ASSOCIATION OF STUDENTS OF COMPUTER SCIENCE FOR INFORMATION INTERCHANGE

ASCII

NEWSLETTER

FEBRUARY 2022

facebook.

Let's talk about it

WELCOME TO THE METAVERSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To be acclaimed internationally for excellence in teaching and research in Computer Science & Engineering, and in fostering a culture of creativity and innovation to responsibly harness state-of-the-art technologies for societal needs.

MISSION

Mission 1: To assist students in developing a strong foundation in Computer Science and Engineering by providing analytical, computational thinking and problem solving skills.
Mission 2: To inculcate entrepreneurial skills to develop solutions and products for interdisciplinary problems by cultivating curiosity, team spirit and spirit of innovation.
Mission 3: To provide opportunities for students to acquire knowledge of state-of-the-art in Computer Science and Engineering through industry internships, collaborative projects, and global exchange programmes with Institutions of international repute.
Mission 4: To develop life-long learning, ethics, moral values and spirit of service so as to contribute to the society through technology.
Mission 5: To be a premier research-intensive department by providing a stimulating environment for knowledge discovery and creation.

PROGRAM EDUCATIONAL OUTCOMES

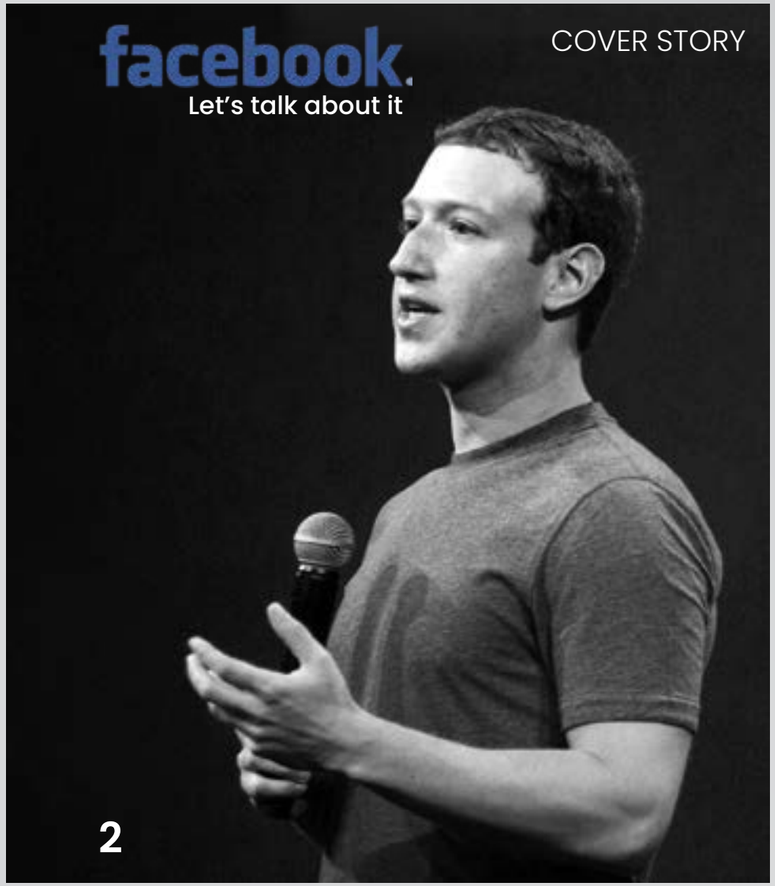
PEO1: Strive on a global platform to pursue their professional career in Computer Science and Engineering.
PEO2: Contribute to product development as entrepreneurs in inter disciplinary fields of engineering and technology.
PEO3: Demonstrate high regard for professionalism,integrity and respect values in diverse culture, and have a concern for society and environment.

PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3: Design and development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.
PSO1: Adopt Standard Practices: Ability to design and engineer, innovative, optimal and elegant computing solutions to interdisciplinary problems using standard practices, tools and technologies.
PSO2: Research and Innovation: Ability to learn emerging computing paradigms for research and innovation.

CONTENTS **FEBRUARY 2022**

ASCII



facebook
Let's talk about it

Nirmal K

The impact that the Internet has had on our life over the past two decades is nothing short of world changing. Thanks to the power of the Internet and air travel, the world has shrunk considerably, and conversations take place in real time across continents, and even between Earth and outer space.

Social media, one of the heaviest used applications on the Internet, has played a large role in this endeavour. However, there's one question being asked over and over, are we giving these large multinationals too much control over our lives? More than 3.6 billion people - half the population of the Earth can be found on Facebook, the largest social media platform.

"The company's leadership knows how to make Facebook and Instagram safer but won't make the necessary changes because they've put their astronomical profits before people", reads the statement published just a few weeks back by ex Facebook employee Francis Haugen, who has leaked internal research from Facebook that among other things, talks about how Instagram is harmful for teenage girls and how Facebook incites and promotes hate speech. Articles have been published in the Wall Street Journal based on her statements.

Francis Haugen

Facebook, a one trillion dollar giant, has been embroiled in privacy and data leak scandals in the past, and its CEO, Mark Zuckerberg, has been asked to testify in the US National Congress about the same, multiple times. This recent scandal's particularly damning as Facebook now has personal details, locations, facial patterns, friends and family, work and education history, likes and dislikes, sleep habits, social behaviour and much more.

It's long been stated that Facebook is causing a disintegration of society as we know it with how its algorithm works. The algorithm promotes content it thinks will get users to stay on and interact with the platform for the longest time possible, and these often end up being hate speech, dissent and rumour mongering posts. Engagement based ranking, which Facebook and the company's other apps follow, tend to push content that gets more engagement, and usually this is hateful, divisive, polarising or misinforming. Thus, it comes as a surprise to no one that Francis Hoegan has likened Facebook to cigarettes - harmful and addicting.
However, this was not the only Facebook news being run on news networks. Just by going through publicly shared records of people on Facebook, hackers have managed to collect personal information on 1.5 billion people by simply scraping the web which is now being sold in the dark web for cheap at a rate of about five thousand dollars for a million accounts. Name, email, phone number, gender, locations, date of birth and user IDs of public accounts. Information that by itself looks harmless but when combined with other details about oneself can scam artists and fraudulent advertisers. A person can pose as a bank official or a retailer using these details to make one believe that they are legit, and can make one click on links that can install malware which will provide backdoor access to PCs and mobile devices.

The fact that Facebook internal research states 'one in three teen girls felt that Instagram made them more depressed about themselves' and that Facebook the company hasn't done anything toward fixing that is astonishing. It has also been revealed that Facebook knew that Instagram posts driving eating disorders in young people.

This brings up an important question - should such companies be allowed to operate at full scale? Should they be required to split up into multiple entities? Should companies be forced to be more direct about the data they're collecting and how it's going to be used? As users, we have to take care not to share confidential or personal details on such platforms. It remains to be seen what legislation will be drawn up in the future to decrease privacy concerns. As of now, some of the suggested reforms include:

- Internal research should be shared with external parties
- Stronger oversight from Government bodies that demand transparency from such tech companies
- Replacing algorithms used on facebook's apps with the simpler chronological ranking

More importantly, should one trust sources on social networks? With the amount of misinformation that is being spread on such platforms, they have become the enemy. Originally created to make it easier for people to interact and share information, misuse has made it easier for misinformation to be shared and also has made it difficult for truth to be differentiated from lies.

Facebook is on damage control. The company has decided to stop the usage of facial recognition for tagging people in posts, and is bringing in other changes to fix its image. As the Facebook universe renames itself as Meta and starts focusing on developing a metaverse - a VR world where people can interact with each other in real time using the Internet, it is up to us users to behave ourselves on the Internet, taking care not to re-share negative or misinformative posts.

Meta

Popular Science: https://www.popsci.com/everything-facebook-knows-about-you/

Wall Street Journal: https://www.wsj.com/articles/the-facebook-files-11631713039

Machine Learning

Machine Learning is a branch of Artificial Intelligence and computer science. It allows software applications to learn from data and become more accurate in predicting the outcomes without being explicitly programmed. Machine learning algorithms takes past or historical data to predict new output values.

Surya Santhosh Reddy

FEATURES OF MACHINE LEARNING

- Machine Learning takes and uses the data to detect patterns and insights from the data.
- Training the model is must before testing the model.
- Test Data set should be more than Train Data to get more accurate results.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.
- Selecting feature selection enables the machine learning algorithm to train faster.
- Selecting efficient and accurate machine learning algorithm for the data will be key.

TYPES OF MACHINE LEARNING

- SUPERVISED
- UNSUPERVISED
- SEMI-SUPERVISED
- REINFORCEMENT

SUPERVISED LEARNING

Supervised learning, as the name indicates, it has the presence of a supervisor in it. Supervised learning is when we teach or train the machine or model using data that is well labelled along with target variables.

Example:
Given Data of fruits and the name of the fruits are given according to their shape. Now we will train the model with the given fruits data.

Now if we gave any fruit, for example, banana, as an input then the model will detect the fruit according to the shape by learning from the previous data, we provide and displays the output as banana.

Supervised Learning is classified into two categories:

1. **REGRESSION**: It is a model which contains mathematical functions that could predict a continuous variable using one or more independent variables.

Example: House Price prediction, iPhone cost prediction, etc...

2. **CLASSIFICATION**: It is a model which identifies or classifies the data to categories or groups based on what it learns from the previous data.

Example: Email is spam or not, Identifying colours, animals, fruits etc...

Types of Supervised Learning:

- Linear Regression

- Logistic Regression: It is used for classification problem. It models the probability of event with the probability lying between 0 and 1.

CLASSIFICATION

- Naive Bayes Classifiers: It is used for classification problem. It uses bayes theorem to classify the Objects.

- K-NN (k nearest neighbours): It is used for both regression and classification problem. It classifies given data point by comparing it with nearest neighbour data points.

- Decision Trees: It is used for both regression and classification problem. It classifies data by splitting the data by certain parameter and constructs trees with calculating information gain and entropy

- Support Vector Machine: It is used for both regression and classification problem. It develops models that assigns examples to new groups as classes to classify data.

UNSUPERVISED LEARNING

As the name suggests it is the opposite to supervised learning in terms of modelling the data. Here, we don't have labelled data or any predefined target variables. It allows the algorithm to act on the data without any guidelines. It groups the data by patterns and similarities derived by algorithm. No training will be provided.

Example:
If we give several images of cats and mice combined, our machine model will have no idea about the features of the cats and mice to categories them.

But it can categories by taking patterns and insights from the images like size, shape, colour, etc... So, here our machine works on its own and discovers the patterns from data and classifies them.

Unsupervised Learning is classified into two categories:

1. **CLUSTERING**: In this problem we discover the clusterings or groups from the data by some parameters.

Example: Grouping the customers according to their salary.

2. **ASSOCIATION**: In this problem we discover some rules or techniques that can describe the large data.

Example: In Netflix if we watch a movie in action genre, after watching the movie, Netflix app recommends all the movies in action genre which we tend to watch.

Types of Unsupervised Learning:

- K-means clustering
- KNN (k-nearest neighbours)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

SEMI-SUPERVISED LEARNING

In supervised learning there is a disadvantage which is, the data should be labelled by a machine learner or a data scientist. But for huge data it is very difficult to provide such labels. Whereas in unsupervised learning there is a greater disadvantage, since it is more difficult than supervised learning and its accuracy might also be less as there are no labelled data.

So, to overcome the above disadvantages, Semi-supervised learning is used. In this model the data contains both labelled and un-label data where majority of the data is un-label. Initially, some part of data is labelled using unsupervised learning and then using this data remaining data will be labelled.

REINFORCEMENT LEARNING

In reinforcement learning the model takes a suitable path to reach the destination in the maximum possible way accurately. Unlike supervised learning it has no answer or output. So it will decide what to do and learn from its experience.

Example: If we consider a route-finding puzzle game as given in picture, the cat needs to find the path to reach the cheese. Likewise, such a model will try all possible paths to reach the destination correctly.

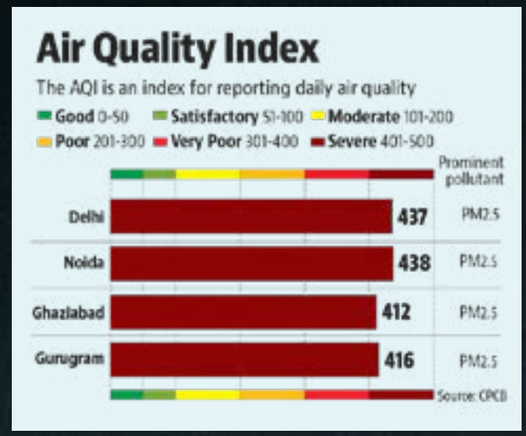
THE BIG CHANGE

HOW AI AND IOT CAN TACKLE AIR POLLUTION

Vaidehi Sridhar

When we think of air pollution, we often relate with largely populated cities with numerous resources, such as Delhi or Shanghai. For the joggers and runners among us, picking a cleaner path or choosing the right time to run makes a huge difference to our health. If you're health conscious, biking or jogging along a polluted route can be analogous to eating ice cream after every workout and hoping to lose weight. After all, family health is a basic factor in any decision. Keep in mind, when you think of how kids are exposed to pollution, half of all the children in Delhi.

As reported by the World Health Organisation (WHO), 9 out of 10 people around the world breathe polluted air. Imagine when you breathe in air with a pollution level of 500, when WHO recommends the ideal level to be 25. We use energy to drive our vehicles and power our houses, and not only these, even gas stoves and refrigerators that we use at our home contributes to air pollution. Indirectly, factories, nuclear plants and waste management sites also cause air pollution. Thanks to the huge amounts of toxic gaseous wastes that they release in the atmosphere. Today, AI and the Internet of Things (IOTs) have become a solution to help address air pollution. It can aid in better measurement, identify its sources, develop policies, and apply logic to solve the problem.



According to a report by Greenpeace, 22 out of 30 world's most-polluted cities are in India. As per recommendations, India requires a minimum of 4,000 air monitoring stations to check air quality. With 4.2 million deaths every year due to exposure to outdoor air pollution, we could leverage technology, artificial intelligence and the internet of things (IoT) with the capabilities from an increasing range of personal devices such as smartphones, smart watches and fitness trackers that measure an individual's heart rate, blood pressure, and breathing rate, which are indicators of overall health. We can also measure their changes with exposure to air pollutants such as oxides of nitrogen and sulphur. They also monitor spatial and GPS data, which, if combined could demonstrate the impact of the external environment on health factors and inform people about the issues. They also monitor infrastructure with satellite data and take inputs of data on human activities such as traffic, construction, garbage burning, industrial source apportionment, and population density. A feature engineered AI can be designed to utilize this data to accurately compute air quality. By understanding about the sources of pollution, AI can be further used to track and predict the growth and reduction of air pollution. For instance, we could predict whether an increase in industrial production can increase air pollution, or a decrease in vehicles can reduce it. These decisions could be evaluated by AI, allowing appropriate actions to be initiated.

AI can also help in modelling chemical reactions between pollutants. With the help of algorithms such as Atmospheric Transport Modelling System (ATMoS), helps us understand PM 2.5 (particulate matter with a diameter less than 2.5 micrometres) concentrations. Besides, there are several advanced algorithms that help to understand and predict smog, haze, visibility, and manage air quality better. It can provide daily updates on air quality levels and recommend safe spots to jog or exercise among others. AI can also track the improvement in the overall health of a person who have followed these measures.

AI can set fuel economy standards in the future. In California, USA, Ford is tackling global warming and air pollution using AI by incorporating machine learning and big data to achieve a more complete picture of the air people breathe. Many people are dying from the effects of air pollution than car crashes and gun violence combined. 46% of vehicle trips are less than three miles. With AI, you can have smart infrastructure, smart street lights and ecosystem tools. AI-powered edge IoT can reduce congestion, emissions and safety incidents and ease the flow of traffic.

Today, air quality is communicated to the general public through useful platforms which use measured and modelled data. With the results, we, as the general public, can minimize our personal exposure to pollutants as well as take certain steps to reduce it ourselves.

https://yourstory.com/2020/08/artificial-intelligence-improve-air-quality
https://www.capgemini.com/2019/06/beatairpollution-how-ai-and-iot-could-help-people-combat-air-pollution-issues
https://yourstory.com/socialstory/2020/04/understanding-air-quality-air-pollution-data-impact
https://youtu.be/aWMQDLNUM0U

Artificial Intelligence
being used to perform Medical Surgeries

Ramya Tadepalli

Introduction

AI helps surgeons to determine what is happening while performing a complex surgery by providing real time data points about the movements the surgeon makes during the procedure. In addition, AI is being used to provide analysis of a surgeon's technical abilities with products like **Care syntax's evident**, a Web-based surgical risk and quality management tool, designed to eliminate the "black box" of surgical visibility.

The robotic system is controlled by a surgeon, whose hand movements are converted into smaller, more precise movements that are performed by a set of "robot hands."

The device also uses AI to stabilize any tremors in the surgeon's movements, to ensure the robot correctly performs the procedure.

The Maastricht University Medical Centre that has used this confirmed that this AI assisted surgery went well and the This kind of AI assisted surgeries are helpful in performing procedures.

Why did I choose this topic?

So let me start my article with something that is actually not relevant to the above-mentioned topic but it's definitely about how I was interested in taking up this topic and writing an article about it.

When I was watching this amazing web series of "Good Doctor" that is streaming in Netflix, I came across the episode of "Intangibles" which showcased how AI helped the surgeons to do rehearsal for their extremely complex surgery. That's when it hit me and has pushed me to explore and write an article about this topic.

What is the article about?

The article gives a brief insight about how Artificial Intelligence is useful for the medical surgeons to perform surgeries.

How the robotic hand system works?

When performing robotic surgery using the da Vinci Surgical System:

- The surgeon works from a computer console in the operating room, controlling miniaturized instruments mounted on three robotic arms to make tiny incisions in the patient.
- The surgeon looks through a 3-D camera attached to a fourth robotic arm, which magnifies the surgical site.
- The surgeon's hand, wrist and finger movements are transmitted through the computer console to the instruments attached to the robot's arms. The mimicked movements have the same range of motion as the surgeon allowing maximum control.
- The surgical team supervises the robot at the patient's bedside.

The term "robotic" often misleads people. Robots don't perform surgery. Your surgeon performs surgery with da Vinci by using instruments that he or she guides via a console. The da Vinci surgical system gives your surgeon an advanced set of instruments to use in performing robotic-assisted minimally invasive surgery.

The da Vinci system translates your surgeon's hand movements at the console in real time, bending and rotating the instruments while performing the procedure. The tiny wristed instruments move like a human hand, but with a greater range of motion. The da Vinci vision system also delivers highly magnified, 3D high-definition views of the surgical area. The instrument size makes it possible for surgeons to operate through one or a few small incisions.

The three components of the da Vinci Surgical System are:

1. *Surgeon Console: The surgeon sits at the console, controlling the instruments while viewing your anatomy in a high-definition 3D (A format of motion picture that allows the viewer to perceive flat images in 3-D or life-like).*
2. *Patient Cart: It is positioned alongside the bed where the patient cart holds the camera and instruments that the surgeon controls from the console.*
3. *Vision Cart: The vision cart makes communication between components possible and supports the 3D high-definition vision system.*

Nukala Venkata Durga Sandeep

Tesla has become a household name as a leader and pioneer in the electric vehicle market, but it also manufactures and sells advanced battery and solar panel technology.

As a tech pioneer with a significant interest in the race to build and market autonomous vehicles, it makes sense that today they would be deeply interested in artificial intelligence. However, it was only this month that the business's billionaire founder and CEO Elon Musk publicly announced it is working on its own AI hardware.

This is definitely interesting if not exactly surprising. Musk, after all, has been outspoken in his views about AI. As well as revolutionizing almost every aspect of society, he has warned that it will cause widespread job losses and possibly even start World War Three.

He is also a co-founder of [Open AI], a research organization dedicated to ensuring that AI is developed and deployed in a safe, manageable way so as to minimize any existential risk robots may one day pose to humanity.

Not many details have yet been made public about Tesla's new AI, though it is believed it will process the "thinking" algorithms for the company's Autopilot software which currently gi ves Tesla vehicles limited ("level 2") levels of autonomous driving capability. Musk has said that he believes his cars will be fully autonomous (level 5 autonomous) by 2019.

The data is used to generate highly data-dense maps showing everything from the average increase in traffic speed over a stretch of road to the location of hazards which cause drivers to take action. Machine learning in the cloud takes care of educating the entire fleet, while at an individual car level, edge computing decides what action the car needs to take right now. A third level of decision-making also exists, with cars able to form networks with other Tesla vehicles nearby in order to share local information and insights. In a near future scenario where autonomous cars are widespread, these networks will most likely also interface with cars from other manufacturers as well as other systems such as traffic cameras, road-based sensors or mobile phones.

Although details are scarce on the new AI technology that Tesla was creating, its current AI – driven by a partnership with hardware manufacturer Nvidia – is largely based on an unsupervised learning model of machine learning.

On its Facebook page, Nvidia state that "In contrast to the usual approach to operating self-driving cars, we did not program any explicit object detection, mapping, path planning or control components into this car. Instead, the car learns on its own to create all necessary internal representations necessary to steer, simply by observing human drivers."

Whatever new tech it develops may veer away from this by stepping back into the more tested waters of supervised learning, where algorithms are trained beforehand about right or wrong decisions. However, it is possible that the theoretically greater gains achievable by truly unsupervised learning may keep them on this track.

Tesla has been criticized by some for appearing over-eager to be first to bring autonomous cars onto the roads, in the light of what is being seen as the first fatal accident involving a car which was driving it.

But as a business decision, it is hoping its pushy tactics will pay off, with experts concluding that the company has trumped its rivals in the data-gathering department. All the vehicles Tesla have ever sold were built with the potential to one day become self-driving, although this fact was not made public until 2014 when a free upgrade was rolled out. This means the company has had a lot more sensors out on the roads gathering data than most of its Detroit or Silicon Valley rivals, many of which are still at the concept stage.

Having just launched its first mass-market car, the Model 3 with a price tag of $35,000, the company is expecting the number of its vehicles on the road to increase by almost two thirds to around 650,000 in 2018 – and that's a lot of extra sensors.

In fact, all Tesla vehicles – whether or not they are Autopilot enabled – send data directly to the cloud. A problem with the engine operation meaning that components were occasionally overheating was diagnosed in 2014 by monitoring this data and every vehicle was automatically "repaired" by software patch thanks to this.

Tesla effectively crowdsources its data from all of its vehicles as well as their drivers, with internal as well as external sensors which can pick up information about a driver's hand placement on the instruments and how they are operating them. As well as helping Tesla to refine its systems, this data holds tremendous value in its own right. Researchers at McKinsey and Co estimate that the market for vehicle-gathered data will be worth $750 billion a year by 2030.

Tesla has clearly always been a company which has put data collection and analysis at the heart of everything it does. It isn't just design and manufacturing either, with the company processing customer data with AI and even parsing its online forum for text insights into common problems. Whether this focus will lead to victory in the upcoming battle for supremacy of the autonomous car market remains to be seen, but it has certainly provided itself with a head start.

FSD Chip

Build AI inference chips to run our Full Self-Driving software, considering every small architectural and micro-architectural improvement while squeezing maximum silicon performance-per-watt. Perform floor-planning, timing and power analyses on the design. Write robust tests and scoreboards to verify functionality and performance. Implement drivers to program and communicate with the chip, focusing on performance optimization and redundancy. Finally, validate the silicon chip and bring it to mass production in our vehicles.

Code Foundations

Throughput, latency, correctness and determinism are the main metrics we optimize our code for. Build the Autopilot software foundations up from the lowest levels of the stack, tightly integrating with our custom hardware. Implement super-reliable bootloaders with support for over-the-air updates and bring up customized Linux kernels. Write fast, memory-efficient low-level code to capture high-frequency, high-volume data from our sensors, and to share it with multiple consumer processes— without impacting central memory access latency or starving critical functional code from CPU cycles. Squeeze and pipeline compute across a variety of hardware processing units, distributed across multiple system-on-chips.

Dojo Systems

Design and build the Dojo system, from the silicon firmware interfaces to the high-level software APIs meant to control it. Solve hard problems with state-of-the-art technology for high-power delivery and cooling, and write control loops and monitoring software that scales. Work with every aspect of system design where the limit is only your imagination, employing the full prowess of our mechanical, thermal and electrical engineering teams to create the next-generation of machine learning compute for use in Tesla data centers. Collaborate with Tesla fleet learning to deploy training workloads using our massive datasets, and design a public facing API that will bring Dojo to the masses.

Neural Networks

Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and 3D objects directly in the top-down view. Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train. Together, they output 1,000 distinct tensors (predictions) at each timestep.

Autonomy Algorithms

Develop the core algorithms that drive the car by creating a high-fidelity representation of the world and planning trajectories in that space. In order to train the neural networks to predict such representations, algorithmically create accurate and large-scale ground truth data by combining information from the car's sensors across space and time. Use state-of-the-art techniques to build a robust planning and decision-making system that operates in complicated real-world situations under uncertainty. Evaluate your algorithms at the scale of the entire Tesla fleet.

Evaluation Infrastructure

Build open- and closed-loop, hardware-in-the-loop evaluation tools and infrastructure at scale, to accelerate the pace of innovation, track performance improvements and prevent regressions. Leverage anonymized characteristic clips from our fleet and integrate them into large suites of test cases. Write code simulating our real-world environment, producing highly realistic graphics and other sensor data that feed our Autopilot software for live debugging or automated testing.

Tesla Bot

Develop the next generation of automation, including a general purpose, bi-pedal, humanoid robot capable of performing tasks that are unsafe, repetitive or boring. We're seeking mechanical, electrical, controls and software engineers to help us leverage our AI expertise beyond our vehicle fleet.

Dojo Chip

Build AI training chips to power our Dojo system. Implement bleeding-edge technology from the smallest training nodes to the multi-die training tiles. Design and architect for maximum performance, throughput and bandwidth at every granularity. Dictate physical methodology, floor-planning and other physical aspects of the chip. Develop pre-silicon verification and post-silicon validation methods to ensure functional correctness. Write compilers and drivers to optimize power and performance for our neural networks throughout the entire Dojo system. For more information about Dojo's arithmetic formats and methods.

free and open source

Nirmal K

In today's world, the word 'free', at most times, doesn't mean something is actually free. For example, you buy a product and you get something free. Here, the 'free' product is being used to incentivize consumers into buying more products and thus is not 'free'.

However, there is still one industry where the word 'free' still carries its true meaning, the software industry. Software engineers invest their free time and valuable skills into developing software which they then provide to the general public for free. This activity makes no sense from a traditional profit point of view, but still, tens of thousands of developers do it. Lakhs of lines of code are written and periodically maintained by developers, very few of whom get donations for their work. You might start wondering why such developers exist, and why these tools are available at all.

To understand this, we need to look back in time, when the first hobbyist computers were coming up. Computers found initial use mostly in Universities and other academic institutions, who were happy sharing their advancements with the rest of the world. With the advent of Operating Systems from traditional companies, the world of free and open-source software saw itself grow smaller and smaller as more and more companies started selling software for profit.

In 1976, after the release of Microsoft's Altair BASIC (the first operating system written for the first hobbyist computer) Bill Gates wrote an essay titled 'Open Letter to Hobbyists' where he expressed his angst against the free sharing of his BASIC software. Companies stopped sharing source code outside the company and instead started sharing executables. Alarmed by this, Richard Stallman, a programmer, founded the GNU Project (GNU's Not UNIX) and started a non-profit organisation called the 'Free Software Foundation'. He then went on to invent the mechanism of copyleft, where users were allowed to read, modify source code and republish as long as they don't monetise it.

It is now the year 1991, and a young and brilliant Computer Science student by the name of Linus Torvalds has just released the Linux kernel for free, source code included. Finally, the collection of utilities, plugins, and add-ons that previously existed for other Operating Systems had a FOSS rallying point - around the Linux kernel.

Today, we have entire operating systems that are able to function normally with only FOSS components. One could be running Linux as the kernel, Gnome or Plasma as the desktop environment, Firefox for browsing, OpenOffice or LibreOffice for document work, and so on. Frameworks like NodeJS and React are developed and maintained by multinational corporations, and companies are now starting to pay for developers to work on open-source projects. Microsoft, a behemoth of the computing world, has bought GitHub, and also provides the Visual Studio Code editor for free, loved by its millions of users. Google develops and maintains the most popular mobile operating system in the world, Android for free.

Why? Here are the reasons that I find interesting, as an aspiring computer science professional. With events like the Hacktoberfest, we are seeing the Linus's Law manifest: Given enough eyeballs, all bugs are shallow'. If your source code is open, every developer can review it, suggest features, and optimise it. We get to tap the knowledge of some of the world's best software developers, from across continents.

Open source is safer. With thousands of people reviewing your code, any unintended mistakes are fixed and don't end up getting lost as they would if the code were not open source. This means there are lesser bugs that can be exploited, and that such bugs, when found, can be patched quickly. Bugs are not identified and hidden from sight.

Open source is also faster! Developers don't have to reinvent the wheel and instead have access to work that has already been done by other developers, which can be reused and built on top of. Imagine having to write a frontend and backend framework for every website you work on! Development time, developer time, and resources are all being saved, and everyone can contribute to whatever they're good at and reuse other components, instead of writing everything from scratch. Shorter development times and better quality of software.

There is one important clarification which is also the biggest reason why I think Open Source exists - open source / free software is not free in the economic sense. For example, RedHat and SUSE offer Linux distributions for enterprises for a cost. Zorin OS has a premium version that's paid. What this means is that the software source code is freely available for the user to read, edit, and redistribute. Adobe DC Reader on the other hand, while being available for free, is not free software in the sense that the code is closed source.

Finally, open-source helps developers learn. Many of us here are learning by working on open source projects. Events like GSoC (Google Summer of Code) help upcoming students to work on open-source projects, where they get to learn cutting-edge development and documentation methods, working alongside helpful experienced programmers. And hey, it's fun!

There are several other such reasons why I think we should, as consumers and programmers, prefer open-source alternatives, and should work on contributing to the world of open source. Please do reach out to me through the e-mail asciidata@cb.amrita.edu if you have any comments. Whatever the reason, open-source is here, and it's here to stay.

FEATURE

Prof. BHARAT JAYARAMAN

Professor Emiritus
Department of Computer Science and Engineering
University at Buffalo

Dean
Amrita School of Computing
Amrita Vishwa Vidyapeetham, Coimbatore

What inspired you to pursue research career in programming languages?

I started my studies in computer science (CS) in 1975 during my M.Tech at IIT-Madras. CS was not available at the bachelors level in those days as the discipline was still in its infancy. Research in core CS areas such as algorithms, programming languages, and operating systems were popular as many fundamental concepts were yet to be developed. In programming languages, for example, the widely used languages were Fortran and COBOL, and concepts such as object-oriented, functional and concurrent programming were still being researched. I always had a fascination for languages. During my Ph.D. studies (1977-81) I was interested in functional and concurrent languages; later, I developed interest in object-oriented, logic and constraint languages. Programming languages have connections with different subjects - artificial intelligence, software engineering, databases, computer architecture, etc. This makes it an important and stimulating subject for study and research.

What was your inspiration behind taking up research in JIVE?

Research in JIVE started in the mid-1990s and it grew out of teaching programming languages. I found that there weren't good notations for explaining program execution and giving a high-level insight into how a program executes. Run-time structures such as stacks are useful at an implementation level, but they are not always ideal for presenting a high-level view of execution to the programmer. With the advent of Java in the mid-1990s and its good libraries for visualizations, it became more feasible to implement a tool for visualizing the execution of a program. Eventually we developed the visualizations for Java itself, thanks to the libraries that Java provided for extracting run-time information.

For more :
https://www.academia.edu/53594280/Methodology_and_architecture_of_JIVE

Do you think that a visualization tool like JIVE should be part of programming languages from other paradigms too?

Yes, I think execution visualization will help all languages. Of course, the specific visual concepts to be used will vary from one language family to another. Visualizations are effective when they are not unwieldy. Thus there should be support for viewing things at different levels of granularity, for choosing what to visualize, and for run-time queries. The ease of implementing such tools will depend upon the support that a specific language provides for accessing its run-time information.

What is the future of Object-Oriented Programming languages? Where is it heading towards?

The software industry has made a huge investment in OO technologies and hence OOP will continue to be important for decades to come, especially for developing large-scale enterprise applications. The success of a language depends upon many factors. Good libraries are critical for ease of software development and hence we will see more libraries being added to cater to new kinds of applications. So, I think the core structure of object-oriented languages will endure and we will see embellishments in the form of libraries.

We are reading a course on Program Reasoning. In what way this course is important for CS students?

All programmers must know how to reason about programs in a precise manner – there should be no disagreement about this point. I take it that the question is more about the importance of formally proving (or, verifying) that a program is correct. It has always been a 'holy grail' of software engineering to be able to develop a piece of software and then give a formal proof that it is correct, thereby giving an assurance that it has no bugs. Such a goal has still not been achieved for real software with all its complexities. Thus the software industry will continue to rely on software testing as opposed to software verification.

Formal verification, however, is gaining importance for the class of safety-critical applications, where the cost of a bug is the loss of life, destruction of an aircraft, etc. Here, we want strong guarantees on the correctness of software as there could be catastrophic consequences of even a single bug. Today, there is a growing interest in verifying such applications and good tools are available for automating the construction of proofs. It is noteworthy that the leading computing companies such as Amazon, Facebook, Intel, and Microsoft are taking interest in verification, and so also are many aircraft, automotive and defense-related organizations.

There is also a close connection between the concerns of software verification and the concerns of software development. Good software design eases software verification; and conversely, difficulty in software verification may indicate that the software needs some re-design. So, the study of software verification will help in writing programs that are easier to understand and also in the design of programming languages that ease the task of software verification.

What will a future programming language look like?

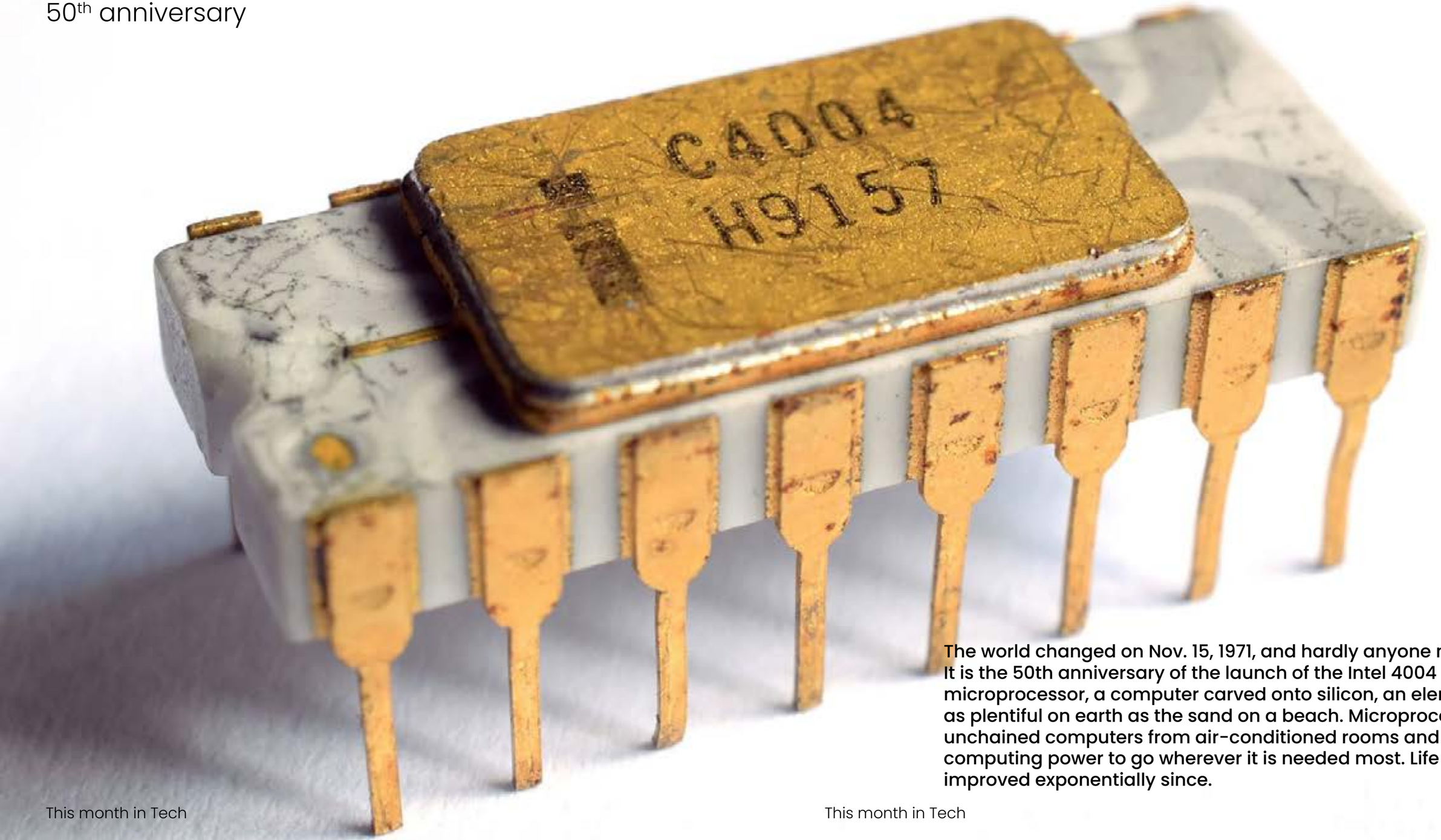
Software development and maintenance are costly and time-consuming and hence software vendors have a vested interest in the preservation of languages that are used to produce today's software; and, equally importantly, in ensuring that there is a trained workforce that is fluent in these languages. So, the well-known languages of today (Java, Python, C++, etc.) will be around for years to come.
Reflecting upon the last four decades, although the computing environment has changed dramatically from mainframes to mobile devices, the core structure of programming languages has evolved in a more gradual manner, with software libraries catering to the needs of graphical interfaces, parallelism, networking, machine learning, etc. I see this trend continuing into the future.
Speaking more broadly, there are some ideals that still have not been fully realized. One is software verification, discussed in question 5. Another is the 'correct by construction' ideal, which says that once a piece of software has been constructed it is guaranteed to be correct. The advantage of this approach is that we avoid the problem of software verification. For example, we can automatically construct a parser for a programming language given its grammar - this was illustrated for the C programming language in the late 1970's itself. The 'correct by construction' ideal has been more successful for domain-specific languages than for general-purpose languages. Crucial to this approach is a high-level specification, e.g., grammar, and a well-defined (and correct!) algorithm for generating code from the specification. Closely related to this discussion are software frameworks and model-driven development. Of course, this agenda is not about programming languages per se as it is about software construction.

To summarize, the core structure of programming languages may remain the same. Innovation is likely to be in libraries and also in automating aspects of software development.

> "All programmers must know how to reason about programs in a precise manner – there should be no disagreement about this point.

THE CHIP THAT CHANGED THE WORLD

50th anniversary



The world changed on Nov. 15, 1971, and hardly anyone noticed. It is the 50th anniversary of the launch of the Intel 4004 microprocessor, a computer carved onto silicon, an element as plentiful on earth as the sand on a beach. Microprocessors unchained computers from air-conditioned rooms and freed computing power to go wherever it is needed most. Life has improved exponentially since.

This month in Tech

NFT
Word of the Year

NFT is an abbreviation for a Non-Fungible Token, which according to the dictionary is “a unique digital certificate, registered in a blockchain, that is used to record the ownership of an asset such as an artwork or a collectible.”. The Collins Dictionary announced NFT as the word of the year. Perhaps the most well-known example of an NFT is “Every day’s: The First 5000 Days,” a collection of digital art made by graphic designer Mike Winkelmann, better known as Beeple. In March, an NFT for the piece sold for $69.3 million.

AI dupes AI
Alexa and Cortana

A team of researchers at the University of Chicago has found that voice-copying algorithms have advanced to the point that they are now capable of fooling voice recognition devices, and in many cases, people listening to them. The two algorithms used—SV2TTS and AutoVC—were tested by obtaining samples of voice recordings from publicly available databases.

Reservoir Computing

A relatively new type of computing that mimics the way the human brain works was already transforming how scientists could tackle some of the most difficult information processing problems. Reservoir computing is a machine learning algorithm developed in the early 2000s and used to solve the "hardest of the hard" computing problems, such as forecasting the evolution of dynamical systems that change over time.

PAYMENTS USING FACIAL RECOGNITION

Moscow on Friday rolled out a facial recognition payment system in the city's metro system, part of a rapid expansion of the controversial technology in Russia. To enter the metro, passengers do not need a card or a smartphone. They just need to look at the camera on the turnstile.

Screen time linked to Myopia in Youth

A new study published in one of the world's leading medical journals has revealed a link between screen time and higher risk and severity of myopia, or short-sightedness, in children and young adults. After analyzing and statistically combining the available studies, the authors revealed that high levels of smart device screen time, such as looking at a mobile phone, is associated with around a 30% higher risk of myopia and, when combined with excessive computer use, that risk rose to around 80%.

The Robinhood Hack

Robinhood Markets Inc. said that an intruder gained access to its systems last week and made off with the personal information of millions of its users. Though the company said that the breach has been contained, the personal details of several million users were already compromised by then.

Let's Google this week

Qubit

Quantum computing is a type of computation that harnesses the collective properties of quantum states, such as superposition, interference, and entanglement, to perform calculations. The devices that perform quantum computations are known as quantum computers. Though current quantum computers are too small to outperform usual (classical) computers for practical applications, they are believed to be capable of solving certain computational problems, such as integer factorization (which underlies RSA encryption), substantially faster than classical computers. It harnesses the phenomena of quantum mechanics to deliver a huge leap forward in computation to solve certain problems.

At the heart of quantum computing lies the idea of a "Qubit". These Qubits have a unique ability to store information in Quantum form. A Qubit is to quantum mechanics what the usual "bit" is to classical mechanics. They have a remarkable ability to possess 3 states. Concepts like Superposition and Entanglement enable the qubit to possess such ability.

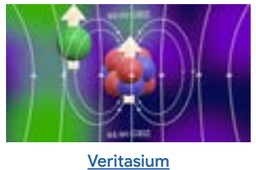
Qubits Explained:

Wikipedia

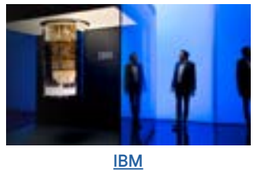
WhatIs

Courses:

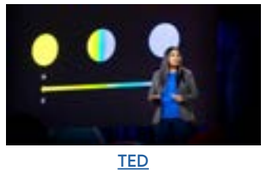
Quantum Computing Explained:



Veritasium
[How To Make a Quantum Bit]



IBM
[What is Quantum Computing?]



TED
[A beginner's guide to quantum computing | Shohini Ghose]

Quantum Computing (Seeker)
[We're Close to a Universal Quantum Computer, Here's Where We're At]

Events

Upcoming

Google DevFest 2021
Microsoft Azure Fiber Training
Microsoft 365 EduCon
Google Solutions Challenge

Just happened

AMRITA VISHWA VIDYAPEETHAM

Congratulations

to

Varadharajan K

and

Kanchamreddy Akshitha

for securing the First and Second places respectively in the **Republic Day Essay Writing Competitions** organised by the **NSS**.

Amrita Vishwa Vidyapeetham
conferred its
First **Honorary Doctorates** on

Dr. Jeffrey Sachs
World-Renowned Economist and Global Leader in Sustainable Development

Mr. Kailash Satyarthi
Nobel Peace Laureate and Children's Rights Activist

IEEE YESS'12

Congratulations

To **Dr. Prashant R. Nair**
Associate Professor
Department of Computer Science & Engineering
Amrita Vishwa Vidyapeetham, Coimbatore campus

For being re-appointed as

Member

2022 IEEE Member & Geographic Activities (MGA)
ieee.tv Advisory Committee

NITI Aayog AIM

I AM A
MENTOR OF CHANGE

Dr. Arunkumar C.(CSE)
Mr. Ritwik M. (CSE) &
Mr. K.P. Peeyush (ECE),

our professors from Amrita Vishwa Vidyapeetham, Coimbatore, have been selected as **Mentors of Change (MoC)** for Atal Tinkering Labs (ATL), the flagship initiative of Atal Innovation Mission (AIM), NITI Aayog by the Government of India.

CONTRIBUTORS

Nirmal K
CB.EN.U4CSE19038

Gautham Krishna
CB.EN.U4CSE19065

Vaidehi Sridhar
CB.EN.U4CSE20668

Manav Patnaik
CB.EN.U4CSE19033

Surya Santhosh Reddy
CB.EN.U4CSE19322

Ramya Tadepalli
CSEC.1001842

N Sandeep
CSEC.1008535

Varadharajan K
CB.EN.U4CSE19257

Adithi Narayan
CB.EN.U4CSE18205

Shriramya KR
CB.EN.U4ECE20255

Ayush Barik
CB.EN.U4CSE19106

Mahimanjali Lolla
CB.EN.U4CSE19128

NEWSLETTER TEAM

EDITOR	**Nirmal K** CB.EN.U4CSE19038	FACULTY	**Dr. Venkataraman D**
EDITOR	**Varadharajan K** CB.EN.U4CSE19257	FACULTY	**Mrs. Jeena K**
DESIGN	**Ayush Barik** CB.EN.U4CSE19106	FACULTY	**Mrs. Bagyammal T**
CONTENT	**Vaidehi Sridhar** CB.EN.U4CSE20668	EXECUTIVE	**Mahimanjali Lolla** CB.EN.U4CSE19128
CONTENT	**Shriramya KR** CB.EN.U4ECE20255	EXECUTIVE	**Adithi Narayan** CB.EN.U4CSE18205
CONTENT	**Manav Patnaik** CB.EN.U4CSE19033	CONTENT	**Gautham Krishna K** CB.EN.U4CSE19065

We’re looking for content

Are you interested in seeing your articles, reader reviews, or opinons published? We welcome content on technical / allied topics. The newsletter is a great place to share your knowledge and achievements with peers.

Mail your content/verifiable achievements to
asciidata@cb.amrita.edu
and be featured in an upcoming issue of the ASCII newsletter.

Find previous issues here.

OUR SOCIALS

Instagram : @ascii_amrita
Mail : asciidata@cb.amrita.edu