students (and even for many CS professionals). Students are much better in writing effective and efficient code than in explaining how to use it and why they write the code this way. Hopefully, AI can help with that too, but it is not yet clear how it can help. Maybe students can use ChatGPT and similar systems to edit their texts – just like now they use spellcheckers and grammar checkers.
- Current education systems – in particular, systems that use AI – are good in delivering the material. However, the main predictor of the student success is not how this student is taught, but this student's motivation level. How to use AI to increase student motivation is still largely an open problem.
- Current systems teach specific subjects, but one of the important goals of education is to teach life skills, to develop general skills of reasoning and judgment, to teach a person to become a good citizen. How to make AI help with this objective is also largely an open problem.

6 Inspirational talks aimed at students and young professionals

This conference included an IEEE CIS Women in Computational Engineering event at which female researchers shared their experiences and gave advice to younger colleagues. This event was jointly organized by Sanaz Mostaghim, Vice President for Membership of the Computational Intelligence Society (CIS), and Keeley Crockett, Chair of the IEEE CIS Technical Committee on Ethical, Legal, Social, Environmental and Human Dimensions of AI/CI (SHIELD). At this event, inspirational talks were given by Catherine Huang, Senior Staff Software Engineer at Google, and by Nancy Min, Founder and CEO of ecoLong, an energy technology company.

Here is their advice to students and young professionals:

- *Selecting a job*: select a job in which you believe that what you are doing is useful, for which you are passionate.
- *Accept help and advice:* be open-minded, listen attentively to what others are saying.
- *Ask for help:* do not hesitate to ask for help if needed; even if you hear No, try other approaches – specialists in negotiations often say that No is a beginning of Yes.
- *Help others:* provide others with the opportunity to grow and to succeed; the only way for all of us to succeed is to help each other and work together.
- *What if you have an idea:* if you have an idea, do not hide it, do not be afraid that someone may steal it − most people have plenty of ideas already; instead, go out and tell it to 100 people, at least 2–3 will give you good advice and thus, help mentor you; if folks criticize your idea, do not take it personally.

7 Somewhat unusual features of the conference organization

The organizers used some innovative ideas to encourage people to attend the conference and to help attendees get as much as possible from this event.

Attracting attendees from academe. One of the reasons why many researchers hesitate to present their results at conferences is that they want to publish their results in serious journals, and journals do not like slightly extended versions of papers already published in conference proceedings.

To attract such researchers, the conference limited paper size to 4 pages:

- just enough to be eligible for IEEE Xplore, but

- small enough to allow authors to submit extended 8-page versions of their papers to journals without the journals worrying about extensions being not significant.

Attracting attendees from industry. Many conferences have industrial liaisons that try their best to attract industry folks to attend. In this process, the conference liaisons learn how to do it better – and then next year next conference starts often from scratch, without using the acquired expertise.

To resolve this problem, IEEE has started to organize, in addition to its usual structure of chapters etc., Industry Hubs. The first such hub – the IEEE Industry Hub in Silicon Valley – was very helpful in organizing the IEEE CAI 2023 conference.

Encouraging attendees to gain as much as possible. for each invited talk, at least 15 minutes were allocated for questions – much more than usual.

In addition to talks, panels, and other usual conference activity, the conference also scheduled special networking breaks when participants were welcome to talk to each other. These breaks were schedules in the same area as posters, exhibits, lunches, and coffee breaks. To make networking more focused, several tables in this area were marked by topics related to the conference "verticals": applications to energy, applications to medicine, etc.

Unusual arrangement of sponsors. Major sponsors got an opportunity to give a presentation about their companies during the plenary meetings – a very unusual arrangement for an academic conference, but an arrangement that helped this conference to get a lot of industry support.

Best papers: somewhat unusual arrangement. Most conferences have best paper awards. In this, the AI conference faced a real challenge. Indeed:

- In a more narrowly focused conference, it makes sense to compare papers and select the best ones.

- In a more general conference (like the annual SMC conference), papers clearly fall into a few areas: for SMC conferences, it is Systems, Cybernetics, and Human-Machine Interaction. It is difficult to compare papers from different areas, so at such conferences, usually, we select the best paper in each of the areas.

However, AI is a very broad field, with applications everywhere. It is difficult to compare papers from different subfields, and there are so many subfields that it is not realistic to select the best paper in each subfield. So instead, the organizers selected 3 best papers – without providing any ranking between them. They also selected 3 best posters.

8 For those who could not attend; thanks; and what's next

For those who could not attend. All the talks were recorded. In a few weeks, the recordings will be available for free on the IEEE CIS Resource Center https://resourcecenter.cis.ieee.org:

- IEEE members can get it by using their usual IEEE login,
- interested folks who are not IEEE members will need to create an account got this Center – but this account is free.

Thanks. Many thanks to Piero Bonissone and Gary Fogel, conference co-chairs, and their team for the great organization. Several SMC folks also participated in the organization, for example, Adrian Stoica served as government liaison, i.e., liaison to Government organizations like NASA and to National Labs.

Next year. Next year's IEEE Conference on AI – IEEE CAI 2024 – will be held in Singapore on June 25–27, 2024. The timing is selected in such a back-to-back way that the participants can also attend the IEEE World Congress on Computational Intelligence IEEE WCCI 2024 (Yokohama, Japan, June 30 – July 5, 2024), a biannual conference that combines IEEE conferences on fuzzy, on neural networks, and on evolutionary computation.

See you all there!

Figure 2: SMC representative participating in the conference discussions