INFOLINE

VOLUME VIII ISSUE III

NOVEMBER 2017

SOUNDCLOUD

Viber

DEPARTMENT OF COMPUTER TECHNOLOGY AND INFORMATION TECHNOLOGY

KONGU ARTS AND SCIENCE COLLEGE

(Autonomous)

Affiliated to Bharathiar University, Coimbatore.

Approved by UGC and AICTE and Re-accredited by NAAC

ISO 9001:2015 Certified Institution

Nanjanapuram, Erode – 638 107.

KONGU

Assuring the Best

CONTENTS

NANO AIR VEHICLE (NAV)

The **AeroVironment Nano Hummingbird** or **Nano Air Vehicle** (NAV) is a tiny, remote controlled aircraft built to resemble and fly like a hummingbird, developed in the United States by AeroVironment, Inc. to specifications provided by the Defense Advanced Research Projects Agency (DARPA). The Hummingbird is equipped with a small video camera for surveillance and reconnaissance purposes and, for now, operates in the air for up to 11 minutes. It can fly outdoors, or enter a doorway to investigate indoor environments.

DARPA contributed $4 million to AeroVironment since 2006 to create a prototype "hummingbird-like" aircraft for the Nano Air Vehicle (NAV) program. The result was called the Nano Hummingbird which can fly at 11 miles per hour (18 km/h) and move in three axes of motion. The aircraft can climb and descend vertically, fly sideways left and right, forward and backward, rotate clockwise and counter-clockwise; and hover in mid-air. The artificial hummingbird using its flapping wings for propulsion and attitude control. It has a body shaped like a real hummingbird, a wingspan of 6.3 inches (160 mm), and a total flying weight of 0.67 ounces (19 g) less than an AA battery. This includes the systems required for flight: batteries, motors, and communications systems; as well as the video camera payload.

Technical goals

DARPA established flight test milestones for the Hummingbird to achieve and the finished prototype met all of them, and even exceeded some of these objectives:

- Demonstrate precision hover flight within a virtual two-meter diameter sphere for one minute.
- Demonstrate hover stability in a wind gust flight which required the aircraft to hover and tolerate a two-meter per second (five miles per hour) wind gust from the side, without drifting downwind more than one meter.
- Demonstrate a continuous hover endurance of eight minutes with no external power source.
- Fly and demonstrate controlled, transition flight from hover to 11

miles per hour fast forward flight and back to hover flight.

- Demonstrate flying from outdoors to indoors and back outdoors through a normal-size doorway.
- Demonstrate flying indoors "heads-down" where the pilot operates the aircraft only looking at the live video image stream from the aircraft, without looking at or hearing the aircraft directly.
- Fly the aircraft in hover and fast forward flight with bird-shaped body and bird-shaped wings.

The device is bigger and heavier than a typical real hummingbird, but is smaller and lighter than the largest hummingbird varieties. It could be deployed to perform reconnaissance and surveillance in urban environments or on battlefields, and might perch on windowsills or power lines, or enter buildings to observe its surroundings, relaying camera views back to its operator. According to DARPA, the Nano Air Vehicle's configuration will provide the warfighter with unprecedented capability for urban mission operations.

KALIKKUMAR V

III B.Sc. (Computer Technology)

REINFORCEMENT LEARNING

Definition

Reinforcement Learning is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behaviour within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behaviour this is known as the reinforcement signal.

There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms. In the problem, an agent is supposed to decide the best action to select based on his current state. When this step is repeated, the problem is known as a Markov Decision Process.

Why Reinforcement Learning?

Motivation

Reinforcement Learning allows the machine or software agent to learn its behaviour based on feedback from the environment. This behaviour can be learnt once and for all, or keep on adapting as time goes by. If the problem is modelled with care, some Reinforcement Learning algorithms can converge to the global optimum; this is the ideal behaviour that maximises the reward.

This automated learning scheme implies that there is little need for a human expert who knows about the domain of application. Much less time will be spent designing a solution, since there is no need for hand-crafting complex sets of rules as with Expert Systems, and all that is required is someone familiar with Reinforcement Learning.

How does Reinforcement Learning work?

Technology

As mentioned, there are many different solutions to the problem. The most popular, however, allow the software agent to select an action that will maximise the reward in the long term (and not only in the immediate future). Such algorithms are known to have infinite horizon.

In practice, this is done by learning to estimate the value of a particular state. This estimate is adjusted over time by propagating part of the next state's reward. If all the states and all the actions are tried a sufficient amount of times, this will allow an optimal policy to be defined; the action which maximises the value of the next state is picked.

When does Reinforcement Learning fail?

Limitations

There are many challenges in current Reinforcement Learning research. Firstly, it is often too memory expensive to store values of each state, since the problems can be pretty complex. Solving this involves looking into value approximation techniques, such as Decision Trees or Neural Networks. There are many consequence of introducing these imperfect value estimations, and research tries to minimise their impact on the quality of the solution.

Moreover, problems are also generally very modular; similar behaviours reappear often, and modularity can be introduced to avoid learning everything all over again. Hierarchical approaches are common-place for this, but doing this automatically is proving a challenge. Finally, due to limited perception, it is often impossible to fully determine the current state. This also affects the performance of the algorithm, and much work has been done to compensate this Perceptual Aliasing.

Who uses Reinforcement Learning?

Applications

The possible applications of Reinforcement Learning are abundant, due to the genericness of the problem specification. As a matter of fact, a very large number of problems in Artificial Intelligence can be fundamentally mapped to a decision process. This is a distinct advantage, since the same theory can be applied to many different domain specific problems with little effort.

In practice, this ranges from controlling robotic arms to find the most efficient motor combination, to robot navigation where collision avoidance behaviour can be learnt by negative feedback from bumping into obstacles. Logic games are also well-suited to Reinforcement Learning, as they are traditionally defined as a sequence of decisions: games such as poker, back-gammom, othello, chess have been tackled more or less succesfully.

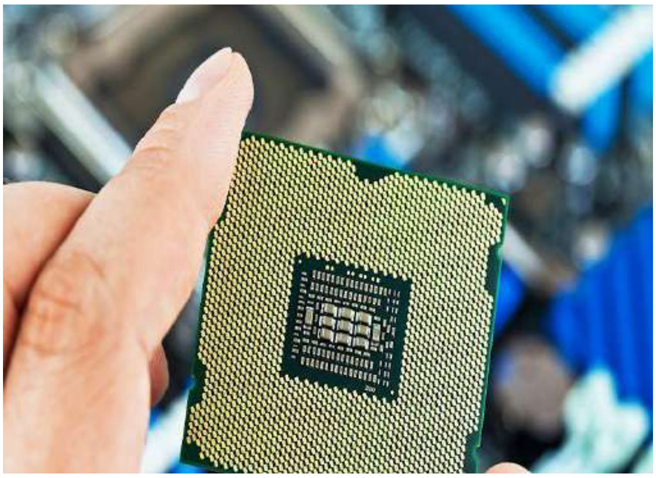
S.AISWARYA
I B.Sc. (Computer Technology)

GRAPHICS PROCESSING UNIT (GPU)

A Graphics Processing Unit (GPU) is a single-chip processor primarily used to manage and boost the performance of video and graphics. GPU features include:

- 2-D or 3-D graphics
- Digital output to flat panel display monitors
- Texture mapping
- Application support for high-intensity graphics software such as AutoCAD
- Rendering polygons
- Support for YUV color space
- Hardware overlays
- MPEG decoding

These features are designed to lessen the work of the CPU and produce faster video and graphics. A GPU is not only used in a PC on a video card or motherboard; it is also used in mobile phones, display adapters, workstations and game consoles.



The first GPU was developed by NVidia in 1999 and called the GeForce 256. This GPU model could process 10 million polygons per second and had more than 22 million transistors. The GeForce 256 was a single-chip processor with integrated transform, drawing and BitBLT support, lighting effects, triangle setup/clipping and rendering engines.

GPUs became more popular as the demand for graphic applications increased. Eventually, they became not just an enhancement but a necessity for optimum performance of a PC. Specialized logic chips now allow fast graphic and video implementations. Generally the GPU is connected to the CPU and is completely separate from the motherboard. The random access memory (RAM) is connected through the accelerated graphics port (AGP) or the peripheral component interconnect express (PCI-Express) bus. Some GPUs are integrated into the northbridge on the motherboard and use the main memory as a digital storage area, but these GPUs are slower and have poorer performance.

Most GPUs use their transistors for 3-D computer graphics. However, some have accelerated memory for mapping vertices, such as geographic information system (GIS) applications. Some of the more modern GPU technology supports programmable shaders implementing textures, mathematical vertices and accurate color formats. Applications such as computer-aided design (CAD) can process over 200 billion operations per second and deliver up to 17 million polygons per second. Many scientists and engineers use GPUs for more in-depth calculated studies utilizing vector and matrix features.

KALIKKUMAR V
III B.Sc. (Computer Technology)

THE HISTORY OF THE C PROGRAMMING LANGUAGE

C is one of the most important programming languages in the history of computing. Today, many different programming languages have popped up offering many different features, but in many ways, C provided the basis for such languages.

It is unclear whether its creators had envisioned the great things C would go on to achieve. Like most innovations, C underwent many changes over time. Probably one of its greatest achievements has been its ability to stay relevant even in modern, dynamic times. It must be fulfilling for the creators of C to observe that their creation is not considered outdated or categorized as useful for only a few niche areas. Instead, C has come to be recognized as a general-purpose, strong language which could be applied to many areas. (Find out more about the history of programming languages in Computer

Programming: From Machine Language to Artificial Intelligence.)

The Beginnings of C

Developing C was not originally the objective of its founders. In fact, various circumstances and problems created the ideal situation for its creation. In the 1960s, Dennis Ritchie, who was an employee of Bell Labs (AT&T), along with some of his colleagues, had been working on developing an operating system which could be used by many users simultaneously. This operating system was known as Multics, and it was meant to allow many users share common computing resources. Multics offered many benefits, but also had many problems. It was a large system and it seemed – from a cost-benefit perspective – that the costs outweighed the benefits. Gradually, Bell Labs withdrew from the project.

That's when Ritchie joined Ken Thompson and Brian Kernighan in another project. The project involved developing a new file system. Thompson developed a new file system for DEC PDP-7 supercomputer in assembly language. Thereafter, the creators of the file system made many improvements to it, resulting in the birth of the UNIX operating system. Even the origin of the name UNIX can be traced to its predecessor, Multics. Originally, the name was Unics (Uniplexed Information and Computing Service) as a pun on Multics (Multiplexed Information and Computer Services). Later, Unics changed to UNIX. UNIX was written in assembly language which, though ideal for machines, was a difficult proposition for human beings. To interpret and operate UNIX, the languages Fortran and B were used. It is here that the idea of developing the C language began to form in the minds of its creators.

Why C Was Developed?

The B language was a useful one in the context of the challenges the creators of UNIX faced with the operating system. The B language was taken from BCPL by Martin Richards. As already stated, UNIX was written in assembly language. To perform even small operations in UNIX, one needed to write many pages of code. B solved this problem. Unlike assembly language, B needed significantly fewer lines of code to carry out a task in UNIX. Still, there was a lot that B could not do. Much more was expected from B in the context of rapidly changing requirements. For example, B did not recognize data types. Even with B, data types were expressed with machine language. B also did not support data structures.

It was clear that something had to change. So, Ritchie and his colleagues got down to overcoming the limitations. The C language was developed in 1971-73. Note that for all its limitations, C owes its birth to B because C retained a lot of what B offered, while adding features such as data types and data structures. The name C was chosen because it succeeded B. In its early days, C was designed keeping UNIX in mind. C was used to perform tasks and operate UNIX. So, keeping performance and productivity in mind, many of the UNIX components were rewritten in C from assembly language. For example, the UNIX kernel itself was rewritten in 1973 on a DEC PDP-11.

Ritchie and Kernighan documented their creation in the form of a book called "The C Programming Language." Though Kernighan claimed that he had no role in the design of C, he was the author of the famous "Hello World" program and many other UNIX programs.

Evolution of C

Over time, C began to be used in personal computers for developing software applications and other purposes. The first change (even if only a little) came when the American National Standards Institute (ANSI) formed a committee in 1983 to standardize C. After a review of the language, they modified it a little so that it was also compatible with other programs that preceded C. So the new ANSI standard came into being in 1989, and is known as ANSI C or C89. The International Organization for Standardization (ISO) has also contributed to the standardization of C.

C has evolved as it has added some significant features like memory management, functions, classes and libraries to its rich feature set. C is being used in some of the biggest and most prominent projects and products in the world. C has also influenced the development of numerous languages such as AMPL, AWK, csh, C++, C--, C#, Objective-C, Bit C, D, Go, Java, JavaScript, Julia, Limbo, LPC, Perl, PHP, Pike, Processing, Python, Rust, Seed7, Vala and Verilog. (To learn more about languages, see The 5 Programming Languages That Built the Internet.)

Do you use Microsoft Windows? Then you have C to thank, because its development is mostly in C. The same goes for MacOS, Linux, Android, iOS and Windows Phone as well, so nearly all modern operating systems are based on C. It's also widely used in embedded systems, such as those found in vehicles, smart TVs and countless internet of things (IoT) devices.

VARSHA R
I B.Sc. (Information Technology)

FOG COMPUTING

Fog computing or fog networking, also known as fogging, is an architecture that uses one or more collaborative end-user clients or near-user edge devices to carry out a substantial amount of storage (rather than stored primarily in cloud data centers), communication (rather than routed over the internet backbone), control, configuration, measurement and management (rather than controlled primarily by network gateways such as those in the LTE core network).

Fog computing can be perceived both in large cloud systems and big data structures, making reference to the growing difficulties in accessing information objectively. This results in a lack of quality of the obtained content. The effects of fog computing on cloud computing and big data systems may vary yet, a common aspect that can be extracted is a limitation in accurate content distribution, an issue that has been tackled with the creation of metrics that attempt to improve accuracy.

Fog networking consists of a control plane and a data plane. For example, on the data plane, fog computing enables computing services to reside at the edge of the network as opposed to servers in a data-center. Compared to cloud computing, fog computing emphasizes proximity to end-users and client objectives, dense geographical distribution and local resource pooling, latency reduction and backbone bandwidth savings to achieve better quality of service (QoS) and edge analytics/stream mining, resulting in superior user-experience and redundancy in case of failure while it is also able to be used in AAL scenarios.

Fog networking supports the Internet of Things (IoT) concept, in which most of the devices used by humans on a daily basis will be connected to each other. Examples include phones, wearable health monitoring devices, connected vehicle and augmented reality using devices such as the Google Glass.

SPAWAR, a division of the US Navy, is prototyping and testing a scalable, secure Disruption Tolerant Mesh Network to protect strategic military assets, both stationary and mobile. Machine control applications, running on the mesh nodes, "take over", when internet connectivity is lost. Use cases include Internet of Things e.g. smart drone swarms.

ISO/IEC 20248 provides a method whereby the data of objects identified by edge computing using Automated Identification Data Carriers [AIDC], a barcode and/or RFID tag, can be read, interpreted, verified and made available into the "Fog" and on the "Edge" even when the AIDC tag has moved on.Both cloud computing and fog computing provide storage, applications, and data to end-users. However, fog computing has a bigger proximity to end-users and bigger geographical distribution.

Cloud Computing – the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. Cloud Computing can be a heavyweight and dense form of computing power.

Fog computing – a term created by Cisco that refers to extending cloud computing to the edge of an enterprise's network. Also known as Edge Computing or fogging, fog computing facilitates the operation of compute, storage, and networking services between end devices and cloud computing data centres. While edge computing is typically referred to the location where services are instantiated, fog computing implies distribution of the communication, computation, and storage resources and services on or close to devices and systems in the control of end-users. Fog computing is a medium weight and intermediate level of computing power.

Mist computing – a lightweight and rudimentary form of computing power that resides directly within the network fabric at the extreme edge of the network fabric using microcomputers and microcontrollers to feed into Fog Computing nodes and potentially onward towards the Cloud Computing platforms. National Institute of Standards and Technology initiated in 2017 the definition of the Fog computing that defines Fog computing as an horizontal, physical or virtual resource paradigm that resides between smart end-devices and traditional cloud computing or data center. This paradigm supports vertically-isolated, latency-sensitive applications by providing ubiquitous, scalable, layered, federated, and distributed computing, storage, and network connectivity.

J.JANSI RANI
II B.Sc. (Computer Technology)

BIG DATA ANALYTICS

The volume of data that one has to deal has exploded to unimaginable levels in the past decade, and at the same time, the price of data storage has systematically reduced. Private companies and research institutions capture terabytes of data about their users' interactions, business, social media, and also sensors from devices such as mobile phones and automobiles. The challenge of this era is to make sense of this sea of data. This is where big data analytics comes into picture.

Big Data Analytics largely involves collecting data from different sources, munge it in a way that it becomes available to be consumed by analysts and finally deliver data products useful to the organization business.

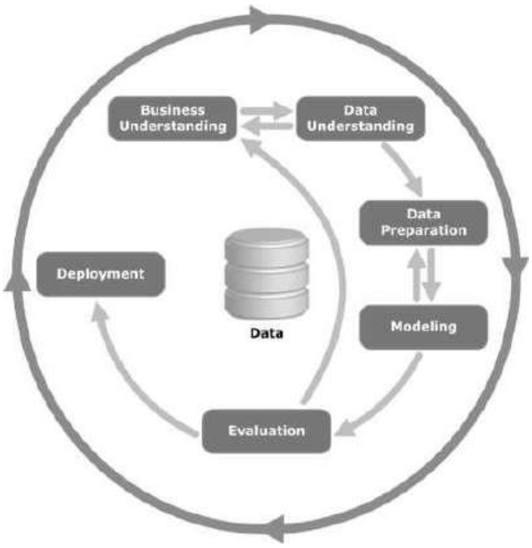
The process of converting large amounts of unstructured raw data, retrieved from different sources to a data product useful for organizations forms the core of Big Data Analytics.

Traditional Data Mining Life Cycle

In order to provide a framework to organize the work needed by an organization and deliver clear insights from Big Data, it’s useful to think of it as a cycle with different stages. It is by no means linear, meaning all the stages are related with each other. This cycle has superficial similarities with the more traditional data mining cycle as described in CRISP methodology.

CRISP-DM Methodology

The CRISP-DM methodology that stands for Cross Industry Standard Process for Data Mining, is a cycle that describes commonly used approaches that data mining experts use to tackle problems in traditional BI data mining. It is still being used in traditional BI data mining teams.



The stages involved in the CRISP-DM life cycle are:

Business Understanding: This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data mining problem definition. A preliminary plan is designed to achieve the objectives. A decision model, especially one built using the Decision Model and Notation standard can be used.

Data Understanding: The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

Data Preparation: The data preparation phase covers all activities to construct the final dataset (data that will be fed into the modeling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection as well as transformation and cleaning of data for modelling tools.

Modelling : In this phase, various modelling techniques are selected and applied and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, it is often required to step back to the data preparation phase.

Evaluation : At this stage in the project, you have built a model (or models) that appears to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to evaluate the model thoroughly and review the steps executed to construct the model, it properly achieves the business objectives.

Deployment: Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that is useful to the customer.

Big Data Life Cycle

In today's big data context, the previous approaches are either incomplete or suboptimal. For example, the SEMMA methodology disregards completely data collection and pre-processing of different data sources. These stages normally constitute most of the work in a successful big data project.

A big data analytics cycle can be described by the following stage

- Business Problem Definition
- Research
- Human Resources Assessment
- Data Acquisition
- Data Munging
- Data Storage
- Exploratory Data Analysis
- Data Preparation for Modelling and Assessment
- Modelling
- Implementation

In this section, we will throw some light on each of these stages of big data life cycle.

Business Problem Definition

This is a point common in traditional BI and big data analytics life cycle. Normally it is a non-trivial stage of a big data project to define the problem and evaluate correctly how much potential gain it may have for an organization. It seems obvious to mention this, but it has to be evaluated what are the expected gains and costs of the project.

Research

Analyze what other companies have done in the same situation. This involves looking for solutions that are reasonable for your company, even though it involves adapting other solutions to the resources and requirements that your company has. In this stage, a methodology for the future stages should be defined.

Human Resources Assessment

Once the problem is defined, it's reasonable to continue analyzing if the current staff is able to complete the project successfully. Traditional BI teams might not be capable to deliver an optimal solution to all the stages, so it should be considered before starting the project if there is a need to outsource a part of the project or hire more people.

Data Acquisition

This section is key in a big data life cycle; it defines which type of profiles would be needed to deliver the resultant data product. Data gathering is a non-trivial step of the process; it normally involves gathering unstructured data from different sources. To give an example, it could involve writing a crawler to retrieve reviews from a website. This involves dealing with text, perhaps in different languages normally requiring a significant amount of time to be completed.

Data Munging

Once the data is retrieved, for example, from the web, it needs to be stored in an easy to-use format. To continue with the reviews examples, let's assume the data is retrieved from different sites where each has a different display of the data. Suppose one data source gives reviews in terms of rating in stars, therefore it is possible to read this as a mapping for the response variable $\mathbf{y \in \{1, 2, 3, 4, 5\}}$. Another data source gives reviews using two arrows system, one for up voting and the other for down voting. This would imply a response variable of the form **y ∈ {positive, negative}**.

In order to combine both the data sources, a decision has to be made in order to make these two response representations equivalent. This can involve converting the first data source response representation to the second form, considering one star as negative and five stars as positive. This process often requires a large time allocation to be delivered with good quality.

Data Storage

Once the data is processed, it sometimes needs to be stored in a database. Big data technologies offer plenty of alternatives regarding this point. The most common alternative is using the Hadoop File System for storage that provides users a limited version of SQL, known as HIVE Query Language. This allows most analytics task to be done in similar ways as would be done in traditional BI data warehouses, from the user perspective. Other storage options to be considered are MongoDB, Redis, and SPARK.

This stage of the cycle is related to the human resources knowledge in terms of their abilities to implement different architectures. Modified versions of traditional data warehouses are still being used in large scale applications. For example, teradata and IBM offer SQL databases that can handle terabytes of data; open source solutions such as postgreSQL and MySQL are still being used for large scale applications.

This stage a priori seems to be the most important topic, in practice, this is not true. It is not even an essential stage. It is possible to implement a big data solution that would be working with real-time data, so in this case, we only need to gather data to develop the model and then implement it in real time. So there would not be a need to formally store the data at all.

Exploratory Data Analysis

Once the data has been cleaned and stored in a way that insights can be retrieved from it, the data exploration phase is mandatory. The objective of this stage is to understand the data, this is normally done with statistical techniques and also plotting the data. This is a good stage to evaluate whether the problem definition makes sense or is feasible.

Data Preparation for Modelling and Assessment

This stage involves reshaping the cleaned data retrieved previously and using statistical preprocessing for missing values imputation, outlier detection, normalization, feature extraction and feature selection.

Modelling

The prior stage should have produced several datasets for training and testing, for example, a predictive model. This stage involves trying different models and looking forward to solving the business problem at hand. In practice, it is normally desired that the model would give some insight into the business. Finally, the best model or combination of models is selected evaluating its performance on a left-out dataset.

Implementation

In this stage, the data product developed is implemented in the data pipeline of the company. This involves setting up a validation scheme while the data product is working, in order to track its performance. For example, in the case of implementing a predictive model, this stage would involve applying the model to new data and once the response is available, evaluate the model.

As mentioned in the big data life cycle, the data products that result from developing a big data product are in most of the cases some of the following :

Machine learning implementation: This could be a classification algorithm, a regression model or a segmentation model.

Recommender system: The objective is to develop a system that recommends choices based on user behaviour. **Netflix** is the characteristic example of this data product, where based on the ratings of users, other movies are recommended.

Dashboard : Business normally needs tools to visualize aggregated data. A dashboard is a graphical mechanism to make this data accessible.

Ad-Hoc analysis : Normally business areas have questions, hypotheses or myths that can be answered doing ad-hoc analysis with data.

V. MOHANAPRIYA
III B.Sc. (Computer Technology)

COMMON NETWORK PROBLEMS

Slow internet: Slow internet is the bane of every worker's existence. It could be caused by a number of problems. For example, your router may not be working properly, you may have too many devices attempting to access the internet at once, a specific app might be drawing too heavily on your total bandwidth, or your internet provider itself may be experiencing service delays. Unfortunately, the only way to address these potential issues is to investigate them and attempt to fix them, one by one, until you find the root of the problem.

A signal without a connection: Occasionally, you'll see a signal from a router, but your device won't be able to connect to the network. There are two potential causes for this first, your device might be out of range of the router. Move the device close to the router, and see if you can connect then. If you still can't connect, there's probably a problem with the hardware. You may be able to replace the network card you're using, or update the drivers associated with it, but in some cases, you'll need to replace the hardware entirely.

Periodic outages: Few networks operate perfectly 100 percent of the time, but if you're seeing periodic total outages a complete inability for any devices to connect to the network you have a problem that requires action. There are many root causes here, but you may be seeing a NetBIOS conflict (especially if you're using an older system). If you disable WINS/NetBT name resolution, you may be able to clear things up. You could also try renaming computers and domains to resolve the issue.

IP conflicts: Windows usually makes sure that there's only one IP address per device that has access to the network at any given time. In rare situations, however, two devices may end up with the same IP address; when this happens, one device will usually be "blocked," which prevents the device from accessing protected files. To make matters worse, it can cause lag for all connected devices not just the ones with the IP conflict. Avoiding this problem is relatively easy if you reconfigure your DHCP setup to exclude static IP addresses. This should reconfigure IP addresses so that all machines can access the network without issue.

VOIP quality issues: Issues with voice calls including delays, interruptions, and quality issues can be caused by many variables (including some of the ones listed above). However, you may be experiencing a network stutter. Install jitter buffers, which create small caches or packets of VOIP information, to ensure a smooth stream of information from one point to another. You could also install new playback codes preferably ones with packet loss concealment as a main feature. While you're at it, update your drivers.

Connections with limited access. If you have a connection with limited access, you're likely receiving a Windows error message that's caused by a technical glitch. Windows has released updates that should prevent the majority of these errors from occurring, but if you encounter this situation, your best bet is to do a hard reset of your network router and the device trying to connect to it.

MYTHILI T
III B.Sc. (Information Technology)

BRIDGING THE HUMAN-COMPUTER INTERACTION

Compare that with how humans interact with each other, face to face smiling, frowning, pointing, tone of voice all lend richness to communication. With the goal of revolutionizing everyday interactions between humans and computers, Colorado State University researchers are developing new technologies for making computers recognize not just traditional commands, but also non-verbal ones gestures, body language and facial expressions.

The project, titled "Communication through Gestures, Expression and Shared Perception," is led by Professor of Computer Science Bruce Draper, and is bolstered by a recent $2.1 million grant from the Defense Advanced Research Projects Agency (DARPA) under its "Communicating with Computers" funding program. "Current human-computer interfaces are still severely limited," said Draper, who is joined on the project by CSU researchers from the computer science and mathematics departments. First, they provide essentially one-way communication users tell the computer what to do. This was fine when computers were crude tools, but more and more, computers are becoming our partners and assistants in complex tasks. Communication with computers needs to become a two-way dialogue."

The team has proposed creating a library of what are called Elementary Compostable Ideas (ECIs). Like little packets of information recognizable to computers, each ECI contains information about a gesture or facial expression, derived from human users, as well as a syntactical element that constrains how the information can be read. To achieve this, the researchers have set up a Microsoft Kinect interface. A human subject sits down at a table with blocks, pictures and other stimuli. Their goal is to make computers smart enough to reliably recognize non-verbal cues from humans in the most natural, intuitive way possible. According to the project proposal, the work could someday allow people to communicate more easily with computers in noisy settings, or when a person is deaf or hard of hearing, or speaks another language.

SELVA BHARATHI A
I B.Sc. (Computer Technology)

ADAPTIVE SECURITY ARCHITECTURE

Adaptive architecture is a system which changes its structure, behaviour or resources according to demand. The adaptation made is usually to non-functional characteristics rather than functional ones.

Something of a misnomer, because the thing that adapts is the working system, rather than the (more abstract) architecture which defines the adaptability that is required of that system. Adaptive software architecture: Used by programmers in relation to a program. An adaptive algorithm "is an algorithm which changes its behaviour based on the resources available. For example, in the C++ Standard Library, the stable partition [program] acquires as much memory as it can get and applies the algorithm using that available memory."

Adaptive infrastructure architecture: Used by infrastructure engineers in relation to the configuration of processors. The computing resources used by applications (the partition size, or the number of servers in a cluster, or the share of a processor, or the number of processes) are configured so that they shrink or grow with demand.

Adaptive business architecture: Could also be used (for example) in connection with a workflow system that assigns human resources to a task or service to match the demand for that task or service. An organization structure that flexes in response to business changes.

RASIKA M
III B.Sc. (Computer Technology)

DIGITAL PLATFORM

A digital platform consists of many services, representing a unique collection of software or hardware services of a company used to deliver its digital strategy. Organizations look for the services that provide businesses with the best performance to cost ratio, but some services are almost always required for all applications or solutions. Digital platforms have gained importance in business due to this 'performance to cost ratio' advantage.

The digital platform strategy will vary from company to company depending on their internal workflow strategy. Some companies will develop a platform business model that encompasses providers, consumers, and employees to create or exchange goods, services, and social interaction while some integrate with other organizations' digital platforms. Regardless of the setup, the platform strategy must be able to integrate business and IT needs to establish a collective leadership vision. Companies must decide what makes sense for their organization and long-term business goals.

The use of platform thinking extends beyond the tech sector onto the marketing industry. The marketing retailers are swiftly shifting from distribution channels selling products, to engagement platforms co-creating value. Online retailers like eBay, Etsy, and Amazon led the way, and now traditional retailers are following the same path. The innovation around latest platform abilities are being driven by three transformative technologies: cloud, social, and mobile. The cloud enables a global infrastructure for production, allowing everyone to create content and applications for a global audience. Social networks connect people globally and maintain their identity online. Mobile devices such as mobiles and tablets allow connection to this global infrastructure anytime, anywhere. The result is a globally accessible network of entrepreneurs, workers, and consumers who are available to create businesses, contribute content, and purchase goods and services. The future has set out to see more and more companies shifting from products to platforms.

Digital platforms allow for companies to pursue business models that bring together multiple buyers and sellers. For example, designers of smart cities will build ecosystems to bring together partners of all sizes. Banks will bring new products and services to customers via an ecosystem of fintech (financial technology).

G.LOKESHWAR
II B.Sc. (Information Technology)

MESH APP AND SERVICE ARCHITECTURE

The device mesh refers to an expanding set of endpoints people use to access applications and information or interact with people, social communities, governments and businesses. The device mesh includes mobile devices, wearable, consumer and home electronic devices, automotive devices and environmental devices such as sensors in the Internet of Things (IoT).

Mesh networks can relay messages using either a flooding technique or a routing technique. With routing, the message is propagated along a path by hopping from node to node until it reaches its destination. To ensure that all its paths are available, the network must allow for continuous connections and must reconfigure itself around broken paths, using self-healing algorithms such as Shortest Path Bridging. Self-healing allows a routing-based network to operate when a node breaks down or when a connection becomes unreliable. As a result, the network is typically quite reliable, as there is often more than one path between a source and a destination in the network. Although mostly used in wireless situations, this concept can also apply to wired networks and to software interaction

G.LOKESHWAR
II B.Sc. (Information Technology)

BLOCKCHAIN TECHNOLOGY

A blockchain, originally block chain, is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a hash pointer as a link to a previous block, a timestamp and transaction data. By design, blockchains are inherently resistant to modification of the data. The Harvard Business Review describes it as "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way." For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without the alteration of all subsequent blocks, which requires collusion of the network majority.

Blockchains are secured by design and are an example of a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been achieved with a blockchain. This makes blockchains potentially suitable for the recording of events, medical records, and other records management activities, such as identity management, transaction processing, documenting provenance, food traceability[15] or voting.

The first blockchain was conceptualized in 2008 by an anonymous person or group known as Satoshi Nakamoto and implemented in 2009 as a core component of bitcoin where it serves as the public ledger for all transactions. The invention of the blockchain for bitcoin made it the first digital currency to solve the double spending problem without the need of a trusted authority or central server. The bitcoin design has been the inspiration for other applications.

Openness

Open blockchains are more user friendly than some traditional ownership records, which, while open to the public, still require physical access to view. Because all early blockchains were permissionless, controversy has arisen over the blockchain definition. An issue in this ongoing debate is whether a private system with verifiers tasked and authorized (permissioned) by a central authority should be considered a blockchain. Proponents of permissioned or private chains argue that the term "blockchain" may be applied to any data structure that batches data into time-stamped blocks. These blockchains serve as a distributed version of multiversion concurrency control (MVCC) in databases. Just as MVCC prevents two transactions from concurrently modifying a single object in a database, blockchains prevent two transactions from spending the same single output in a blockchain. Opponents say that permissioned systems resemble traditional corporate databases, not supporting decentralized data verification, and that such systems are not hardened against operator tampering and revision. Computerworld claims that "many in-house blockchain solutions will be nothing more than cumbersome databases." The Harvard Business Review defines blockchain as a distributed ledger or database open to anyone.

Permissionless

The great advantage to an open, permissionless, or public, blockchain network is that guarding against bad actors is not required and no access control is needed. This means that applications can be added to the network without the approval or trust of others, using the blockchain as a transport layer.

Bitcoin and other cryptocurrencies currently secure their blockchain by requiring new entries include a proof of work. To prolong the blockchain, bitcoin uses Hashcash puzzles developed by Adam Back in the 1990s.

Financial companies have not prioritised decentralized blockchains. In 2016, venture capital investment for blockchain related projects was weakening in the USA but increasing in China. Bitcoin and many other cryptocurrencies use open (public) blockchains.

As of November 2017, bitcoin has the highest market capitalization.

Permissioned (private) blockchain

Permissioned blockchains use an access control layer to govern who has access to the network. In contrast to public blockchain networks, validators on private blockchain networks are vetted by the network owner. They do not rely on anonymous nodes to validate transactions nor do they benefit from the network effect. Permissioned blockchains can also go by the name of 'consortium' or 'hybrid' blockchains.

The New York Times noted in both 2016 and 2017 that many corporations are using blockchain networks "with private blockchains, independent of the public system."

Disadvantages

Nikolai Hampton pointed out in Computer world that "There is also no need for a ‘51 percent’ attack on a private blockchain, as the private blockchain already controls 100 percent of all block creation resources. If you could attack or damage the blockchain creation tools on a private corporate server, you could effectively control 100 percent of their network and alter transactions however you wished." This has a set of particularly profound adverse implications during a financial crisis or debt crisis like the financial crisis of 2007–08, where politically powerful actors may make decisions that favor some groups at the expense of others. and "the bitcoin blockchain is protected by the massive group mining effort. It's unlikely that any private blockchain will try to protect records using gigawatts of computing power it's time consuming and expensive." He also said, "Within a private blockchain there is also no ‘race’; there's no incentive to use more power or discover blocks faster than competitors. This means that many in-house blockchain solutions will be nothing more than cumbersome databases."

G.LOKESHWAR
II B.Sc. (Information Technology)

DIGITAL TWIN

Digital twin refers to a digital replica of physical assets, processes and systems that can be used for various purposes. The digital representation provides both the elements and the dynamics of how an Internet of Things device operates and lives throughout its life cycle.

Digital twins integrate artificial intelligence, machine earning and software analytics with data to create living digital simulation models that update and change as their physical counterparts change. A digital twin continuously learns and updates itself from multiple sources to represent its near real-time status, working condition or position. This learning system, learns from itself, using sensor data that conveys various aspects of its operating condition from human experts, such as engineers with deep and relevant industry domain knowledge; from other similar machines; from other similar fleets of machines; and from the larger systems and environment in which it may be a part of. A digital twin also integrates historical data from past machine usage to factor into its digital model.

In various industrial sectors, twins are being used to optimize the operation and maintenance of physical assets, systems and manufacturing processes. It is a formative technology for the Industrial Internet of Things, where physical objects can live and interact with other machines and people virtually.

An example of how digital twins are used to optimize machines is with the maintenance of power generation equipment such as power generation turbines, jet engines and locomotives.

Another example of digital twins is the use of 3D modelling to create digital companions for the physical objects. It can be used to view the status of the actual physical object, which provides a way to project physical objects into the digital world. For example, when sensors collect data from a connected device, the sensor data can be used to update a "digital twin" copy of the device's state in real time the term "device shadow" is also used for the concept of a digital twin. The digital twin is meant to be an up-to-date and accurate copy of the physical object's properties and states, including shape, position, gesture, status and motion

A digital twin also can be used for monitoring, diagnostics and prognostics to optimize asset performance and utilization. In this field, sensory data can be combined with historical data, human expertise and fleet and simulation learning to improve the outcome of prognostics.

NIVEDHA N
I1I B.Sc. (Computer Technology)

PUZZLE

A solid, four-inch cube of wood is coated with blue paint on all six sides. Then the cube is cut into smaller one-inch cubes. These new one-inch cubes will have three blue sides, two blue sides, one blue side, or no blue sides. How many of each will be there ?

Puzzle Solution:

Subtract the outside squares, which will all have some paint on them

16 top

16 bottoms

8 more left

8 more right

4 more front

4 more back

= 56

So, only 8 will have no blue side.

The only cubes that will have blue on only one side will be the four center squares of each side of the 64-cube so, 4 x 6 sides = 24 cubes.

The only cubes that will have blue on three sides are the corner pieces there are 8 corners, so 8 cubes.

The cubes with two sides blue are the edge cubes between the corners two on each side of the top and bottom, so 2 x 4 sides x 2 (top and bottom) = 16, + the side edge/non-corner pieces, which will be another 2 x 4 = 8,So No blue side = 8, 1 side = 24 2 sides = 24 3 sides = 8, Total = 64 cubes.

NIVEDHA N

I1I B.Sc. (Computer Technology)

RIDDLES

1. What can travel around the world while staying in a corner?
   Ans: A stamp.

2. What has a head and a tail, but no body?
   Ans: Coin

3. What has an eye but cannot see?
   Ans: Needle.

4. There was a green house. Inside the green house there was a white house. Inside the white house there was a red house. Inside the red house there were lots of babies. What is it?
   Ans : Watermelon.

5. What kind of room has no doors or windows?
   Ans: Mushroom.

6. Which creature walks with four legs in the morning, two legs in the afternoon, and three legs in the evening?
   Ans: Man. He crawls on all fours as a baby, then walks on two feet as an adult, and then walks with a cane as an old man.

7. If you have me, you want to share me. If you share me, you haven't got me. What am I?
   Ans: Secret.

8. Forward I am heavy, but backward I am not. What am I?
   Ans: Forward I am ton, backwards I am not.

9. What is as light as a feather, but even the world's strongest man couldn't hold it for more than a minute?

   Ans: His breath.

10. What is always coming but never arrives?

    Ans: Tomorrow.

SINDUJA T

I B.Sc. (Computer Technology)

SUCCESS IS WHEN YOUR SIGNATURE CHANGES TO AUTOGRAPH

-APJ ABDUL KALAM