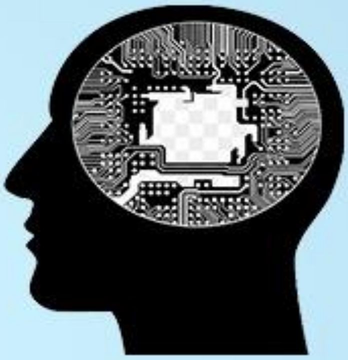
INFOLINE

Volume XIII Issue 3 January 2023



DEPARTMENT OF COMPUTER TECHNOLOGY AND INFORMATION TECHNOLOGY

KONGU ARTS AND SCIENCE COLLEGE
(AUTONOMOUS)

Affiliated to Bharathiar University,
Coimbatore
Approved by UGC and AICTE,New Delhi
Re-accredited by NAAC,DBT STAR College Scheme
An ISO 9001 : 2015 Certified Institution
Nanjanapuram,Erode - 638 107

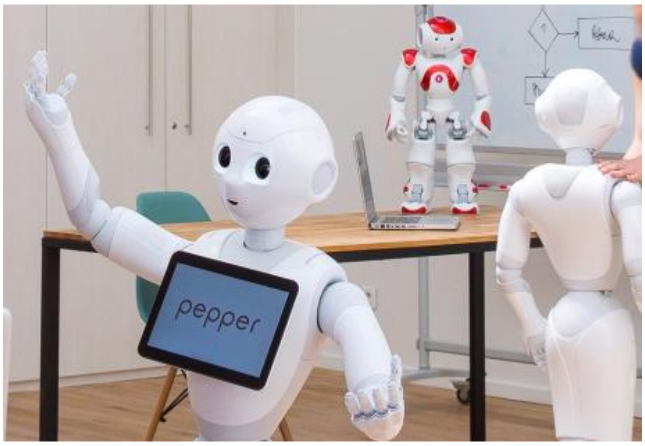
INFOLINE

EDITORIAL BOARD

CONTENTS

PEPPER: A TALKING HUMANOID ROBOT

Pepper is the world's first social humanoid robot able to recognize faces and basic human emotions. Pepper was optimized for human interaction and is able to engage with people through conversation and his touch screen. Pepper is available today for businesses and schools. Over 2,000 companies around the world have adopted Pepper as an assistant to welcome, inform and guide visitors in an innovative way.



Pepper is a semi-humanoid robot manufactured by SoftBank Robotics (formerly Aldebaran Robotics), designed with the ability to read emotions. It was introduced in a conference on 5th June 2014 and was showcased in SoftBank mobile phone stores in Japan. Pepper's ability to recognize emotion is based on detection and analysis of facial expressions and voice tones. Production of Pepper was paused in June 2021, due to weak demand.

Applications

Commercial

Pepper is currently being used as a receptionist at several offices in the UK and is able to identify visitors with the use of facial recognition, send alerts for meeting organisers and arrange for drinks to be made. Pepper is able to chat autonomously to prospective clients. The first functioning of Pepper as receptionist in the UK was supplied by a SoftBank distributor and was installed in London at Brainlabs. The robot has also been used in banks and for medical facilities in Japan, using applications created by Seikatsu Kakumei and it is also used in all branches of Hamazushi restaurants in Japan.

Pepper is being used in North American airports such as Pierre Elliott Trudeau International Airport in Montreal, Canada. The robot is used to greet travellers, offer menus and recommendations. In 2018, Pepper robot was introduced first time in the UAE. In December 2019, a dozen of Pepper robots were installed at the "Pepper Parlor Café" in Tokyo, Japan.

Sports

On 9th July 2020, a team of Pepper robots performed as cheerleaders at a baseball game between the Fukuoka SoftBank Hawks and the Rakuten Eagles, supported by a team of Boston Dynamics Spot quadrupedal robots.

Academic

Pepper is available as a research and educational robot for schools, colleges and universities to teach programming and conduct research into human-robot interactions. In 2017, an international team began research into using Pepper as versatile robot to help look after older people in care homes or sheltered accommodation. The project CARESSES aimed at developing the world's first culturally-competent robot, received funding worth more than two million Euros, with donors including the European Union and the Japanese government. The project was expected to run for three years. Institutions involved in the research include University of Genoa (Project Coordinator), Örebro University, Middlesex University, the University of Bedfordshire, SoftBank Robotics, Advinia HealthCare, Japan, Advanced Institute of Science and Technology (Japanese coordinator), Nagoya University, Chubu University. On Tuesday 16 October 2018, a Pepper robot mentioned the CARESSES project while giving evidence to the Education Committee of the House of Commons of the United Kingdom Parliament.

Long-term research with Pepper could show that residents of care home are willing to interact with humanoid robots and benefit from cognitive and physical activation that is led by the robot Pepper. Another long-term study in a care home could show that people working in the care sector are willing to use robots in their daily work with the residents. But it also revealed that even though that the robots are ready to be used, they do need human assistants, they cannot replace the human work force but they can assist them and give them new possibilities.

Specifications

The robot's head has four microphones, two HD cameras (in the mouth and forehead), and a 3-D depth sensor (behind the eyes). There is a gyroscope in the torso and touch sensors in the head and hands. The mobile base has two sonars, six lasers, three bumper sensors, and a gyroscope. It is able to run the existing content in the app store designed for SoftBank's Nao robot.

B.THARNIKA

III B.Sc. (Information Technology)

AI-BASED CYBERSECURITY

According to Ed Bowen, Managing Director of Deloitte and AI Advisory Leader, cybersecurity threats will be more effectively detected and responded through AI and machine learning techniques.

As a result of improved detection efficiency, improved agility and increased resiliency in the face of disruption, Bowen said that AI-supported cyber programs could better manage multi-faceted, dynamic risks. He added that organizations that do not integrate AI are likely to fall behind the curve in security and experience a higher rate of negative impacts.

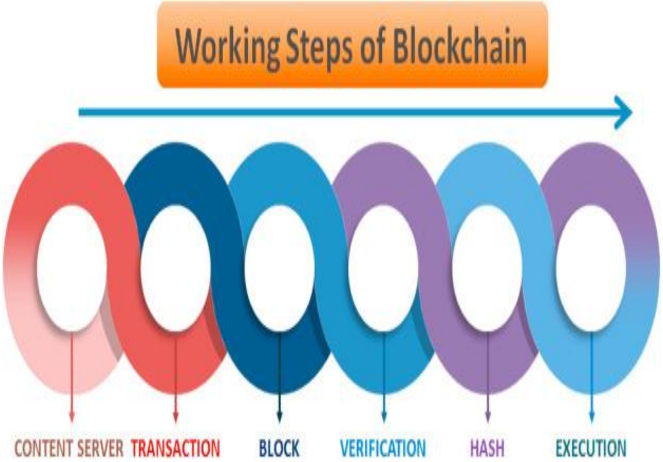
As cyberattacks grow in volume and complexity, artificial intelligence (AI) is helping under-resourced security operations analysts stay ahead of threats. Curating threat intelligence from millions of research papers, blogs and news stories, AI technologies like machine learning and natural language processing provide rapid insights to cut through the noise of daily alerts, drastically reducing response times.

M. HARINI

I B.Sc. (Computer Technology)

BLOCKCHAIN TECHNOLOGY APPLICATIONS IN HEALTHCARE

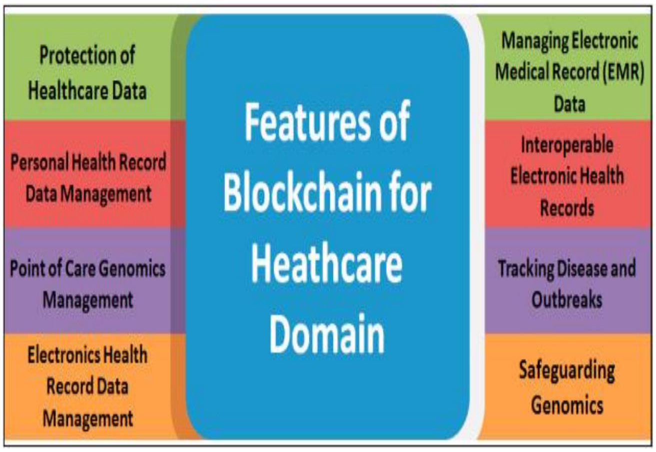
Blockchain is a decentralised node network that stores the data. It is an excellent technology for protecting confidential data within the system. This technology helps to exchange critical data and keeps it secure and confidential. It is a perfect tool to hold all the related documents in one location and securely. Blockchain also speeds up searches for applicants that fulfil specific trial criteria using a single patient database. The Blockchain can be described as a decentralised peer-to-peer (P2P) network of personal computers called nodes which maintains, stores and records historical or transaction data. It allows a reliable collaboration as the information is stored and exchanged by all network members and keeps a constant track of past and current experiences. This technology can integrate disparate networks to provide insights into the importance of individual treatment. Thus, Blockchain can well be recognised for immutability and safety. Blocks, nodes and miners are the three main ideas in Blockchain. Blockchain does not save any of its data in a single location. Instead, a network of computers copies and spreads the Blockchain. Every computer on the web updates its Blockchain to reflect a new block to the Blockchain.



A Blockchain system runs on top of the internet on a P2P network of computers that all run the protocol and have an identical copy of the transaction ledger, allowing for P2P value transactions without using an intermediary by machine consensus. There are various types of Blockchain technologies such as public, private, hybrid or consortium. Each Blockchain network has different advantages and disadvantages that essentially influence its optimal applications.

Blockchain Technology in Healthcare

In healthcare, Blockchain has a wide range of applications and functions. The ledger technology helps healthcare researchers uncover genetic code by facilitating the secure transfer of patient medical records, managing the drug supply chain and facilitating the safe transfer of patient medical records. The below figure reflects the variety of features and critical enablers of Blockchain philosophy in umpteen healthcare spheres and its allied domains. Protection of healthcare data, various genomics management, electronic data management, medical records, interoperability, digitalised tracking and issues outbreak etc.., are some of the technically derived and impressive features employed to develop and practice Blockchain technology. The complete digitalised aspects of Blockchain technology and its use in healthcare-related applications are the significant reasons for its adoption.


Features of Blockchain for Heathcare Domain
Protection of Healthcare Data
Personal Health Record Data Management
Point of Care Genomics Management
Electronics Health Record Data Management
Managing Electronic Medical Record (EMR) Data
Interoperable Electronic Health Records
Tracking Disease and Outbreaks
Safeguarding Genomics

The Blockchain makes the entire prescription process transparent from manufacturing to pharmacy shelves. Congestion, freight direction and speed may all be tracked using IoT and Blockchain. It offers the chance to schedule acquisitions efficiently to prevent disruptions and shortages in clinics, pharmacies, and other medical facilities with a given medication. The deployment of digital frameworks built on Blockchain would help ensure that the logistics data avoid uncontrolled adjustments. It increases trust and prevents the illicit handling of records, payments and medication themselves by various people interested in purchasing drugs. The technology can effectively improve the condition of patients while at a competitive cost retaining the funds. It eliminates all obstacles and barriers in multi-level authentication. Because Blockchain can preserve an incorruptible, decentralised and transparent log of all patient data, it is ripe for security applications. Furthermore, while Blockchain is visible, it is also private, hiding any individual's identity behind complicated and secure algorithms that can preserve the sensitivity of medical data.

Thanks to the technology's decentralised structure, patients, doctors and healthcare providers can all share the same information swiftly and safely.

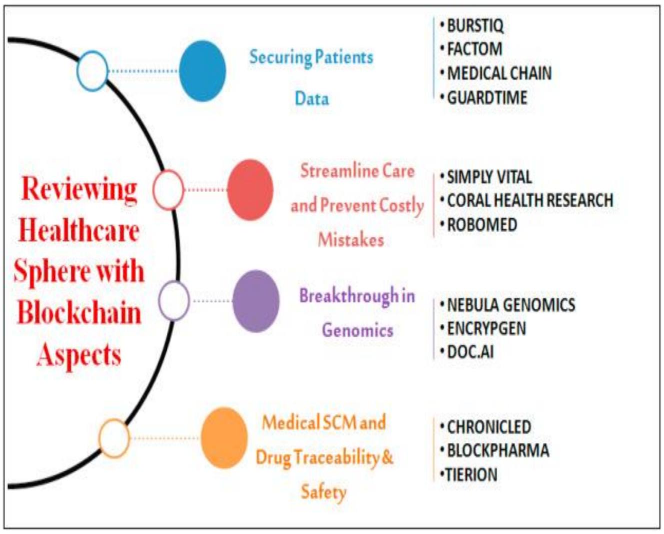
Blockchain technology makes the transition to interoperability led by patients easier as it allows patients to make their medical data accessible and access laws. This gives a patient greater power over personal information and improves confidentiality and privacy. The measurement and implementation of quality management and enforcement are difficult. Any of these technical issues could be solved by Blockchain applications throughout the industry. Blockchain headlines will assist regulatory authorities in tracing legal drugs against falsified ones. This ensures that all approved parties exchange digital transactions containing the patient's details. Patients who move medical practitioners may merely update a single consent to exchange their complete records.

Blockchain has reached the healthcare industry with a rising acceptance rate. Also, in the early stages, people in the health ecosystem accept the technologies favourably. In the coming years, the holistic vision of Blockchain to transform the healthcare market will be to address problems affecting the present structure. It allows physicians, patients and pharmacists to conveniently access all the available information at a given moment. Medical firms are exploring, experimenting, discovering day and night for Blockchain technologies, in the medical field for health records. It has confirmed itself as an irreplaceable instrument in healthcare by following pharmaceuticals, improving payment options and decentralising patient health history records. In addition to robust technology such as machine learning & artificial intelligence, the medical sector is highly dependent on Blockchain. There are some genuine uses of how Blockchain changes the healthcare industry. The program is built on Blockchain monitoring technology to fine-tune the medical supply chain.

Blockchain's ability makes for a sophisticated data storage framework that records a person's whole health history of diagnosis, test reports, prior regimes and even measurements by intelligent sensors. A doctor can conveniently obtain all the details available to make correct diagnoses and recommendations using this method. Because all the data in a single Blockchain system is stored, it is safe from loss and shift. To avoid an organisation's internal networks, it can use Blockchain. A significant organisation of many separate players with various control levels on an encrypted Blockchain database can save organisations from external risks and assaults. If a healthcare organisation correctly implements a Blockchain network, this will eliminate such rescue attacks and other problems such as computer corruption or hardware failure.

Blockchain Technology for reviving healthcare services

The figure illustrates the several on-ground industrial representatives of Blockchain capabilities to successfully implement healthcare culture perspectives and overall development. There have been various associated industrial/medical-care supporters or providers which helps carry out the research and investigations for realising the Blockchain practices in healthcare and its core domains, too. These observed providers BurstIQ, Guardtime, Robomed, Simply vital, Encrypgen, Chronicled, Tieion, etc., are the few agencies supplying and favouring the practising of Blockchain technology at ground levels.



Blockchain involves the development of new patient data cards for medical practitioners in other hospitals. Newly added information is usually repetitive and leads to loss of time which is a severe health malfunction. Each individual can have different privileges or accessibility choices due to the location on the supply chain. In addition, any block containing the medication information would have a hash connected to it to another block. Furthermore, the data transparency feature in the Blockchain framework will help find the entire root route and eradicate the distribution of fabricated drugs. A new medical card is established for the patient and deposited in a specific facility when the patient visits a new clinic. These details are ordinarily incomplete to the general public and contain collected records from the caregivers at the facilities concerned. Blockchain can conveniently assemble problems in the data processing. This technology will create open and similar Blockchain medical records around the world.

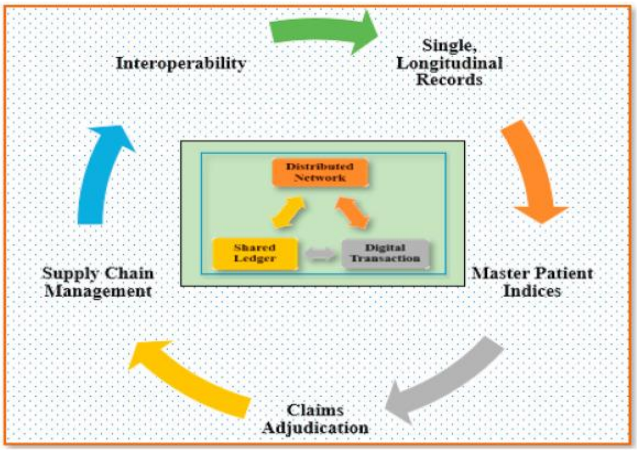
In healthcare, clinical trials are being conducted to assess the effectiveness of such therapies to treat or provide a partial remedy to a particular disease. Scientists can record data on test outcomes, person numbers, patient records and other variables. Data collected during clinical trials should be authenticated so that scientists, pharmaceutical firms and policymakers can be confident in the quality of results. In clinical trials, Blockchain technology could provide greater transparency and accountability. The health care Blockchain has enormous record-keeping leverage, as the blocks are made available to clinicians and patients while the processing of medical history is done with an awareness of patient issues. Blockchain in the supply chain is very popular, and it also fits well for medicines in the healthcare area. Blockchain provides easy forward-thinking on the practices and services of health professionals. In the health care industry, this Blockchain power operates to handle approval efficiently to process and acquire. It is convenient to avoid wastage by standing in a line to improve productivity and spruce up workflow with Blockchain. This technology aims to encourage personalised medicine, clinical advice and practical research into health. Blockchain has become one of today's most popular technologies. The newest thing about the Blockchain is that the company believes that the platform can turn the healthcare industry more effectively. This will, in many ways, turn the healthcare industry into a reliable and stable digital directory. Blockchain healthcare technologies will strengthen various challenges such as clinical studies, patient records management and prescription traceability.

Blockchains can be used successfully in healthcare to make the right choices in the health ecosystem. The distributed Blockchain platform offers the health sector opportunities to trace fraud, reduce overhead costs, reliably manufacture jobs, eliminate duplication of labour, enforce openness in the health environment. Further, it is used to hold assets such as immutability and confidence and decentralisation. The clinical trials and the approval organisations for the subject are the areas where Blockchain has the opportunity to boost medical professionals' and researchers' reliability, auditability, and accountability. The benefit for patients is that their medical histories are protected more confidently and that their diagnostic accuracy improves chances of further care.

In the processing of their health records, patients will now have a better voice. It will be allowed to exchange data as partners of the Blockchain network, thereby ensuring further privacy and control. Blockchain's pledge has widespread consequences for health care stakeholders. Disparate networks are likely to interact to provide insights and help evaluate treatment's importance based on this technology. An electronic medical records Blockchain network around the country will increase efficiencies and promote improved patient health results in the long run. In particular, Blockchain is a mutual, unchanging record of transactions made from connected transaction blocks and held in a digital booklet. Medical details like patient life, medical equipment logs, or medical products' temperatures can be recorded during the shipping, following the medical field.

Unified Work-Flow Process of Blockchain Technology realization in healthcare amenities

The below figure explores the schematic reflection about the integrated workflow for the overall development of Blockchain technology and its associated benefits/merits from a healthcare services perspective.



The master patient indices claim adjustments, devoted medical supply chain management, interoperability, single and longitudinal records capturing etc., are related benefits associated with the Blockchain practices in healthcare sectors. The interactive work-process flow started with the distributed network flow, digitalised transactions and shared data and ledger which ultimately enables the Blockchain drivers to work more emphasised to make healthcare services healthier and more innovative than before.

P.S. MOHANKUMAR
II B.Sc. (Computer Technology)

REAL-TIME COLLISION AVOIDANCE IN DYNAMIC ENVIRONMENTS

The problem of collision avoidance has existed in robotics since the very beginning. It occurs in industrial manipulators, ground, aquatic and flying robots and refers to the ability of such autonomous agents to detect and respond to obstacles in their environment to avoid damage. It typically remains in opposition or detriment, to the main task performed by the robot which is the intended outcome or objective of the robot's actions, such as reaching a certain location, completing a specific task or moving an object from one place to another. The goal is often defined by the user or the system that controls the robot.

In recent years, robots are used not only in industrial settings, no-go zones and safety cages. Service robotics and cobots, robots that are designed to collaborate and interact with a human being in close proximity, have been developed. In this new scenario, real-time collision avoidance in dynamic environments has become even more critical. Robots now have the potential to cause harm not only to themselves, other robots or the task space elements but also to humans and other living beings. Research and development in these areas will continue and robots will increasingly appear both in production processes and in our everyday life.

The 'Real-time collision avoidance in dynamic environments' topic is dedicated to all aspects of real-time detecting of obstacles and taking appropriate, safe and efficient actions to avoid collisions in dynamic environments. Our aim is to receive contributions in a variety of subjects including but not limited to the following:

-traffic collision avoidance
- platooning
- collision avoidance in multi-agent systems
- pedestrian safety in the presence of a robot

- avoiding aircraft collisions
- safety in manipulator applications
- obstacle detection and modeling
- avoiding collisions in an unstructured environment
- collision avoidance based on artificial potential functions and navigation functions
- obstacle detection data fusion
- modern sensors in collision avoidance
- collision avoidance in unmanned aerial vehicles
- collision avoidance in service robots and cobots

G.AAKASH

II B.Sc. (Information Technology)

THE NEEDS OF MODERNIZED SOC FOR HYBRID CLOUD

Microsoft's Azure-hosted OpenAI language models are now generally available and it's surprisingly simple to use them in the code.

Modern machine learning and AI research have moved rapidly from the lab to our IDEs with tools like Azure's Cognitive Services providing API-based access to pretrained models. There are many different approaches to delivering AI services with one of the more promising methods for working with language being a technique called generative pretraining or GPT which handles large amounts of text.

OpenAI and Microsoft

The OpenAI research lab pioneered this technique, publishing the initial paper on the topic in 2018. The model it uses has been through several iterations, starting with the unsupervised GPT-2 which used untagged data to mimic humans. Built on top of 40GB of public internet content, GPT-2 required significant training to provide a model with 1.5 billion parameters. It was followed by GPT-3, a much larger model with 175 billion parameters. Exclusively licensed to Microsoft, GPT-3 is the basis for tools like the programming code-focused Codex used by GitHub Copilot and the image-generating DALL-E.

With a model like GPT-3 requiring significant amounts of compute and memory, on the order of thousands of petaflop/s-days, it's an ideal candidate for cloud-based high-performance computing on specialized supercomputer hardware. Microsoft has built its own Nvidia-based servers for supercomputing on Azure with its cloud instances appearing on the TOP500 supercomputing list. Azure's AI servers are built around Nvidia Ampere A12000 Tensor Core GPUs, interconnected via a high-speed InfiniBand network.

Adding OpenAI to Azure

OpenAI's generative AI tools have been built and trained on the Azure servers. As part of a long-running deal between OpenAI and Microsoft, OpenAI's tools are being made available as part of Azure with Azure-specific APIs and integration with Azure's billing services. After some time in private preview, the Azure OpenAI suite of APIs is now generally available with support for GPT-3 text generation and the Codex code model. Microsoft has said it will add DALL-E image generation in a future update.

That doesn't mean that anyone can build an app that uses GPT-3; Microsoft is still gating access to ensure that projects comply with its ethical AI usage policies and are tightly scoped to specific use cases. You also need to be a direct Microsoft customer to get access to Azure OpenAI. Microsoft uses a similar process for access to its Limited Access Cognitive Services where there's a possibility of impersonation or privacy violations. Those policies are likely to remain strict and some areas such as health services will probably require extra protection to meet regulatory requirements. Microsoft's own experiences with AI language models have taught it a lesson it doesn't want to repeat. As an added protection, there are content filters on inputs and outputs with alerts for both Microsoft and developers.

Exploring Azure OpenAI Studio

Once your account has been approved to use Azure OpenAI, you can start to build code that uses its API endpoints. The appropriate Azure resources can be created from the portal, the Azure CLI, or Arm templates. If you're using the Azure Portal, create a resource that's allocated to your account and the resource group you intend to use for your app and any associated Azure services and infrastructure. Next, name the resource and select the pricing tier. At the moment, there's only one pricing option, but this will likely change as Microsoft rolls out new service tiers. With a resource in place you can now deploy a model using Azure OpenAI Studio. This is where you'll do most of your work with OpenAI. Currently, you can choose between members of the GPT-3 family of models including the code-based Codex. Additional models use embeddings, complex semantic information that is optimized for search.

Within each family, there is a set of different models with names that indicate both cost and capability. If you're using GPT-3, Ada is the lowest cost and least capable and Davinci is the highest. Each model is a superset of the previous one, so as tasks get more complex, you don’t need to change your code, you simply choose a different model. Interestingly, Microsoft recommends starting with the most capable model when designing an OpenAI-powered application, as this lets you tune the underlying model for price and performance when you go into production.

Working with model customization

Although GPT-3’s text completion features have gone viral, in practice your application will need to be much more focused on your specific use case. You don’t want GPT-3 to power a support service that regularly gives irrelevant advice. You must build a custom model using training examples with inputs and desired outputs which Azure OpenAI calls “completions.” It’s important to have a large set of training data and Microsoft recommends using several hundred examples. You can include all your prompts and completions in one JSON file to simplify managing your training data.

With a customized model in place, you can use Azure OpenAI Studio to test how GPT-3 will work for your scenario. A basic playground lets you see how the model responds to specific prompts with a basic console app that lets you type in a prompt and it returns an OpenAI completion. Microsoft describes building a good prompt as “show, don’t tell,” suggesting that prompts need to be as explicit as possible to get the best output. The playground also helps train your model, so if you’re building a classifier, you can provide a list of text and expected outputs before delivering inputs and a trigger to get a response.

One useful feature of the playground is the ability to set an intent and expected behaviors early, so if you’re using OpenAI to power a help desk triage tool, you can set the expectation that the output will be polite and calm, ensuring it won’t mimic an angry user. The same tools can be used with the Codex model, so you can see how it works as a tool for code completion or as a dynamic assistant.

Writing code to work with Azure OpenAI

Once you’re ready to start coding, you can use your deployment’s REST endpoints, either directly or with the OpenAI Python libraries. The latter is probably your quickest route to live code. You’ll need the endpoint URL, an authentication key and the name of your deployment. Once you have these, set the appropriate environment variables for your code. As always, in production it’s best not to hard-code keys and to use a tool like Azure Key Vault to manage them.

Calling an endpoint is easy enough: Simply use the OpenAI completion create method to get a response, setting the maximum number of tokens needed to contain your prompt and its response. The response object returned by the API contains the text generated by your model which can be extracted, formatted and then used by the rest of your code. The basic calls are simple and there are additional parameters your code can use to manage the response. These control the model's creativity and how it samples its results. You can use these parameters to ensure responses are straightforward and accurate.

If you're using another language, use its REST and JSON parsing tools. You can find an API reference in the Azure OpenAI documentation or take advantage of Azure's GitHub-hosted Swagger specifications to generate API calls and work with the returned data. This approach works well with IDEs like Visual Studio.

Azure OpenAI pricing

One key element of OpenAI models is their token-based pricing model. Tokens in Azure OpenAI aren't the familiar authentication token; they're tokenized sections of strings which are created using an internal statistical model. Open AI provides a tool on its site to show how strings are tokenized to help you understand how your queries are billed. You can expect a token to be roughly four characters of text, though it can be less or more; however, it should end up with 75 words needing about 100 tokens (roughly a paragraph of normal text).

The more complex the model, the higher priced the tokens. Base model Ada comes in at about $0.0004 per 1,000 tokens, and the high-end Davinci is $0.02. If you apply your own tuning, there's a storage cost and if you're using embeddings, costs can be an order of magnitude higher due to increased compute requirements. There are additional costs for fine-tuning models, starting at $20 per compute hour. The Azure website has sample prices, but actual pricing can vary depending on your organization's account relationship with Microsoft. Perhaps the most surprising thing about Azure OpenAIclo is how simple it is. As you're using prebuilt models (with the option of some fine tuning), all you need to do is apply some basic pretraining, understand how prompts generate output and link the tools to your code, generating text content or code as and when it's needed.

DINESH S
I B.Sc. (Computer Technology)

THE BEST PRACTICES FOR EVERY MONGODB DEPLOYMENT

MongoDB is a non-relational document database that provides support for JSON-like storage. Its flexible data model allows you to easily store unstructured data. First released in 2009, it is the most commonly used NoSQL database. It has been downloaded more than 325 million times. MongoDB is popular with developers because it is easy to get started with. Over the years, MongoDB has introduced many features that have turned the database into a robust solution able to store terabytes of data for applications.

As with any database, developers and DBAs working with MongoDB should look at how to optimize the performance of their database, especially nowadays with cloud services where each byte processed, transmitted and stored costs money. The ability to get started so quickly with MongoDB means that it is easy to overlook potential problems or miss out on simple performance improvements.

practice #1: Enable authorization and authentication on your database right from the start.

practice #2: Don't use 'not recommended versions' or 'end-of-life versions' in production instances and stay updated.

practice #3: Use MongoDB replication to ensure HA and check the status of your replica often.

practice #4: Use $regex queries only when necessary and choose text search instead where you can.

practice #5: Think wisely about your index strategy.

practice #6: Check your queries and indexes frequently.

practice #7: Don't run multiple mongodb or mongoose instances on the same server

practice #8: Back up frequently.

practice #9: Know when to share your replica set and choose a shared key carefully.

S.JAGADESH
III B.Sc. (Information Technology)

SkySQL-MariaDB ADDS SERVERLESS ANALYTICS AND COST MANAGEMENT

The new release of the managed database as a service removes the need for extracting, transforming and loading data by adding a 'serverless' layer powered by Apache Spark SQL.

MariaDB is adding features such as serverless analytics and cost management to the new release of its managed database-as-a-service (DBaaS) SkySQL. SkySQL which is a managed instance of the MariaDB platform, offers OLAP (online analytical processing) and OLTP (online transaction processing) along with enterprise features like sharding, load balancing and auto-failover via a combination of MariaDBXpand, In order to help enterprises bring down the cost of databases and to manage expenditure better, MariaDB has introduced an autoscaling feature for both compute and storage. "Rules specify when autoscaling is triggered, for example, when CPU utilization is above 75% over all replicas sustained for 30 minutes, then a new replica or node will be added to handle the increase," the company said in a statement.

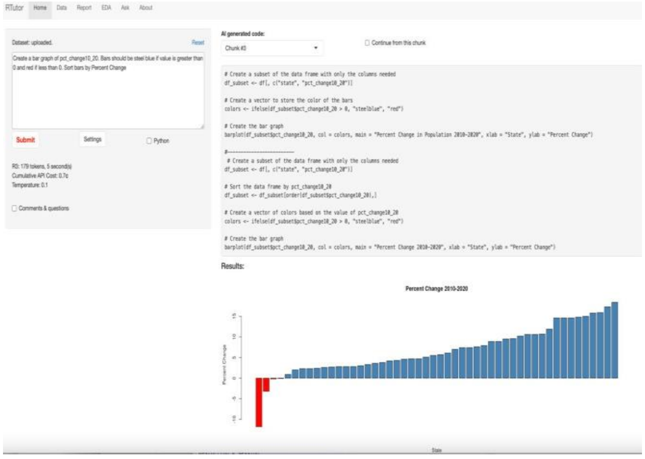
K. DURAIMURUGAMI
B.Sc. (Computer Technology)

C

ChatGPT TOOLS FOR R PROGRAMMING

RTutor

This app is an elegant and easy way to sample ChatGPT and R. Upload a data set, ask a question and watch as it generates R code and your results including graphics. Although it's named RTutor, the app can also generate Python code. RTutor is on the web at https://rtutor.ai/. It's currently the only app or package listed that doesn't require a ChatGPT API key to use, but you're asked to supply your own for heavy use so as not to bill the creators' account.



Dataset: uploaded.

Create a bar graph of pct_change10_20. Bars should be steel blue if value is greater than 0 and red if less than 0. Sort bars by Percent Change

□ Python

R3: 179 tokens, 5 second(s)
Cumulative API Cost: 0.7¢
Temperature: 0.1

□ Comments & questions

AI generated code:

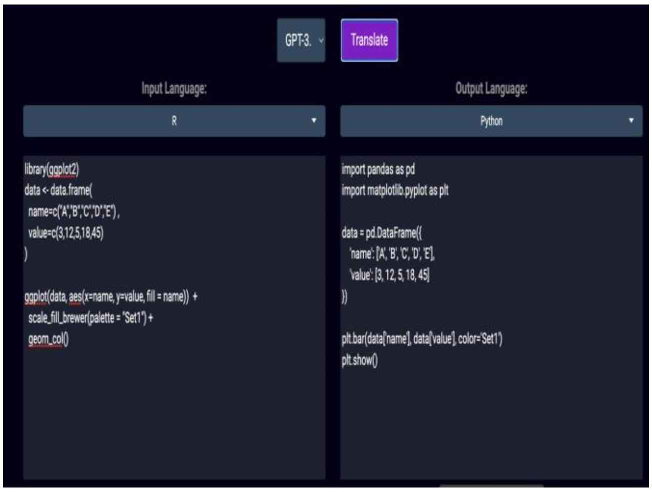
Chunk #3

□ Continue from this chunk

```
# Create a subset of the data frame with only the columns needed
df_subset <- df[, c("state", "pct_change10_20")]

# Create a vector to store the color of the bars
colors <- ifelse(df_subset$pct_change10_20 > 0, "steelblue", "red")

# Create the bar graph
barplot(df_subset$pct_change10_20, col = colors, main = "Percent Change in Population 2010-2020", xlab = "State", ylab = "Percent Change")

#-----------------------
 # Create a subset of the data frame with only the columns needed
df_subset <- df[, c("state", "pct_change10_20")]

# Sort the data frame by pct_change10_20
df_subset <- df_subset[order(df_subset$pct_change10_20),]

# Create a vector of colors based on the value of pct_change10_20
colors <- ifelse(df_subset$pct_change10_20 > 0, "steelblue", "red")

# Create the bar graph
barplot(df_subset$pct_change10_20, col = colors, main = "Percent Change 2010-2020", xlab = "State", ylab = "Percent Change")
```

CodeLingo

This multi-language app "translates" code from one programming language to another. Available languages include Java, Python, JavaScript, C, C++, PHP and more, including R.



Input Language: R

```
library(ggplot2)
data <- data.frame(
  name=c("A","B","C","D","E"),
  value=c(3,12,5,18,45)
)

ggplot(data, aes(x=name, y=value, fill = name)) +
  scale_fill_brewer(palette = "Set1") +
  geom_col()
```

Output Language: Python

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.DataFrame({
   'name': ['A', 'B', 'C', 'D', 'E'],
   'value': [3, 12, 5, 18, 45]
})

plt.bar(data['name'], data['value'], color='Set1')
plt.show()
```

ChatGPT in the CodeLingo app attempts to translate ggplot2 graph code to Python. A request to translate code for a ggplot2 R graph into JavaScript generated output using the rather hard-to-learn D3 JavaScript library, as opposed to something a JavaScript newbie would be more likely to want such as Observable Plot or Vega-Lite.

askgpt

This package, available at https://github.com/JBGruber/askgpt, can be a good starting point for first-time users who want ChatGPT in their console, in part because it gives some instructions upon initial startup. Load the package with library (askgpt) and it responds with:

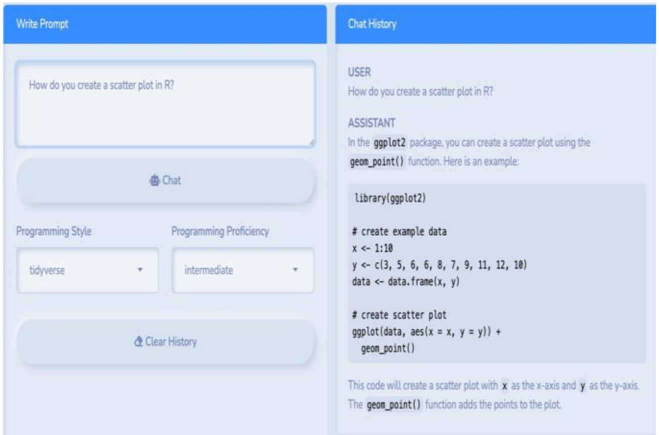
```
Hi,thisisaskgpt☺.

•To start error logging, run
`log_init()` now.

•To see what you can
douse`?askgpt()`.

•Or just run `askgpt()`with any
question you want!
```

```
iIt looks like you have not
provided an API key yet.

1.Go to
<https://platform.openai.com/ac
count/api-keys>

2.(Loginto your account if you
haven't done so yet)

3. On the site, click the
button + Create new secret key
to create an API key

4. Copy this key into R/RStudio
```

gptstudio

This package and its sibling, gpttools (discussed below), feature RStudio add-ins to work with ChatGPT, although there are also some command-line functions that will work in any IDE or terminal. You can access add-ins within RStudio either from the add-in drop-down menu above the code source pane or by searching for them via the RStudio command palette (Ctrl-shift-p). According to the package website, gptstudio is a general-purpose helper "for R programmers to easily incorporate use of large language models (LLMs) into their project workflows." It is on CRAN.

One add-in, ChatGPT, launches a browser-based app for asking your R coding questions and offers options for programming style (tidyverse, base, or no preference) and proficiency (beginner, intermediate, advanced, and genius).



How do you create a scatter plot in R?

Programming Style: tidyverse

Programming Proficiency: intermediate

USER

How do you create a scatter plot in R?

ASSISTANT

In the `ggplot2` package, you can create a scatter plot using the `geom_point()` function. Here is an example:

```
library(ggplot2)

# create example data
x <- 1:10
y <- c(3, 5, 6, 6, 8, 7, 9, 11, 12, 10)
data <- data.frame(x, y)

# create scatter plot
ggplot(data, aes(x = x, y = y)) +
  geom_point()
```

This code will create a scatter plot with `x` as the x-axis and `y` as the y-axis. The `geom_point()` function adds the points to the plot.

S.M HARI VISHNUII

B.Sc. (Information Technology)

PYTHON 3.12: FASTER, LEANER, MORE FUTURE-PROOF

Improvements to the next (and future) versions of Python are set to speed it up, slim it down, and pave the way toward even better things.

Python is a dynamic language, making it faster has been a challenge. But over the last couple of years, developers in the core Python team have focused on various ways to do it. At PyCon 2023, held in Salt Lake City, Utah, several talks highlighted Python's future as a faster and more efficient language. Python 3.12 will showcase many of those improvements. Some are new in that latest version, others are already in Python but have been further refined.

Mark Shannon, a longtime core Python contributor now at Microsoft, summarized many of the initiatives to speed up and streamline Python. Most of the work he described in his presentation centered on reducing Python's memory use, making the interpreter faster and optimizing the compiler to yield more efficient code.

The perinterpreter GIL and subinterpreters

What keeps Python from being truly fast? One of the most common answers is "lack of a better way to execute code across multiple cores." Python does have multithreading but threads run cooperatively, yielding to each other for CPU-bound work. And Python's support for multiprocessing is top-heavy: you have to spin up multiple copies of the Python runtime for each core and distribute your work between them. One long-dreamed way to solve this problem is to remove Python's GIL, or Global Interpreter Lock. The GIL synchronizes operations between threads to ensure objects are accessed by only one thread at a time. In theory, removing the GIL would allow true multithreading. In practice and it's been tried many times it slows down non-threaded use cases, so it's not a net win.

Core python developer Eric Snow, in his talk, unveiled a possible future solution for all this: subinterpreters, and *a* per-interpreter GIL. In short: the GIL wouldn't be removed, just sidestepped. Subinterpreters is a mechanism where the Python runtime can have multiple interpreters running together inside a single process, as opposed to each interpreter being isolated in its own process (the current multiprocessing mechanism). Each subinterpreter gets its own GIL, but all subinterpreters can share state more readily. While subinterpreters have been available in the Python runtime for some time now, they haven't had an interface for the end user. Also, the messy state of Python's internals hasn't allowed subinterperters to be used effectively.

With Python 3.12, Snow and his cohort cleaned up Python's internals enough to make subinterpreters useful and they are adding a minimal module to the Python standard library called interpreters. This gives programmers a rudimentary way to launch subinterpreters and execute code on them. Snow's own initial experiments with subinterpreters significantly outperformed threading and multiprocessing. One example, a simple web service that performed some CPU-bound work, maxed out at 100 requests per second with threads, and 600 with multiprocessing. But with subinterpreters, it yielded 11,500 requests, and with little to no drop-off when scaled up from one client. The interpreters module has very limited functionality right now and it lacks robust mechanisms for sharing state between subinterpreters. But Snow believes by Python 3.13 a good deal more functionality will appear and in the interim developers are encouraged to experiment.

A faster Python interpreter

Another major set of performance improvements Shannon mentioned, Python's new adaptive specializing interpreter was discussed in detail in a separate session by core Python developer Brandt Bucher. Python 3.11 introduced new bytecodes to the interpreter, called adaptive instructions. These instructions can be replaced automatically at runtime with versions specialized for a given Python type, a process called quickening. This saves the interpreter the step of having to look up what types the objects are, speeding up the whole process enormously. For instance, if a given addition operation regularly takes in two integers, that instruction can be replaced with one that assumes the operands are both integers.

Not all code specializes well, though. For instance, arithmetic between ints and floats is allowed in Python, but operations between ints and ints or floats and ints, don't specialize well. Bucher provides a tool called specialist, available on PyPI, to determine if code will specialize well or badly, and to suggest where it can be improved. Python 3.12 has more adaptive specialization opcodes, such as accessors for dynamic attributes which are slow operations. Version 3.12 also simplifies the overall process of specializing with fewer steps involved.

Future-proofing Python's internals

The default Python implementation, CPython, has three decades of development behind it. That also means three decades of cruft, legacy APIs and design decisions that can be hard to transcend all of which make it hard to improve Python in key ways. Core Python Developer Victor Stinner, in a presentation about how Python features are deprecated over time, touched on some of the ways Python's internals are being cleaned up and future-proofed. One key issue is the proliferation of C APIs found in CPython, the reference runtime for the language. As of Python 3.8, there are a few different sets of APIs, each with different maintenance requirements. Over the last five years, Stinner worked to make many public APIs private, so programmers don't need to deal as directly with sensitive CPython internals. The long-term goal is to make components that use the C APIs, like Python extension modules, less dependent on things that might change with each version.

A third-party project named HPy aims to ease the maintenance burden on the developer. HPy is a substitute C API for

Python—stabler across versions, yielding faster code at runtime and abstracted from CPython's often messy internals. The downside is that it's an opt-in project, not a requirement but various key projects like NumPy are experimenting with using it and some (like the HPy port of ultrajson) are enjoying big performance gains as a result. The biggest win for cleaning up the C API is that it opens the door to many more kinds of improvements that previously weren't possible. Like all the other improvements described here, they're about paving the way toward future Python versions that run faster and more efficiently than ever.

SOWMIYA A
B.Sc. (Information Technology)

APACHE SPARK: THE BIG DATA PLATFORM THAT CRUSHED HADOOP

Apache Spark

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the worlds of big data and machine learning which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

From its humble beginnings in the AMPLab at U.C. Berkeley in 2009, Apache Spark has become one of the key big data distributed processing frameworks in the world. Spark can be deployed in a variety of ways, provides native bindings for the Java, Scala, Python, and R programming languages, and supports SQL, streaming data, machine learning and graph processing. You’ll find it used by banks, telecommunications companies, games companies, governments and all of the major tech giants such as Apple, IBM, Meta, and Microsoft.

Spark RDD

At the heart of Apache Spark is the concept of the Resilient Distributed Dataset (RDD), a programming abstraction that represents an immutable collection of objects that can be split across a computing cluster. Operations on the RDDs can also be split across the cluster and executed in a parallel batch process leading to fast and scalable parallel processing. Apache Spark turns the user's data processing commands into a Directed Acyclic Graph, or DAG. The DAG is Apache Spark's scheduling layer; it determines what tasks are executed on what nodes and in what sequence.

RDDs can be created from simple text files, SQL databases, NoSQL stores (such as Cassandra and MongoDB), Amazon S3 buckets, and much more besides. Much of the Spark Core API is built on this RDD concept, enabling traditional map and reduce functionality, but also providing built-in support for joining data sets, filtering, sampling and aggregation.

J.KAVYA SHREE

II B.Sc. (Information Technology)

CLOUD COMPUTING VS TRADITIONAL COMPUTING

Cloud computing is a technology that allows users to access applications, data and services through a shared network. It is a form of distributed computing that enables users to access data from a remote server rather than from their computer. This type of computing is advantageous because it saves users the time and cost of purchasing and maintaining their hardware and software. It also allows users to access their data anywhere, regardless of their physical location.

Cloud computing is a technology that delivers computing services like applications, storage, and other resources over the Internet. This service allows users to access data from any device with an internet connection instead of having to install and maintain software applications. Cloud computing will enable companies to access applications and services from a central server, reducing the costs associated with managing and maintaining software applications. Additionally, cloud computing can help businesses reduce operating costs by allowing for a pay-as-you-go model for services and storage.

Cloud computing is a technology that allows users to access applications, data and services through a shared network. It is a form of distributed computing that enables users to access data from a remote server rather than from their computer. This type of computing is advantageous because it saves users the time and cost of purchasing and maintaining their hardware and software. It also allows users to access their data anywhere, regardless of their physical location.

Cloud computing is a technology that delivers computing services like applications, storage and other resources over the Internet. This service allows users to access data from any device with an internet connection instead of having to install and maintain software applications. Cloud computing will enable companies to access applications and services from a central server, reducing the costs associated with managing and maintaining software applications. Additionally, cloud computing can help businesses reduce operating costs by allowing for a pay-as-you-go model for services and storage.

Traditional Computing

Traditional computing uses hardware, software and other related services from a centralized location. This type of computing requires businesses to invest in hardware, software and other services and personnel to manage and maintain them. Traditional computing can be expensive for companies, especially when compared to the cost-saving advantages of cloud computing. Furthermore, traditional computing does not allow for the same remote access and scalability levels that cloud computing offers.

Difference Between Cloud Computing and Traditional Computing

Cloud computing has emerged as a popular technology due to its numerous advantages over traditional computing. Unlike traditional computing, cloud computing allows businesses to access software, hardware and other services remotely while scaling up or down as needed. Companies only pay for the services they need, which can significantly reduce initial investment and ongoing operational costs. Additionally, cloud computing is more secure and reliable than traditional computing due to the ability to access data remotely and the high levels of encryption and security protocols used by cloud service providers.

Cloud computing is a newer approach that eliminates the need for businesses to purchase and manage hardware, software and other related services. Instead, businesses rent or lease access to computing resources hosted and managed remotely by a third-party provider. This model of computing offers significant cost savings and allows businesses to scale up or down their computing resources as needed quickly. Additionally, cloud computing allows for greater access to data from anywhere, anytime.

Cloud or Traditional Computing

When it comes to which model of computing is better, cloud or traditional computing, it depends on the needs of the business. While cloud computing offers cost savings and scalability, traditional computing provides greater control and customization. Ultimately, the decision comes down to what the company needs and what its budget allows. Cloud computing is likely the better option for businesses requiring a lot of flexibility. Ultimately, Cloud computing offers cost-savings, scalability and the ability to access data from anywhere. Traditional computing requires businesses to purchase and manage hardware, software and other related services which can be expensive and difficult to scale. Therefore, cloud computing may be a better choice for businesses looking to reduce initial investment costs and take advantage of the scalability of the cloud.

Is Cloud Computing the Future?

Cloud Computing is a relatively new technology that has been gaining traction in the computing world. It allows users to access powerful computing resources remotely and do more with less. It has already revolutionized how businesses operate and is quickly becoming a standard for many applications. Its ability to scale rapidly and allow for rapid deployment of applications makes it an attractive option for many companies looking to remain competitive in today’s economy. Cloud computing is becoming increasingly popular as more businesses and individuals rely on the Internet for day-to-day operations. This technology allows users to access data from the cloud rather than store it on their computers or servers. This is beneficial for many reasons such as having access to data from anywhere, anytime and not worrying about the costs associated with maintaining servers. Additionally, cloud computing can reduce energy and help organizations remain agile and adaptable in changing times.

Is Cloud Computing Safe?

In cloud computing, security is essential when using the cloud and organizations should take measures to protect their data. Companies should encrypt data before sending it to the cloud, use two-factor authentication for access and use firewalls to prevent unauthorized access. Organizations should also ensure to keep their systems up-to-date with the latest security patches and software updates. Despite the many benefits of cloud computing, some people still need to be more hesitant to use the technology due to security concerns. While it is true that cloud computing can present some security risks, these risks can be mitigated with suitable security protocols in place. By implementing strong authentication and encryption, organizations can ensure that their data is kept safe and secure. Cloud service providers invest heavily in security to ensure that their customers' data remains private and secure.

Benefits of Cloud Computing

There are numerous benefits to using cloud computing. Cloud services offer scalability and flexibility, allowing organizations to quickly and easily access additional resources as needed. Cloud services also provide cost savings since organizations no longer need to purchase and maintain their hardware and software. Additionally, cloud services are easier to manage and support, as the cloud provider is responsible for the security and maintenance of their systems.

Cloud computing includes cost savings, scalability and flexibility. By allowing organizations to access computing resources on-demand, cloud computing reduces IT costs and enables them to focus their resources on more value-added tasks. By leveraging the economies of scale of cloud computing, organizations can access powerful computing resources without investing in expensive hardware or software. Cloud computing also provides organizations with increased flexibility and scalability to expand or contract as needed.

Both cloud computing and traditional computing have various advantages and disadvantages. Cloud computing is increasingly becoming more popular due to its ease of use and cost-effectiveness. However, traditional computing still has its place in the market, as it provides better security and control. Ultimately, it is up to the user to decide which is the best option for them, depending on their individual needs and preferences. Additionally, with cloud computing, businesses can focus on innovation and development, leaving the routine maintenance of IT systems to the service provider. As such, cloud computing has proven to be a valuable choice for businesses needing a more modern, reliable and cost-effective way to manage their data and applications.

Cloud computing has been gaining traction recently with many businesses and organizations turning to it as an alternative to traditional computing. It offers several advantages such as scalability and cost-effectiveness, as well as improved access to data and applications. In addition, cloud computing is more secure than traditional systems due to the ability of providers to keep infrastructure and procedures up to date. Ultimately, cloud computing provides organizations with a reliable, efficient and cost-effective way to store and access data, applications and services.

S. MIRUNALINI
III B.Sc. (Computer Technology)

JDK 2: THE NEW FEATURES IN JAVA 21

Feature proposals for the next long-term support release of standard Java have been turned up to 11, with the addition of new language features designed to make it easier to learn Java. Java Development Kit (JDK) 21, due in September as the next long-term support release of Oracle's standard Java implementation, now has 11 features officially proposed for it, with three more features added in recent days.

The latest features include a generational Shenandoah garbage collector, unnamed classes and instance main methods and unnamed patterns and variables. Separately, JDK 21 also is due to change how network names are assigned to network interfaces on Windows. The three newly proposed features join eight features proposed previously: generational ZGC (Z Garbage Collector), record patterns, pattern matching for switch expressions and statements, a vector API, sequenced collections, virtual threads, a preview of string templates, and a third preview of a foreign function and memory API. Early-access binaries under the GPL are available at jdk.java.net. Oracle publishes new releases of standard Java every six months, with the most recent, JDK 20, having arrived March 21.

The specific proposals for JDK 21 so far include:

- **Generational Shenandoah**, a proposal to enhance the Shenandoah GC with experimental generation collection capabilities to boost sustainable throughput, load-spike resilience and memory utilization. The main goal of the proposal is to provide an experimental generational mode without breaking non-generational Shenandoah with the intent to make the generational mode the default in a future release. Other goals include reducing the sustained memory footprint without sacrificing low GC pauses, reducing CPU and power usage, sustaining high throughput and continuing to support compressed object pointers. The proposal initially would support x64 and AArch64, with support for other instruction sets added as the experimental mode progresses to readiness. It is not a goal to replace non-generational Shenandoah which will continue as the default mode of operation with no regressions in its performance or functionality. Improving performance of every conceivable workload also is not a goal.

- A preview of **unnamed classes and instance main methods** to evolve the language so that students will be able write their first Java programs without needing to understand language features designed for large programs. Far from using a separate dialect of Java, students could write streamlined declarations for single-class programs and then seamlessly expand programs to use more advanced features as their skills grow. The intent is to offer a smooth onramp to Java.
- A preview of **unnamed patterns and variables** to enhance the language with unnamed patterns and unnamed variables. Unnamed patterns match a record component without stating the component's name or type, while unnamed variables can be initialized but not used. Both are denoted by an underscore character, _. This proposal is intended to improve the readability of record patterns by eliding unnecessary nested patterns and to improve maintainability of all code by identifying variables that must be declared but will not be used.
- **Generational ZGC** is intended to improve application performance by extending ZGC to maintain separate generations for young and old objects. Young objects tend to die young and maintaining separate generations will allow ZGC to collect them more frequently. Applications running with generational ZGC should see the following benefits: lower risks of allocation stalls, lower required heap memory overhead, and lower garbage collection CPU overhead. These benefits should happen without significant throughput reduction compared to non-generational ZGC.
- **Record patterns,** previewed in both JDK 19 and JDK 20, would deconstruct record values. Record patterns and type patterns can be nested to enable a powerful, declarative and composable form of data navigation and processing. Goals of the proposal include extending pattern matching to destructure instances of record classes and adding nested patterns, enabling more composable data queries. This feature has co-evolved with pattern matching for switch. Record patterns in the current JEP (JDK Enhancement Proposal) proposes to finalize the feature with further refinements based on continued experience and feedback. Apart from minor editorial changes, the main change since the second preview is to remove support for record patterns appearing in the header of an enhanced for statement. The feature may be re-proposed in a future JEP.
- **Pattern matching** for switch enables a switch expression or statement to be tested against a number of patterns, each with a specific action, so complex data-oriented queries can be expressed safely and concisely. This feature originally was proposed in JDK 17 and subsequently was refined in JDK 18, JDK 19, and JDK 20. It would be finalized in JDK 21 with further refinements based upon feedback and experience. Main changes from previous JEPs are the removal of parenthesized

patterns and allowing qualified enum constants such as case constants with switch expressions and statements. Goals include expanding the expressiveness and applicability of switch expressions and statements by allowing patterns to appear in case labels, allowing historical null-hostility of switch to be relaxed when desired and increasing the safety of switch statements by requiring that pattern switch statements cover all potential input values. Another goal is ensuring existing switch expressions and statements continue to compile with no changes and execute with identical semantics.

- A sixth incubator of a **vector API** to express vector computations that reliably compile at run time to optimal vector instructions on supported CPU architectures, achieving performance superior to equivalent scalar computations. This previously has been incubated in JDK 16 through JDK 20. The latest incarnation includes performance enhancements and bug fixes. Goals of the proposal include being clear and concise, being platform agnostic and offering reliable runtime compilation and performance on x64 and AArch64 architectures. Other goals include graceful degradation, for when a vector computation cannot be fully expressed at runtime as a sequence of vector instructions, and alignment with **Project Valhalla.**
- The **foreign function and memory API** enables Java programs to interoperate with code and data outside the Java runtime. By efficiently invoking foreign functions and safely accessing foreign memory, this preview API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI (Java Native Interface). The API previously was previewed in JDK 20 which debuted last month and JDK 19, which was released in September 2022. Refinements in the latest preview include enhanced layout paths with a new element to dereference address layouts, centralized management of the lifetimes of native segments in the Arena interface, a fallback native linker implementation and removal of the VaList. Goals of the proposal include ease of use, performance, generality and safety. It is not a goal to either reimplement JNI on top of this API or change JNI in any way.
- **Virtual threads** are lightweight threads that promise to “dramatically” reduce the effort of writing, maintaining, and observing high-throughput concurrent applications. Goals of the plan include enabling server applications written in the simple thread-per-request style to scale with near-optimal hardware utilization, enabling existing code that uses the lang. Thread API to adopt virtual threads with minimal change and enabling easy debugging and profiling of virtual threads with current JDK tools. Previously previewed in both JDK 20 and JDK 19, virtual threads will be finalized in JDK 21. With JDK 21, virtual threads now support thread-local variables all of the time, and make it impossible to create virtual threads

that do not have these variables. Guaranteed support for thread-local variables ensures that more existing libraries can be used unchanged with virtual threads and assists with migrating task-oriented code to use virtual threads.

- **Sequenced collections** introduces interfaces to represent collections with a defined encounter order. Each collection has well-defined first and second elements and so forth, to the last element. Uniform APIs are provided for accepting first and last elements and processing elements in reverse order. Motivating the proposal is a situation in which Java's collections framework lacks a collection type to represent a sequence of elements with a defined encounter order. It also lacks a uniform set of operations that apply across these collections. These gaps have been a problem and a source of complaints. The proposal calls for defining interfaces for sequencing for collections, sets and maps and retrofitting this into the existing collections type hierarchy. All of these new methods have default implementations.
- **String templates**, to appear as a preview feature, complement Java's existing string literals and text blocks by coupling literal text with embedded expressions and processors to produce specialized results. This language feature and API is intended to simplify writing of Java programs by making it easy to express strings that include values computed at runtime. It promises to enhance readability of expressions, improve program security, retain flexibility and simplify the use of APIs that accept strings written in non-Java languages. Enabling development of non-string expressions derived from combining literal text and embedded expressions also is a goal.

S.JAGADESH

III B.Sc. (Information Technology)

POPULAR MACHINE LEARNING ALGORITHMS

Linear regression

Linear regression, also called least squares regression, is the simplest supervised machine learning algorithm for predicting numeric values. In some cases, linear regression doesn't even require an optimizer, since it is solvable in closed form. Otherwise, it is easily optimized using gradient descent (see below). The assumption of linear regression is that the objective function is linearly correlated with the independent variables. That may or may not be true for your data.

To the despair of data scientists, business analysts often blithely apply linear regression to prediction problems and then stop, without even producing scatter plots or calculating correlations to see if the underlying assumption is reasonable. Don't fall into that trap. It's not that hard to do your exploratory data analysis and then have the computer try all the reasonable machine learning algorithms to see which ones work the best. By all means, try linear regression, but treat the result as a baseline, not a final answer.

Gradient descent

Optimization methods for machine learning, including neural networks, typically use some form of gradient descent algorithm to drive the back propagation, often with a mechanism to help avoid becoming stuck in local minima, such as optimizing randomly selected mini-batches (stochastic gradient descent) and applying momentum corrections to the gradient. Some optimization algorithms also adapt the learning rates of the model parameters by looking at the gradient history (AdaGrad, RMSProp, and Adam).

Logistic regression

Classification algorithms can find solutions to supervised learning problems that ask for a choice (or determination of probability) between two or more classes. Logistic regression is a method for solving categorical classification problems that uses linear regression inside a sigmoid or logit function, which compresses the values to a range of 0 to 1 and gives you a probability. Like linear regression for numerical prediction, logistic regression is a good first method for categorical prediction but shouldn't be the last method you try.

Support vector machines

Support vector machines (SVMs) are a kind of parametric classification model, a geometric way of separating and classifying two label classes. In the simplest case of well-separated classes with two variables, an SVM finds the straight line that best separates the two groups of points on a plane.

In more complicated cases, the points can be projected into a higher-dimensional space and the SVM finds the plane or hyperplane that best separates the classes. The projection is called a kernel and the process is called the kernel trick. After you reverse the projection, the resulting boundary is often nonlinear.

When there are more than two classes, SVMs are used on the classes pairwise. When classes overlap, you can add a penalty factor for points that are misclassified; this is called a soft margin.

Decision tree

Decision trees (DTs) are a non-parametric supervised learning method used for both classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Decision trees are easy to interpret and cheap to deploy, but computationally expensive to train and prone to overfitting.

Random forest

The random forest model produces an ensemble of randomized decision trees and is used for both classification and regression. The aggregated ensemble either combines the votes modally or averages the probabilities from the decision trees. Random forest is a kind of bagging ensemble.

XGBoost

XGBoost (eXtreme Gradient Boosting) is a scalable, end-to-end, tree-boosting system that has produced state-of-the-art results on many machine learning challenges. Bagging and boosting are often mentioned in the same breath. The difference is that instead of generating an ensemble of randomized trees (RDFs), gradient tree boosting starts with a single decision or regression tree, optimizes it, and then builds the next tree from the residuals of the first tree.

K-means clustering

The k-means clustering problem attempts to divide n observations into k clusters using the Euclidean distance metric, with the objective of minimizing the variance (sum of squares) within each cluster. It is an unsupervised method of vector quantization and is useful for feature learning and for providing a starting point for other algorithms.

Lloyd's algorithm (iterative cluster agglomeration with centroid updates) is the most common heuristic used to solve the problem. It is relatively efficient, but doesn't guarantee global convergence. To improve that, people often run the algorithm multiple times using random initial cluster centroids generated by the Forgy or random partition methods.

K-means assumes spherical clusters that are separable so that the mean converges towards the cluster center, and also assumes that the ordering of the data points does not matter. The clusters are expected to be of similar size, so that the assignment to the nearest cluster center is the correct assignment.

Principal component analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated numeric variables into a set of values of linearly uncorrelated variables called principal components. Karl Pearson invented PCA in 1901. PCA can be accomplished by eigenvalue decomposition of a data covariance (or correlation) matrix, or singular value decomposition (SVD) of a data matrix, usually after a normalization step applied to the initial data.

Popular deep learning algorithms

There are a number of very successful and widely adopted deep learning paradigms, the most recent being the transformer architecture behind today's generative AI models.

Convolutional neural networks

Convolutional Neural Networks (CNNs) are a type of deep neural network often used for machine vision. They have the desirable property of being position-independent.

The understandable summary of a convolution layer when applied to images is that it slides over the image spatially, computing dot products; each unit in the layer shares one set of weights. CNNs can also have pooling and fully connected layers, although there is a trend toward getting rid of these types of layers.

DINESH S

I B.Sc. (Computer Technology)

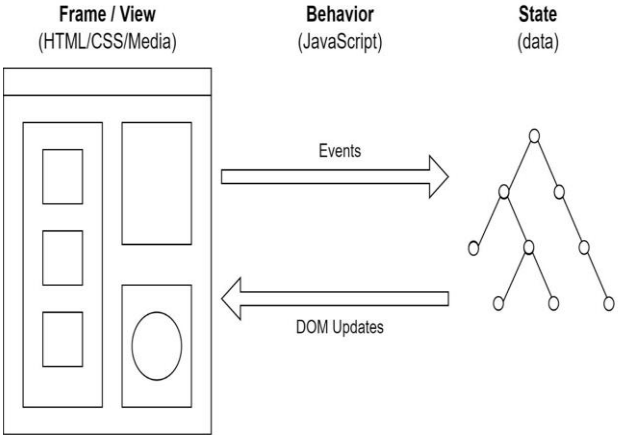
REACTIVE JAVASCRIPT: THE EVOLUTION OF FRONT-END ARCHITECTURE

Improving the client-side web experience means overcoming the challenges of hydration, a fascinating engineering problem being tackled in many different ways.

One of the most dynamic areas in software development today is front-end architecture. Several innovators are pushing the state of the art to devise more powerful ways to build dynamic user interfaces. Much of this work is happening at a furious pace and right out in the open.

What is hydration?

Much of the activity around improving modern front-end architecture is focused on what's called hydration. To understand what hydration is and why it's central to modern front-end architecture, let's get a grip on the high-level concepts at play. To deliver the wonder of reactivity, every framework must handle the three aspects illustrated in the diagram below.



The basic message in the diagram is that the framework is responsible for framing the view, holding the state and managing the interaction between them. The naive or default, approach is to simply take everything the client needs the frame, the reactive code, and the state and send it over. The client (the browser) then does the work of displaying the frame (aka, painting the UI), interpreting the JavaScript, and tying in the state.

This approach has the wonderful benefit of simplicity, both for the code at work and for the human minds trying to understand it. It also has a big downside: The initial page render has to wait on everything and the user has to sit through all of that network and browser churn. Also, unless care is taken, the page will tend to display and then embarrassingly rearrange itself into the final layout.

This inspired developer to try rendering the initial page on the server first *(*server-side rendering or SSR) and send it over. Then, the user has a decent page to look at while the rest of the code and state is sent and bootstrapped. This is a great simplification but that's the basic idea. The time it takes to get the basic layout in place is called first contentful paint (FCP). The next milestone the page needs to reach is measured by time to interactive (TTI), meaning the time until the user is able to actually use the page.

Limits of server-side rendering

The bottom line is that SSR tends to improve FCP but worsen TTI. Thus the goal has become striking a balance between the two while maximizing them both, while hopefully maintaining a pleasant developer experience (DX).

A variety of approaches have been proposed, adopted, abandoned, modified and combined in this effort to improve hydration. Once one starts looking at the implementation details, one is amazed at how complex it becomes. A balanced enhancement of FCP and TTI with a decent DX? Sounds easy but it isn't.

One reason for the complexity is that we're smack in the middle of sorting through all of the trade-offs; it's an unfolding scene. Once the way forward crystallizes though, we should expect two results from the client architecture that emerges. First, it should create web apps that feel "next generation," in the same way that well-built apps today provide a subtly but clearly better experience than one from a few years ago.

Second and perhaps even more importantly, our improved client architecture should have far reaching consequences beyond better performance. By wading into and resolving the complexity, front-end engineers will arrive at a better model, for both the system and the mind. A better architecture actually represents a more powerful heuristic. This results in follow-on benefits that are often unpredictable. You can see this in action with reactivity itself. Reactivity burst onto the scene because it offered a way to offload state binding from the developer's brain to the framework. But the benefits didn't stop there. The architecture became not only simpler but more consistent. This netted performance and functionality gains across the board.

Because modern JavaScript frameworks incorporate both server and client, the outcomes of these developments may have broad consequences for application architecture in general.

Approaches to improving hydration

The basic trick to improving the hydration situation is to look at things more granularly. By breaking the view, the interactivity and the state into smaller pieces, we can load and activate them stepwise, optimized for FCP and TTI. Here is a tour of some of the approaches.

Avoiding JavaScript entirely

One approach that has been absorbed in best practice is to analyze sites for those pages that don't require JavaScript at all. This relates to the newer notion of multipage apps (MPA). It is a kind of middle ground between single page apps (SPA) and straight-up per-page navigation (default web behavior). The idea here is to find the parts of the app that can be shipped immediately as HTML plus assets, resulting in the best possible SEO and load times.

The no-JS approach is seen in SvelteKit, for example. This doesn't do anything for those pages that require reactive interaction, of course. Frameworks still must address hydration on those pages that act as SPA.

Front-end architecture evolution

The activity around JavaScript's front-end architecture has created some of the most interesting code work I've ever witnessed. It's a space full of passionate individuals who are exploring new conceptual territory and doing the ground breaking programming to go with it. And they're interacting and sharing their ideas in an open and collaborative way.

B.THARNIKA

III B.Sc. (Information Technology)

INTERVIEW TECHNICAL QUESTIONS

1. What type of process creates a smaller file that is faster to transfer over the internet?
(A) Compression
(B) Fragmentation
(C) Encapsulation
(D) None of the above
ANS: A

2. IC chips used in computers are usually made of
(A) Lead
(B) Silicon
(C) Chromium
(D) Gold
ANS:B

3. To find the user's password for hacking purposes refers to
(A) Sniffing
(B) Spoofing
(C) Cyberstalking
(D) Spamming
ANS: A

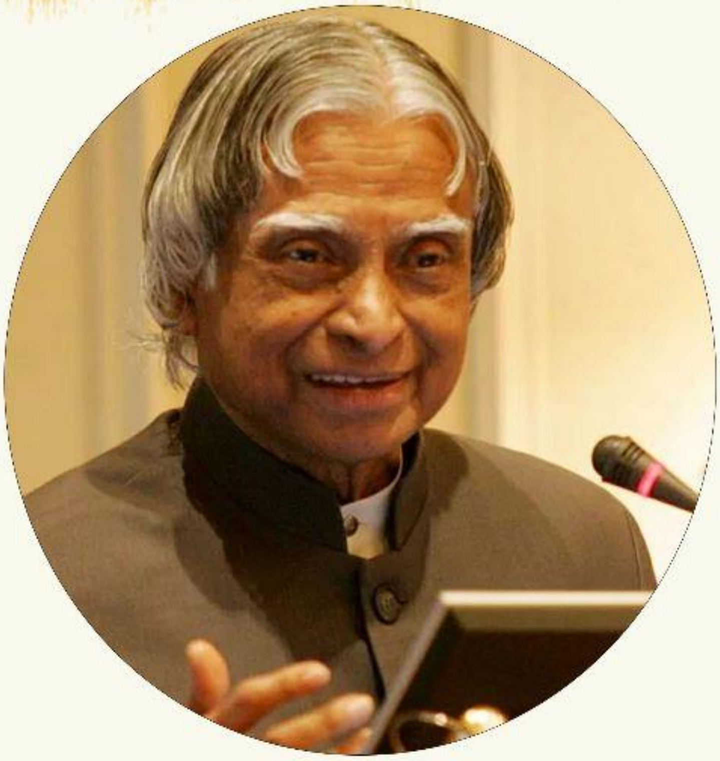
4. Which one of the following is not a common Networktopology?
(A)Bus
(B)Star
(C)Ring
(D) Grid
ANS:D

5. Blowfish is a type of

(A) Symmetric Encryption Algorithm
(B) Hashing Algorithm
(C) Digital Signature Algorithm
(D) Asymmetric Encryption Algorithm
ANS:A

M. HARINI

I B.Sc. (Computer Technology)



The Right Signal is that Technology is going to boost (Economic) development of Our Nation..

-ABJ ABDUL KALAM-