NOV-DEC 2025 VOLUME 6 | ISSUE 6

DIGITAL OUTLET

REVOLUTIONIZING INDUSTRIES, TRANSFORMING LIVES

CYBER FUTURE

HOW TECHNOLOGY IS SAFEGUARDING THE DIGITAL WORLD

GREEN TECH

SUSTAINABILITY MEETS SMART SOLUTIONS

Unlocking the Potential of Augmented Reality

Space Tech

Smart Gadgets to Simplify Your Life

Cybersecurity 2025

Are We Prepared?

PUBLISHED BY-
CCET ACM STUDENT CHAPTER
CCET, DEGREE WING, SECTOR 26
CHANDIGARH

CONTENT

Note from our *Mentors*

Dr. Manpreet Singh

Principal CCET (Degree Wing)

Our mission at CCET is not only to produce engineering graduates but to produce engineering minds

Dr. Sunil K. Singh

Professor and HOD, CSE | Faculty Mentor

ACM CCET provides student a great opportunity to learn scientific and practical approach of computer science.

Dr. Sudhakar Kumar

Assistant Professor, CSE | Faculty Sponsor

Every person should be provided with an opportunity to learn and explore the field of computer science.

"The greatest achievement of technology is not how it changes life, but how it improves it."

ACM'S VISION AND MISSION

VISION

Chandigarh College of Engineering and Technology aims to be a center of excellence for imparting technical education and serving the society with self-motivated and highly competent technocrats.

MISSION

1. To provide high quality and value based technical education.
2. To establish a center of excellence in emerging and cutting-edge technologies by encouraging research and consultancy in collaboration with industry and organizations of repute.
3. To foster a transformative learning environment for technocrats focused on inter-disciplinary knowledge; problem-solving; leadership, communication, and interpersonal skills.
4. To imbibe spirit of entrepreneurship and innovation for development of enterprising leaders for contributing to Nation progress and Humanity.

DEPARTMENT VISION AND MISSION

VISION

To produce self-motivated and globally competent technocrats equipped with computing, innovation, and human values for ever changing world and shape them towards serving the society.

MISSION

M1: To make the department a smart centre for learning, innovation and research, creativity, and entrepreneurship for the stakeholders (students/scholar, faculty, and staff).
M2: To inculcate a strong background in mathematical, theoretical, analytical, and practical knowledge in computer science and engineering.
M3: To promote interaction with institutions, industries and research organizations to enable them to develop as technocrats, entrepreneurs, and business leaders of the future.
M4: To provide a friendly environment while developing interpersonal skills to bring out technocrat's inherent talents for their all-round growth.

ASSOCIATION FOR COMPUTING MACHINERY AT CCET

ABOUT

The CCET ACM Student Chapter brings together the Association for Computing Machinery (ACM) and ACM-W, fostering a vibrant community of computing enthusiasts committed to innovation, learning, and inclusivity. Under the expert mentorship of Dr. Sunil K. Singh and Dr. Sudhakar Kumar, the chapter actively organizes technical workshops, coding competitions, hackathons, and outreach programs that encourage both skill development and collaboration. While ACM focuses on advancing computing as a science and profession, ACM-W works towards empowering and supporting women in computing, ensuring equal opportunities and representation. Together, they create a dynamic platform at CCET where students can explore emerging technologies, share knowledge, and grow as competent and responsible computing professionals.

Student Speaker Program

Designing & Digital Art

Research and Development

Competitive Codong

Internship & Career Opportunity

CCET ACM STUDENT CHAPTER

ABOUT

ACM boosts up the potential and talent, supporting the overall development needs of the students to facilitate a structured path from education to employment. Our Chapter CASC focuses on all the aspects of growth and development towards computer technologies and various different elds. Overall, we at CCET ACM Student Chapter, through collaboration and engagement in a plethora of technical activities and projects, envision building a community of like-minded people who love to code, share their views, technical experiences, and have fun. We have been trying to encourage more women to join the computing eld, so we started an ACM-W Chapter to increase the morale of women. CASC launched an app which aimed at maintaining deco- rum of reading among CS members and sharing their ideas.

Student Speaker Program

Designing & Digital Art

Research and Development

Competitive Codong

Internship & Career Opportunity

CCET ACM-W STUDENT CHAPTER

ABOUT

The CCET ACM-W was founded in October 2021 with an aim to empower women in the field of computing and increase the global visibility of women in the field of research as well as development. We provide a platform for like-minded people so that they can grow together and contribute to the com- munity in a way that shapes a better world. Our chapter was founded to encourage students, especially women, to work in the field of computing. The chapter's main goal is to create even opportunities and a positive environment for students, where they can work to develop themselves profession- ally. We at the ACM Student chapter aim to build a globally visible platform where like-minded people can collaborate and develop in their field of interest.

Student Speaker Program

Designing & Digital Art

Research and Development

Competitive Codong

Internship & Career Opportunity

ACHIEVEMENTS

Our team's recent publications showcase notable progress in diverse areas of Computer Science, including deep learning, security, and intelligent systems. These works have appeared in high-impact journals, conferences, and book chapters during the initial part of 2025.

Journal Articles

- Advanced Web Traffic Modelling and Forecasting with a Hybrid Predictive Approach
  Author- Ujjwal Thakur, Sunil Kr Singh, Sudhakar Kumar, Kwok Tai Chui

- AI Developments for Industrial Robotics and Intelligent Drones
  Author- Divyansh Manro

- Innovations in Modern Cryptography
  Author- Divyansh Manro

- Quantum-Resistant Cryptographic Primitives Using Modular Hash Learning Algorithms for Enhanced SCADA System Security
  Author- Sunil Kr Singh, Sudhakar Kumar, Manraj Sing, Brij B. Gupta

- FPA-based weighted average ensemble of deep learning models for classification of lung cancer using CT scan images
  Author- Liang Zhou, Achin Jain, Arun Kumar Dubey, Brij B. Gupta

Book Chapters

- Understanding Cyber Threats in Modern Space Missions published in ResearchGate
  Author- Anoop Pant, Harmanjot Singh

- Multi-Modal Sensor Fusion With CRNNs for Robust Object Detection and Simultaneous Localization and Mapping (SLAM) in Agile Industrial Drones
  Author- Ujjwal Thakur

- Robust and Secure Communication Protocols for Space Missions February 2025
  Author- Anoop Pant, Harmanjot Singh

- Future Trends in AI-Driven Green Computing and Security
  Author- Deepak Bhattarai

- Book Chapters : Computing in Modern Healthcare Systems Emerging Trends for Simulation, Training and Smart Healthcare
  Author-Devesh Vaidya

Conference Papers

- Research Paper published: AI ML Model Based
  Author- Atul A

- Research paper
  Author- Mehak Negi

- Novel Graph Neural Network for Real-Time Blockchain Anomaly Detection with Smart Contract Support
  Author- Sanatan Sharma

- Blockchain-based data security in smart cities: Ensuring data integrity and trustworthiness
  Author- Sahil Garg

- Gestational Diabetes Prediction using Machine Learning for Consumer Electronics Healthcare
  Author- Sahil Garg

EVENTS

COMPETITIVE PROGRAMMING

The Competitive Programming Contest at APRATIM'25 was a time-limited, individual coding competition intended to assess competitors' skills in data structures, algorithms, and logical problem-solving. Under tight time constraints, participants completed a series of carefully chosen tasks, and their accuracy and efficiency were assessed. The event encouraged strategic thinking and optimal solutions by simulating a real competitive coding environment with live rankings. It gave students practical experience with contest-style problem solving that is applicable to coding interviews and international programming competitions.

GLIMPSE OF THE EVENT

EVENTS

DESIGN-SCAPE

The Design-Scape competition at APRATIM'25 was a creative and technical design challenge aimed at testing participants' proficiency in digital art, visual communication, and photo editing. Tasked with transforming raw assets into original masterpieces, competitors were evaluated on their aesthetic judgment, technical execution, and ability to blend imagination with functional design.

The event strictly prohibited AI-generated content, focusing instead on the manual mastery of professional software like Adobe Photoshop, Illustrator, and Blender. By simulating a fast-paced creative studio environment, Design-Scape provided students with a platform to showcase their artistic vision and technical agility—skills highly sought after in the fields of UI/UX design, digital marketing, and the creative industries.

GLIMPSE OF THE EVENT

EVENTS

HACK-SPRINT

The Hack-Sprint competition at APRATIM'25 was a high-octane, collaborative marathon designed to challenge participants' innovation, rapid prototyping, and full-stack development skills. Working in dynamic teams, competitors were tasked with building functional solutions to real-world problems within a rigorous timeframe. The event emphasized not only clean and efficient code but also the viability, scalability, and user experience of the final product. By bridging the gap between theoretical knowledge and practical application, Hack-Sprint provided students with an intense, industry-simulated environment to sharpen their project management and technical execution—essential qualities for the modern tech landscape.

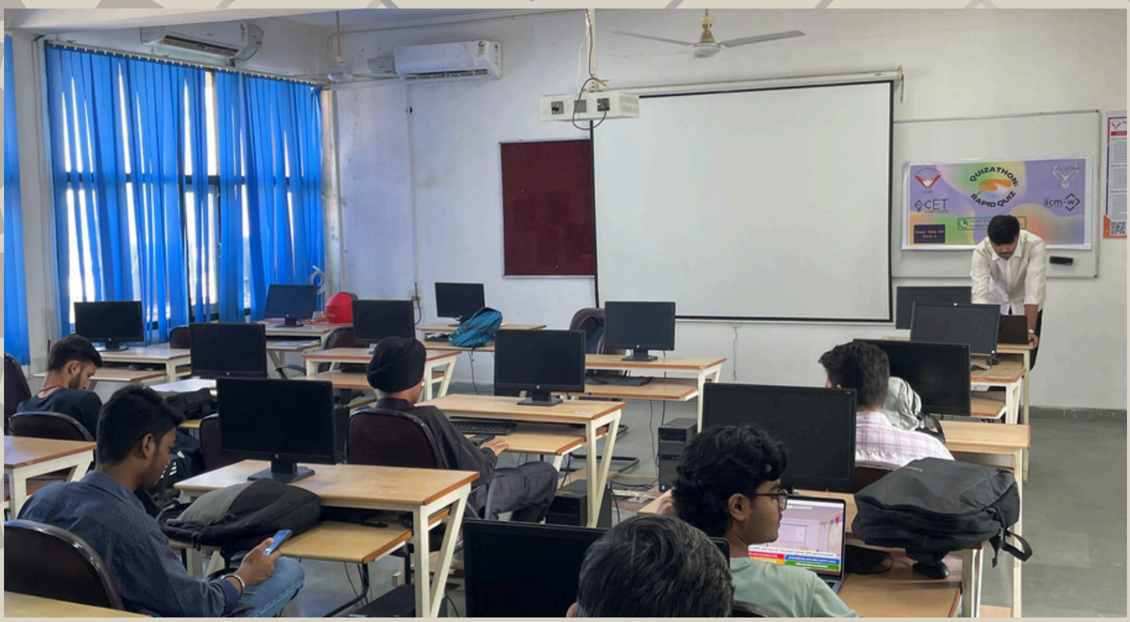
GLIMPSE OF THE EVENT

EVENTS

QUIZATHON

The Quizathon at APRATIM'25 was an intellectually stimulating battle of wits aimed at testing participants' depth of knowledge, quick thinking, and mental agility. Spanning a diverse range of topics—from core technical concepts to current affairs and general awareness—the competition utilized a multi-round format to filter for the sharpest minds. Under pressure, contestants had to demonstrate both accuracy and speed, navigating through complex question sets that rewarded curiosity and broad learning. This event served as a platform for students to benchmark their general and technical literacy against their peers, fostering a culture of continuous learning and intellectual curiosity.

GLIMPSE OF THE EVENT



Article

NTHE EVOLUTION OF OPERATING SYSTEMS: FROM MONOLITHS TO MICROKERNELS TO UNIKERNELS

Introduction: Why OS Architectures Matter

Operating systems (OS) are essential to the smooth operation of all computing devices, from laptops and smartphones to data center servers and embedded systems. By controlling memory, processes, files, networking, and system security, the operating system serves as an essential bridge between hardware and applications. For many years, the monolithic kernel architecture—an all-in-one strategy where almost all essential functions operate together in a single, privileged memory space—has been used to create popular operating systems like Windows, macOS, and Linux. However, other models like microkernels and unikernels are challenging this traditional design as computing paradigms change to require greater scalability, modularity, and security.

Lightweight, quick, and secure systems that can function under stringent resource and performance limitations are required due to the evolving computing environment, which includes cloud-native apps, serverless architectures, edge computing, and Internet of Things devices. In these kinds of settings, traditional OS designs can introduce needless costs. Newer paradigms can help with it. While unikernels push the boundaries by assembling programs into highly efficient, single-purpose machine images, microkernels seek to increase modularity and fault isolation.

Understanding the advantages and disadvantages of each OS model is essential for developers, system architects, and cybersecurity experts as these architectural advancements represent a larger change in the way we create and implement software.

Monolithic Kernels

The majority of general-purpose operating systems, such as Linux, Windows, and early Unix variations, employ the monolithic kernel, a classic design. According to this concept, the kernel's single, unified address space houses all of the fundamental OS functions, including memory management, device drivers, file systems, process management, and system calls.

By eliminating context switching and message transmission between components, this architecture improves speed. It is quick, strong, and, from a systems standpoint, easy to apply. However, fragility and intricacy are its main disadvantages. The stability and security of the entire system can be jeopardized by a defect in a single driver or module since everything operates in kernel space. Monolithic kernels are also susceptible in high-assurance settings due to their enormous attack surface. The monolithic architecture is beginning to show its age as systems grow increasingly linked and security becomes more important.

Microkernels

Microkernels, on the other hand, try to reduce the amount of software that operates in kernel space by moving the majority of OS operations, including file systems, networking, and drivers, into user space. The kernel itself is kept as thin as possible, usually including simply memory management, basic scheduling, and inter-process communication (IPC).

Because crashing a file system service, for example, doesn't bring down the entire system, this division improves fault isolation. Because each component may be sandboxed and runs with restricted rights, microkernels are also more secure and modular.

MINIX 3, L4, and seL4 are well-known instances; the latter is utilized in safety-critical settings such as military and aerospace systems and has been publicly shown to be mathematically valid. Because of the frequent message passing between kernel and user-space services, microkernels have historically suffered from performance overhead despite their benefits. Modern microkernel designs and hardware advancements, however, have greatly closed this performance gap, making them a viable substitute for systems that demand a high level of security and dependability.

Unikernels: The Minimalist Revolution

Unikernels represent the most profound rethinking of OS design. Application code and only the required operating system components are compiled to build a tailored, one-use machine image known as a unikernel. Just the very minimum required to operate the program is there; there is no general kernel, user space, or system daemons.

Unikernels provide remarkable speed, incredibly low attack surfaces, lightning-fast boot times (milliseconds), and small footprints (often less than 5 MB). Many conventional attack vectors are just nonexistent because there is no shell, users, or system processes. They can also run on cloud systems like AWS and Google Cloud, as well as hypervisors like Xen, and are incredibly portable.

OCaml-built MirageOS, C++-based IncludeOS, and OSv, which is intended to execute Linux-based programs in a unikernel form, are a few examples. Wherever you require the least amount of overhead and the most separation, such as in serverless computing, the Internet of Things, edge deployments, and high-performance microservices, Unikernels are perfect.

But adaptability suffers as a result of their extreme simplicity. There is little runtime insight, debugging is challenging, and typical Unix tools are absent. Nevertheless, the advantages of these trade-offs outweigh the drawbacks in many contemporary applications.

Real-World Use Cases

Every OS architecture offers advantages that fit particular use cases.

For general-purpose computing—desktops, cellphones, and large-scale Linux servers—where legacy compatibility and application diversity are crucial, monolithic kernels continue to rule the market.

Microkernels are excellent in safety-critical systems including military-grade hardware, medical equipment, and automobile ECUs. Defense contractors employ the formally validated seL4 microkernel to guarantee mission-critical accuracy.

In cloud-native settings, where security and resource efficiency are top concerns, unikernels are becoming more popular. While some businesses employ unikernels to implement isolated functions in serverless infrastructures or lightweight REST APIs, Cloudflare has investigated utilizing them for its DDoS prevention systems.

Benefits and Trade-offs

Each architecture offers unique benefits—but also introduces trade-offs:

Microkernels and unikernels are more appropriate for modular, distributed, and security-focused systems, whereas monolithic kernels provide extensive hardware support and familiarity with the ecosystem. Unikernels go farther to accomplish specialization through compilation, whereas microkernels strive for robustness through isolation.

The Future of OS Design

In the future, hybrid approaches and more domain-specific operating systems are probably in store. For instance, a lot of cloud services are investigating unikernel-like microVMs through the use of technologies like Kata Containers and Firecracker (from AWS Lambda), which combine the flexibility of containers with security isolation.

Another trend is the emergence of language-based operating systems, such Redox OS (written in Rust), which use safer programming languages in an effort to lessen memory safety concerns. These systems combine services into minimal, high-assurance execution environments, blurring the boundaries between OS and application.

In real-time or security-first applications, unikernels and microkernels could be the standard. Monolithic kernels will still be used for general-purpose devices, but they will have more modularity and sandboxing capabilities.

Ultimately, the “one size fits all” model for OS design is fading. Modularity, isolation, and purpose-built runtimes are the new hallmarks of modern computing environments.

Architecture	Pros	Cons
Monolithic	Fast, mature, rich features	Large attack surface, less fault tolerant
Microkernel	Secure, modular, fault-isolated	Slightly slower, more complex IPC
Unikernel	Minimal, fast, secure, boot in ms	Hard to debug, limited tooling

Modern Threat Models and Their Influence on OS Design

Modern threat models are now a major influence on operating system architecture due to the increasing complexity and frequency of cyber attacks. In a time of supply chain breaches, side-channel assaults, and advanced persistent threats (APTs), the conventional presumptions of reliable user-space apps and static perimeter protections are no longer valid. As a result, zero-trust concepts are being used to rethink OS designs, with a focus on secure-by-default setups, minimum trusted computing bases (TCBs), and stringent isolation. For example, by limiting the attack surface and confining failures to isolated components, microkernel and unikernel designs closely match contemporary threat models. In order to enable secure enclaves and confidential computing, OS kernels are increasingly incorporating hardware-backed security features like AMD SEV, ARM TrustZone, and Intel SGX, which protect data even while it is being executed. Moreover, modern OSes are prioritizing fine-grained access control, capability-based security, and real-time attestation mechanisms to proactively defend against threats in increasingly hostile and distributed environments.

Conclusion

Operating system design development is a reflection of larger computer trends toward performance, security, and specialization. OS architecture is getting increasingly workload-specific, from the monolithic giants that power our PCs to the svelte unikernels that enable serverless operations. These days, developers and engineers need to look outside the box and think about how the correct OS design may simplify deployments, lower vulnerabilities, and open up new efficiencies. Building the future of robust and efficient computing systems requires an awareness of this architectural change, regardless of whether you're designing for the cloud, the edge, or a mission-critical device.

References

- Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.

○ A foundational text covering OS architecture, including monolithic and microkernel design.

- Liedtke, J. (1995). "On Micro-Kernel Construction." Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP).
- https://doi.org/10.1145/224056.224075

○ Seminal paper outlining microkernel efficiency and implementation techniques.

- Madhavapeddy, A., & Scott, D. J. (2013). "Unikernels: Library Operating Systems for the Cloud." Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS).
- https://doi.org/10.1145/2451116.2451167

○ Introduces the unikernel concept and its use in cloud-native environments.

- Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., ... & Winwood, S. (2009). "seL4: Formal Verification of an OS Kernel." Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP).
- https://doi.org/10.1145/1629575.1629596
- ○ Discusses formal verification of microkernel-based OS for high-assurance systems.
- Baumann, A., Peinado, M., & Hunt, G. C. (2011). "Shielding Applications from an Untrusted Cloud with Haven." OSDI.
- https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/osdi14haven.pdf

○ Illustrates use of Intel SGX and trusted execution environments in OS design.

- Heiser, G., & Elphinstone, K. (2016). "A Verified Foundation for the Internet of Things." Communications of the ACM, 60(1), 62–70.
- https://doi.org/10.1145/3015455

○ Explores how verified OS kernels help address modern IoT threat models.

- Zhang, F., Chen, Y., & Mao, Z. M. (2016). "Operating System Support for Confidential Computing." IEEE Security & Privacy, 14(6), 56–60.
- https://doi.org/10.1109/MSP.2016.128

○ Overview of how OSs are adapting to confidential computing trends.

- National Institute of Standards and Technology (NIST). (2020). Zero Trust Architecture (SP 800-207).
- https://doi.org/10.6028/NIST.SP.800-207

○ Defines zero-trust principles that now inform OS-level threat models.

Dhruv Bali

Article

ARTIFICIAL INTELLIGENCE

Introduction

In modern time, artificial intelligence is everywhere from voice assistants to self-driving cars.AI is present everywhere for e.g. whenever a user types on any platform like google it suggests the word or when YouTube recommends videos. Artificial intelligence (AI) is changing how people live, work and their ability to think. It is one of the fastest growing technologies in the world. It is changing the way people interact with each other [1]. Many big companies are using it to improve their products. It can also be used to improve life but also has many disadvantages which can cause many issues too. In this article we have discussed its history, application, merits, demerits, future scope.

Historical Background and Evolution of AI

Long before computers were invented, the concept of artificial intelligence (AI) was conceived. The idea of building sentient machines has its roots in Greek mythology. However, the science of artificial intelligence began in the 1950s.

British mathematician Alan Turing developed the Turing Test in 1950 to determine whether a machine could think and behave similarly to a human [4].

John McCarthy, also known as the Father of AI, coined the term "Artificial Intelligence" at a conference held at Dartmouth College in 1956 [1].

.Several significant turning points in the evolution of AI include:

The first chatbot, ELIZA, was created in 1966 to mimic human speech [5].

In 1997, IBM's Deep Blue defeated Garry Kasparov, the world chess champion, demonstrating the effectiveness of AI in rational games [1].

2011: IBM's Watson defeated human champions in the Jeopardy! quiz show [1].

Beginning in 2018, AI systems such as Google Duplex began to make phone calls that resembled those of a human.

The 2020s saw the emergence of generative AI, such as ChatGPT, DALL·E, and GPT-4, which demonstrated AI's capacity for creation, writing, and communication [3].

Core Concepts and Working of AI

Artificial intelligence is defined as a means of teaching and making computers understand in the same way humans do [1]. It teaches machines by helping how to solve a problem, learn from data.Earlier computers were able to do only work which was asked by humans. But gradually AI was improving and by 1990 computers were able to play games like chess against humans.. Today AI is not used in games but also in many places such as hospitals, banks, schools etc. Artificial Intelligence works by taking data such as images, text, audio as input.

The system then analyzes the data; this process is called preprocessing. Artificial intelligenceis made to learn on the input data and allowed to make decisions. Based on the input data it gives output in the form of some graphs or text.

Types of Artificial Intelligence

Based on Capability

There are three types of Artificial intelligence: Narrow AI, General AI, Super AI.

Narrow AI is a type of AI. It is built to perform specific tasks like face unlocking in phones, calculator. The calculator can only perform mathematical operations on numbers only; it is not used to click pictures. Similarly Narrow AI can perform only specific tasks not all tasks [1].Some of the examples are Siri, Google Translate, Face Unlock.

General AI is a type of AI which thinks in the same way as humans think. It can perform multiple tasks at a time like can learn, understand and solve problems [1]. All these tasks can be performed at the same time. Scientists are working on inventing this type of AI.Some of the examples are Human-like robots (in research).

Super AI is a future idea of how AI will work. The idea is to build AI which is much smarter than all humans. It could perform all tasks better and faster than humans [1]. But at the moment it is all ideas and subject of research.

Based on Learning

Artificial Intelligence can be classified on the basis of learning methods also like: supervised learning, unsupervised learning, reinforcement learning. When Artificial intelligence learns from labelled data then that learning is called supervised learning [1]. When it finds hidden patterns from unlabeled data. This type of learning is called unsupervised learning [1]. AI learns through feedback that type of learning is called reinforcement learning [3].

Key Components of AI

Machine learning is part of Artificial Intelligence. It is defined as giving the ability to computers to learn from data given as input and improve with time and experience [1]. For e.g. if a user shows a machine thousands of images, then it will learn from the images and will be able to tell the difference on the basis of color, size, shape. The three types of machine learning are: classification, regression, clustering. Classification means sorting the data into categories like color, shape etc. Regression means predicting the numbers like what the temperature will be tomorrow. Clustering means grouping the similar unlabeled data together [4].

Deep learning is a part of machine learning that uses neural networks. A neural network is the machine that works exactly in the same way as humans do [3]. The human brain has neurons to pass signals to each other. Similarly a neural network has artificial neurons called nodes that are also used to pass signals to each other. A neural network has 3 main layers: input layers which take input like images, text etc.

The second layer is the hidden layer which learns the patterns of the data and last is the output layer which gives results [3]. Deep learning is generally used when input data is image and audio.

An Expert System is a type of Artificial Intelligence that acts to think like a human expert. It is designed to solve specific problems by using knowledge and logic. Expert systems have two main parts: a knowledge base, which stores facts and rules, and an inference engine, which applies those rules to solve problems [2]. For example, a medical expert system can help diagnose diseases based on symptoms. If a patient has a fever and sore throat, the system uses its rules to suggest possible illnesses and treatments. These systems are useful in areas like medicine, weather forecasting, and machine fault diagnosis. Although they don't truly "understand" things like humans do, they can still offer smart advice based on the rules they follow.

Data mining is the process of finding useful patterns and hidden information in large sets of data [4]. It helps organizations make better decisions by analyzing the data they already have. For example, a shopping website might use data mining to see which products are often bought together or what kinds of customers prefer certain products. The process involves several steps, such as cleaning the data, choosing the important parts, and using algorithms to spot trends or connections. Data mining is used in many fields like business (to understand customers), banking (to catch fraud), healthcare (to predict illnesses), and marketing (for targeted ads). It plays a key role in AI and machine learning because it provides the information these systems need to learn and make decisions.

Current Applications of AI

Artificial intelligence is used in many areas like Healthcare, Education, Finance, Entertainment and in daily life activities.

Healthcare: In healthcare it is used in the study of x-rays and blood sample results to diagnose any disease faster and accurately. For e.g. it can be used in predicting whether any person is suffering from brain tumor by analyzing the brain scans. It can also help in detecting the early stage of Alzheimer diseases which is difficult to detect manually. Even these days robots are also accompanying doctors in surgeries [3].

Education: In the Education field it helps students as there are numerous education apps that use AI to teach and make students understand the concepts. It helps teachers to make students understand the concept as it allows the teachers to display the concept visually and animately [1].

Finance: In Finance it helps banks to detect any kind of fraud, and approves loans faster. It is also used in customer support and can even chat and communicate with customers to resolve any kind of issue they face [4].

Entertainment: In Entertainment it is used to suggest or recommend videos and shows based on the user's liking and choices. It is also used in voice assistants like Alexa, Siri that answer all questions which users ask, can also play music and control any kind of smart appliance or devices [1].

Advantages and Disadvantages of AI

The advantages of using Artificial intelligence are that it can study large amounts of data quickly and can find correct answers faster as compared to human beings. It can be used for various tasks like tasks which are repetitive or tasks like entering data which take a lot of time to do manually but AI can complete the task in minutes. It can increase the productivity of work in many parts of industries. It is less prone to make mistakes as compared to human beings.AI does not need breaks like humans require and can work all time. The disadvantages of using Artificial intelligence are since AI can complete all tasks efficiently so it is replacing humans in industries like customer services and transport. If AI is trained using some wrong data, then it can make some wrong decisions too like rejecting a good application just because it was trained using unfair data. It requires a lot of personal data for its training so if the data is not handled properly, it can cause a lot of problems to the user as data can get misused. Relying much on AI can affect the thinking ability of human beings.

Future Scope of AI

Artificial intelligence has a bright future ahead of it. AI will advance in intelligence, resemblance to humans, and utility in all domains. The future scope of using AI could be any person's DNA report and can recommend medicines that are suitable for that person. It can also help in managing waste like smart dustbins can be established which contains different sections for all kinds of waste and if a person puts the wrong type of waste in any section, then it senses and alarms so that all waste gets segregated easily. It can also be used in parking areas for efficiently using the parking space. AI will also help with environmental protection, agriculture, education, and disaster forecasting. As generative AI advances, machines will be able to produce software code, artwork, and other types of content. It can be used to enhance the work of doctors, teachers, and engineers instead of replacing them.

Conclusion

Artificial Intelligence is one of the most important inventions of the time. It is changing a lot of things like how people live, learn. It can solve big problems very easily and quickly as compared to human beings. But like it has its own pros it has cons too. If it is used properly then it can be useful in multiple work but if not used properly then it can also harm people. It is immensely powerful due to its capacity for learning, adaptation, and improvement. However, that authority also carries responsibility. We must make sure AI is applied safely and ethically as it becomes more intelligent. We must create transparent, neutral, and fair AI systems. AI should improve human capabilities rather than replace them. If properly directed, artificial intelligence (AI) will develop into more than just a tool in the upcoming years; it will be a reliable collaborator for advancement, assisting us in creating a more intelligent and superior future.

References

1. Russell, S. and Norvig, P. (2021). Artificial Intelligence: A Modern Approach. 4th ed. Pearson Education.
2. Han, J., Pei, J., and Kamber, M. (2012). Data Mining: Concepts and Techniques. 3rd ed. Morgan Kaufmann.
3. Rich, E. and Knight, K. (2010). Artificial Intelligence. Tata McGraw-Hill Education.
4. Turing, A.M. (1950). Computing Machinery and Intelligence. Mind, 59(236), pp.433–460.
5. Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press.
6. Weizenbaum, J. (1966). ELIZA — a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), pp.36–45.

Khushi Mittal

Article

ACCELERATING DEVELOPER WORKFLOWS: THE ROLE OF VIM MOTIONS IN MODERN CODE EDITING ENVIRONMENTS

Abstract

In an era of rapid software development, efficiency is paramount. Vim motions enable a keyboard-based approach to navigating and manipulating text, reducing reliance on a mouse and eliminating the time required to reach for it and perform further actions. This paper explores the advantages of integrating Vim motions into modern development environments, from cognitive benefits to improved productivity. Additionally, the paper addresses some common adoption challenges and techniques to overcome them while easing into the learning curve, particularly for teams and individuals accustomed to mouse-driven and GUI-based workflows. The goal is to provide developers with a deeper understanding of Vim motions and encourage them to incorporate this powerful editing technique into their workflow for greater speed, precision, and control.

Introduction

Text editors have evolved from simple line editors to sophisticated tools that can be used for coding, writing documents, and publishing. While the advent of self-coding tools seems imminent, discussing methods to improve and enhance your coding experience is still worth it. The scope of this article is to emphasize why developers should integrate Vim Motions into their workflow.

The Cognitive and Efficiency Advantages of Vim Motions

A modal text editor has multiple modes of operation, and the keys' functions change depending on which mode it is currently in. Vi and Vim are the most popular instances of modal text editors. Modal editing helps you navigate code, manipulate text, and manage selections without taking your hands off the keyboard. Vim allows you to switch between Insert mode and Command Mode. The Insert mode allows you to add text, which becomes part of the document; essentially, write code. The command mode, on the other hand,allows you to manipulate the written text using keystrokes and commands and perform actions like copying, deleting lines, moving chunks of code, etc.

Modal editing aims to reduce cognitive load by minimising the need and eliminating the time wasted between the different input methods (the mouse and keyboard), allowing users to stay focussed on the tast at hand with a single input mode, thus reducing the mental and physical effort required to navigate and interact with the interface, leading to improved efficiency and flow. Vim motions prioritise keyboard commands, which can be executed quickly and efficiently, minimising the need to reach for the mouse.

FEATURES	CURSOR-BASED NAVIGATION	MOUSE-DRIVEN INTERACTIONS	VIM MOTIONS
Speed	Slow for large movements, requires multiple	Faster for selection, but requires hand	Extremely fast due to minimal keystrokes and
Precision	Requires repeated movement or holding keys	High precision for selection but imprecise for rapid	Very precise with words, sentences, paragraph, and
Editing Efficiency	Limited to moving and deleting characters or	Requires dragging and clicking; inefficient for	Combines navigation with text objects (ciw,
Learning Curve	Very low, easy to understand	Minimal learning required	High but rewards users with extreme efficiency

Integrating Vim Motions into Modern Development Workspaces

Vim was originally released in 1991 and was a decent editor for its time. Considering the advancements made in IDEs, using vanilla Vim in 2025 would not be the best experience. Unless, of course, you add plugins and tweak it to match your preferences, but that is beyond the scope of this paper. But we shouldn't lose heart since many modern code editors support Vim keybindings (modal editing) through extensions. In VS Code, the "vim" extension brings modal editing and motions commands to the editor. Sublime Text's "Vintage mode" replicates Vim's keybinding, and IntelliJ & JetBrains IDEs provide the IeaVim plugin to integrate Vim Motions into the developing environment.

It was reported that developers transitioning from standard cursor-based navigation to Vim motions experience a significant boost in productivity. Case studies also show reduced reliance on the mouse, hence faster navigation through larger codebases. Vim motions are also customizable as per your needs, allowing for a personal touch to the already extensive list.

In fast-paced development workflows, efficiency in navigating and editing code is crucial. Vim motions provide a streamlined approach to moving through large codebases, reducing dependency on a mouse or the arrow keys. Developers can quickly jump between functions, classes, and specific symbols, enabling faster debugging and implementation. Motions like w, b, ge, and gg allow for instant movement within files, while advanced commands such as [{,]} help in swiftly navigating code blocks.

For large-scale projects, Vim motions help developers reduce cognitive load by minimizing repetitive navigation patterns. When working across multiple files, commands likeb:bnext, :bprev, and :tag simplify switching between files and definitions, making them invaluable for teams handling microservices or monolithic architectures.

Automation through Vim macros and scripting further enhances efficiency. Macros (q to record, @ to execute) enable repetitive tasks to be automated, significantly reducing manual effort in formatting, refactoring, or inserting boilerplate code. Additionally, Vim scripts can be used to customize motions for specific workflows, such as automatically jumping to error logs or highlighting modified lines in a CI/CD pipeline.

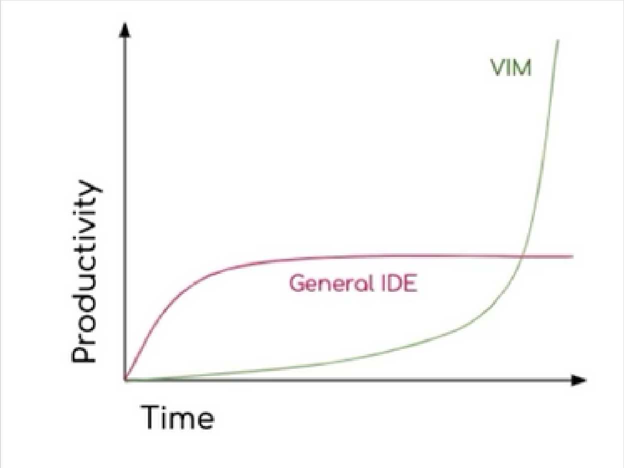
Learning Curve and Adoption Strategies

Considering you're now convinced to integrate Vim motions into your development workflow, doing the same presents both opportunities and challenges. Many teams rely on feature-rich IDEs with point-and-click interfaces, for whom the transition to Vim motions may seem daunting.

While Vim is efficient when it comes to editing and navigating, its adoption can be hindered by a steep learning curve and resistance from teams accustomed to cursor-based and mouse-heavy workflows.

Unlike traditional mouse-driven interfaces, Vim requires users to understand modes, motions, and composable commands. To battle the steep learning curve, interactive tutorials like "vimtutor", "cheat sheets", and plugins like "WhichKey" can guide new users through common motions, hence reducing the friction involved in the transition.

A fully Vim-based workflow isn't necessary for every developer. A hybrid approach, utilizing Vim motions within modern IDEs via plugins and settings (as discussed earlier), can be employed. This allows you to leverage all the debugging tools, Git integrations, and UI-based refactoring features.



By addressing these challenges with structured learning and hybrid integrations, teams and individuals can effectively incorporate Vim motions, unlocking faster, more efficient workflows in any development environment.

Conclusion

Vim motions prioritize speed, accuracy, and efficiency, providing a radically new method to traverse and edit code. Even while the learning curve would seem high at first, any developer would find the long-term productivity gains to be well worth the investment. Vim motions can be gradually incorporated into conventional workflows, even for those using GUI-based IDEs, to achieve a balance between efficiency and familiarity. Vim motions are a timeless skill that improves code editing in all situations, even as AI-powered tools and keyboard-driven development patterns continue to change. Vim motions can drastically change the way code is written and navigated, thus every developer should at least try them out.

References

-Robbins, Arnold. Learning the Vi and Vim Editors. O'Reilly Media, 2008.
-Moolenaar, Bram. Vim Documentation. Available at vim.org.
-Ramsay, Ben. Mastering Vim Quickly: From WTF to OMG in No Time. Independently published, 2017.
-Neil, Drew. Practical Vim: Edit Text at the Speed of Thought. Pragmatic Bookshelf, 2012.
-JetBrains Developer Ecosystem Report, 2023. "IDE Usage and Vim Keybindings in Modern Development."

ADITYA KRISHAN

Article

COGNITIVE DIGITAL TWINS: AI-AUGMENTED SIMULATIONS FOR INTELLIGENT DECISION-MAKING

Introduction

Decision-makers in a variety of sectors, including manufacturing and healthcare, find it difficult to comprehend the current state of systems, forecast future events, and optimize operations in an increasingly complex, data-driven world. Simulating high-dimensional, dynamic environments with uncertainty is a challenge for traditional modeling and analytics tools. Digital twins are virtual representations of physical systems that allow for simulation, monitoring, and control while mirroring real-time processes. However, digital twins must change along with industries.

Cognitive digital twins—intelligent digital copies improved by artificial intelligence (AI), specifically machine learning, natural language processing (NLP), and reinforcement learning—are the next development. These systems learn, reason, adapt, and make recommendations on their own—they do more than just simulate.

This article examines how cognitive digital twins are changing industries through improved personalization, decreased downtime, simulating high-risk situations, and facilitating smarter decision-making. We'll look at their design, AI application, case studies, difficulties, and potential paths forward.

WHAT ARE DIGITAL TWINS?

A digital twin is a real-time digital model of a system, process, or physical object. It mimics the current condition and actions of its real-world counterpart by combining data from sensors, operational logs, and control systems.

Condition monitoring, performance optimization, and predictive maintenance all make extensive use of digital twins. However, conventional twins are frequently passive, reactive models with low IQs. Cognitive abilities come into play here.

COGNITIVE DIGITAL TWINS

Artificial intelligence (AI) algorithms that offer learning, adaptation, prediction, and autonomous decision-making are used to improve cognitive digital twins (CDTs).

Essential skills:

- Contextual awareness: Recognize environmental factors and system dynamics
- Learning from data: Apply machine learning to update models based on past performance
- Reasoning and planning: Use AI for control, optimization, and inference
- Interaction: Use natural language processing (NLP) to communicate with people and provide instructions.
- Autonomy: Perform tasks or sound notifications without the need for human involvement.

Essentially, CDTs function as intelligent agents that use adaptive cognition to bridge the gap between the digital and physical worlds.

AI TECHNIQUES

- ·Using Machine Learning to Identify Anomalies: Patterns that diverge from typical system behaviour are identified by both supervised and unsupervised models. An ML model might, for instance, pick up on a turbine's vibration patterns and identify irregularities.
- ·Optimizing with Reinforcement Learning:Through simulation, RL agents discover the best control strategies. For instance, depending on demand and outage circumstances, a smart grid's CDT may learn how to reroute electricity.
- ·CDTs can communicate with people thanks to natural language processing, or NLP. When maintenance engineers ask, "What's the projected failure rate of motor #3?" they can get a precise response in natural language.
- · Diffusion models and GANs can enhance training data or produce possible future states. This is particularly helpful in situations (like pandemics) with little historical data.

APPLICATIONS

- · Industry and Manufacturing 4.0: Cyber-Physical Digital Twins (CDTs) are used in manufacturing to improve quality assurance through defect prediction, optimize assembly line throughput, and perform predictive maintenance. In order to lessen disruptions, they also aid in modeling supply chain resilience. Siemens, for instance, successfully reduces unplanned downtime by 40% by optimizing gas turbine performance with CDTs.
- Smart Cities: Real-time traffic flow optimization, energy consumption forecasting, and environmental hazard simulation are made possible by CDTs in smart cities. Urban planners can enhance emergency response plans and infrastructure with the aid of these systems. One prominent example is Singapore, which effectively manages emergency responses and simulates city-wide planning scenarios using a digital twin platform enhanced with AI.
- ·Medical Care: In the medical field, CDTs facilitate drug interaction modeling, ICU monitoring for early deterioration detection, and patient-specific simulations for treatment planning. Digital twins that mimic real-time organ functions and offer tailored intervention recommendations to enhance outcomes are being developed for intensive care unit patients by companies such as Philips and GE Healthcare.
- ·Defence and Aerospace: In the aerospace and defence industries, CDTs are used for mission rehearsal via simulation, flight path optimization, and predictive diagnostics of aircraft components. In order to improve mission resilience against unanticipated failures and enable fault prediction, NASA's Jet Propulsion Laboratory simulates spacecraft systems using AI-powered digital twins.
- ·Utilities and Energy: In the energy sector, CDTs model the integration of renewable energy sources into grids and help with load balancing, outage management, and predictive asset failure. In order to optimize energy output and guarantee operational efficiency, General Electric uses digital twins to manage wind farms by fusing weather data with AI forecasting.

CHALLENGES

1.Availability and Quality of Data

Clean, real-time data is necessary for high-fidelity simulations. CDT performance can be harmed by missing, noisy, or biased data. The digital twin might not accurately depict the actual condition of the physical system in the absence of reliable data streams. To ensure CDT reliability, data governance, validation procedures, and real-time monitoring are essential.

2. Complexity of the Model

It is computationally demanding and challenging to validate hybrid twins that combine models based on AI and physics. Over time, scaling and maintaining the models may become difficult as a result of this complexity. Continuous testing and improvement are necessary to guarantee that these hybrid systems maintain their accuracy in the face of changing circumstances.

3.Compatibility

It's still difficult to integrate cross-platform sensors, proprietary hardware, and legacy systems. The smooth communication between components is frequently complicated by the absence of standardized protocols. It might take specialized middleware and constant vendor cooperation to achieve true interoperability.

4. Privacy and Security

Because CDTs frequently manage physical processes, maintaining cybersecurity is essential. Physical harm or data leaks could result from a compromised CDT.

Vulnerabilities could be used by attackers to control simulations or actual operations. Strong encryption, access restrictions, and ongoing security audits are essential protections.

5. Trust and Explainability

Critical domains such as aerospace and healthcare require CDTs to provide decisions that can be explained. Stakeholders or regulators might not approve of black-box models. Building trust and adhering to compliance standards are facilitated by transparent reporting and decision logs. Additionally, explainability facilitates improved communication and quicker troubleshooting between AI systems and human operators.

FUTURE DIRECTIONS

- Twin Networks with Cognitive Functions

Networks of twins can coordinate across systems in place of separate CDTs; for example, a power plant CDT can work with a city's energy grid CDT to jointly optimize the distribution of electricity.

- Cognitive Twins with Edge Capabilities

Without depending on cloud connectivity, localized, offline decision-making is made possible by deploying lightweight AI models on edge devices.

- Digital Twins with Quantum Enhancement

Particularly for high-dimensional physical simulations, quantum computing may be able to resolve challenging simulation issues that traditional systems find difficult to handle.

- AI Ethics in CDTs

Fairness, accountability, and transparency must be ingrained in CDT decision-making, especially in areas like patient care or urban planning.

CONCLUSION

Cognitive Digital Twins are a potent tool for wise decision-making as businesses traverse complex, uncertain, and volatile environments. These systems become more than just digital copies when AI and real-time simulation are combined; they become proactive, flexible, and independent collaborators in complexity management.

CDTs are changing the future by modeling climate interventions, simulating medical treatments, and optimizing industrial processes. Although there are still issues with data integration, validation, and governance, the trend is clear: Cognitive Digital Twins are a strategic necessity as well as a technological advancement.

REFERENCES

1. Tao, F., et al. (2019). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access, 6, 3585–3593.
2. Grieves, M. (2014). Digital Twin: Manufacturing Excellence through Virtual Factory Replication. White Paper, University of Michigan.
3. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971.
4. Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology, 29, 36–52.

Aanshi Bansal

Article

MIGRATION TO POST-EXPLAINABLE AI (XAI): OPENING THE BLACK BOX OF MACHINE LEARNING

Abstract

Deep learning models' opaque decision-making raises serious ethical, regulatory, and trust issues as they proliferate in high-stakes industries like healthcare, finance, and law enforcement. This is addressed by Explainable Artificial Intelligence (XAI), which makes AI models interpretable without sacrificing efficiency. This article highlights the crucial role that XAI plays in the responsible deployment of AI by examining its principles, taxonomy, popular methods (LIME, SHAP, saliency), domain-specific use cases, evaluation methodologies, and future research directions.

Introduction

Critical decisions in fields like automated sentencing, loan approvals, and medical diagnosis are now driven by deep learning and sophisticated machine learning models. However, these models frequently operate as opaque "black boxes," preventing stakeholders and users from understanding the decision-making process.

Explainability has become imperative—not optional—for trusted, fair, and accountable AI systems. Explainable AI (XAI) has emerged to demystify these systems, bridging the gap between technical performance and human understanding [4]

.Why Explainability Matters

Interpretability is built into old-fashioned models such as decision trees and logistic regression, but modern AI models such as transformers and deep neural networks trade interpretability for more accuracy so that they are harder to trust in important decision-making scenarios.

Domains calling for transparency are:

·Healthcare: Diagnosis and treatment decisions

·Finance: Decision-making of the scoring of credit and loans

· Law: Predictive policing and law rulings

· Independent systems: car, and medical image diagnosis

Privacy regs like Europe's GDPR and features like "right to explanation," HIPAA and new AI regulations also require explanatory power over algorithmic results [2]

XAI Taxonomy & Frameworks

XAI research provides multiple taxonomies, discriminating between:

· Model-specific vs. model-agnostic methods.

· Post-hoc vs. intrinsically interpretable models.

· Local vs. global explanations

· User types: developers, domain experts, end users [4]

Barredo Arrieta etal. (2019) presents a detailed taxonomy of current approaches, and how they are applicable to different audiences and types of models [4].

Even more recent ones (2024–25) extend on this by assessing the reliability and domain-specific performance of explanations. [4][1].

Core Explainability Methods

1.LIME (Local Interpretable Model-agnostic Explanations)

Approximates a black box locally using an interpretable surrogate (often linear). Useful for tabular, image, and text data. However, explanations vary based on input perturbation and super pixel segmentation [4] [6]

2 SHAP (SHapley Additive exPlanations)

Based on game theory, SHAP gives feature contributions using Shapley values. It has provable guarantees of consistency and supports both local and global interpretations [3].

3. Gradient-Based Attribution (e.g., Grad-CAM, SmoothGrad, RISE)

Mostly used for vision tasks to see what the pixels are contributing towards the predictions of the model. Methods such as Grad-CAM focus on image regions activating predictions, while recent approaches such as RISE provide more robust and global coverage [9].

4 Hybrid and Emerging Methods

Recent studies recommend combining the LIME and SHAP methods (e.g., in autonomous driving) to improve interpretability and runtime performance [8].

Wang etal. (2024) give a comprehensive taxonomy and comparison between various gradient-based attribution techniques, with a focus on trade-offs and fidelity between explanation methods [2].

5. Comparative summary

Aspect	LIME	SHAP	Grad-CAM
Method Type	Model-agnostic, local	Model-agnostic, local + global	Model-specific, typically CNN-based
Explanation Basis	Local surrogate (e.g., linear model)	Shapley values from cooperative game	Gradients w.r.t. convolutional layers
Applicable Domains	Tabular, text, image	Tabular, text, image	Primarily image (computer vision)
Interpretability Scope	Local explanations (per prediction)	Local + global explanations	Visual saliency maps for individual
Stability/Consistency	Low (sensitive to perturbation and	High (mathematically	Medium (depends on layer and model
Computational Cost	Moderate	High (computationally	Low to moderate
Output Format	Feature importance scores	Additive feature contributions	Heatmap over image regions
Strengths	Fast and flexible across models	Theoretically grounded,	Highly intuitive for visual domains
Limitations	Sensitive to perturbations; less	Slower; assumes feature	Limited to vision models; harder to

Evaluation & Application Domains

A recent systematic review (2024) compared 512 XAI applications in multiple-domain healthcare, cyber- security, finance, environment, education, social care. SHAP and LIME were most commonly used methods; however, most of the studies did not have consistency in quantitative evaluation metrics [1].

In specialized areas:

· Alzheimer's stage prediction:Overviews support usefulness of SHAP/LIME in explaining clincial predictions and reinforcing the trust of end users in support systems [7].

· Vehicle XAI: Hybrid LIME–SHAP for real-time interpretability in autonomous vehicles and tradeoff between fidelity and performance [8].

· Finance & credit risk: LIME and SHAP can shed light on the prediction factors underlying the fraud detection and credit scoring, which can assist banks in explanations to both the clients and regulators [11] [4].

Challenges and Limitations

However, despite the common use of LIME and SHAP, the following problems remain:

· Both model-dependent and feature collinearity induce biased explanations, particularly for correlated features [3].

· Lack of fidelity and stability : LIME frequently has low recall or different importance over runs, SHAP is generally more stable, but more computation intensive as well [10].

·Regulatory & malfeasance considerations: Excessive sharing of rationales could open up attack vectors or invite adversarial exploration. Policy mechanisms require explainability but are not technically viable for black box LLMs [13]

Future Directions

1 Standardized Evaluation Frameworks

There is a pressing need for robust, quantitative metrics to assess explanation fidelity, interpretability, and user trust—moving beyond anecdotal or expert-based qualitative evaluation [1]

2 Causality-Aware and Concept-Based Methods

Future XAI research aims to move from correlative explanations toward causal reasoning models and concept bottleneck approaches, offering more intuitive insights [4].

3 Explainability for LLMs and Multi-Modal AI

Recent efforts by OpenAI and Anthropic focus on interpreting features and concept activations inside large language models—these approaches map internal representational neurons to human-understandable concepts to improve safety and trust [13].

4 Interactive & User-Centric Tools

Emerging techniques allow users to ask models "why" or "counterfactual-like" queries during inference—to make explanations dynamic, context-aware, and accessible to non-technical users.

Conclusion

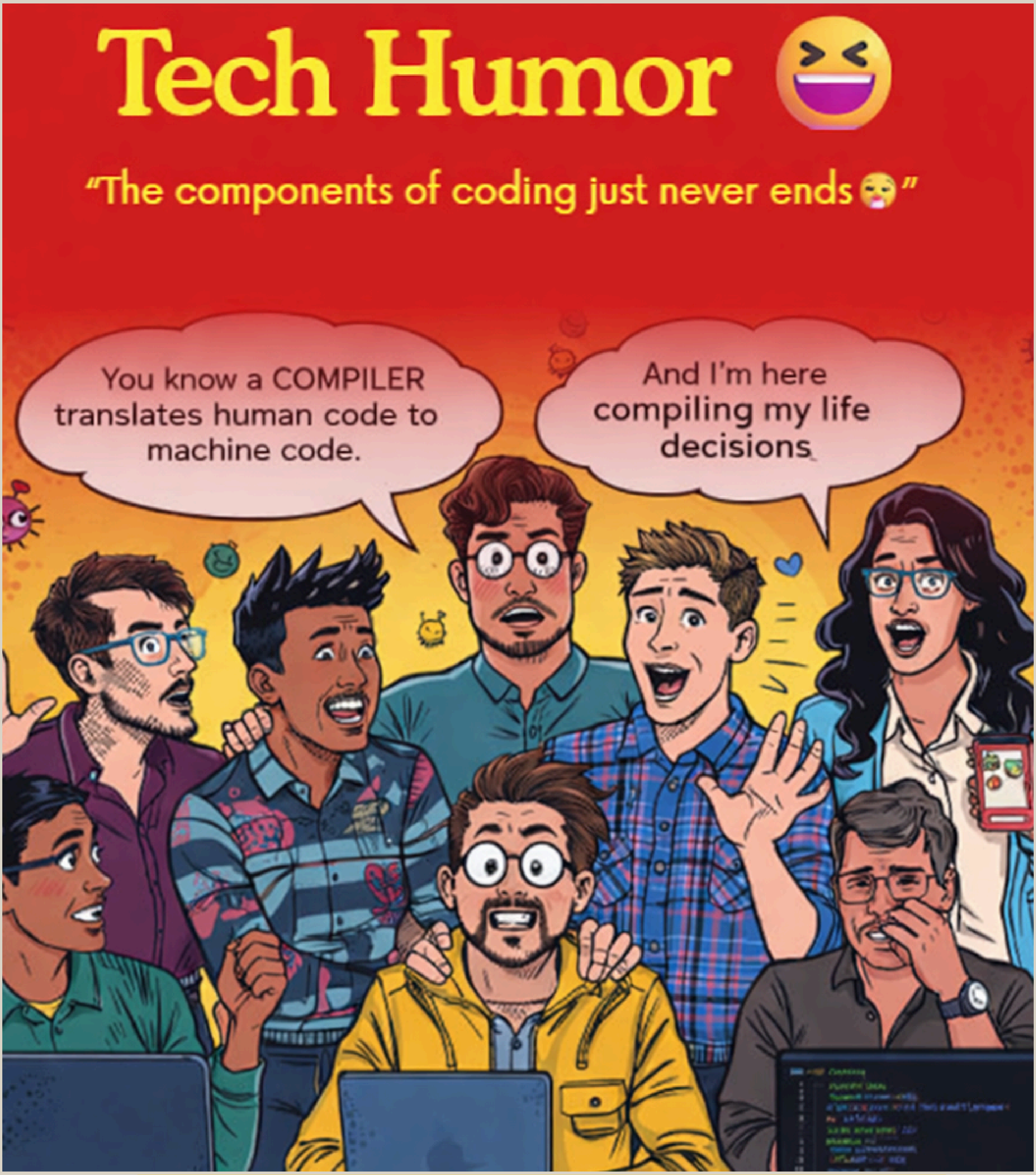
Explainable AI is crucial to the deployment of reliable AI systems in high-stakes domains. Although existing techniques like LIME, SHAP, Grad CAM, and hybrid methods offer useful explanations, they are not perfect—model choice-sensitive, domain context-sensitive, and dataset-specific. Multi-disciplinary advances, improved evaluation paradigms, and recent developments in concept-level and interactive XAI are mapping the future towards more transparent, dependable, and human-centered AI.

References

1. Shor, P. W. (1994). Algorithms for quantum Barredo Arrieta, A., et al. "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI." Information Fusion, Oct 2019.
2. Wang, Y., Zhang, T., Guo, X., Shen, Z. "Gradient Based Feature Attribution in Explainable AI: A Technical Review." arXiv (Mar 2024).
3. Salih, A., et al. "A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME." Advanced Intelligent Systems (2025).
4. University of Jyväskylä & Maribor. "Recent Applications of Explainable AI (XAI): A Systematic Literature Review." Appl. Sci. (Oct 2024).
5. Altukhi, Z. M., Pradhan, S., Aljohani, N. "A Systematic Literature Review of the Latest Advancements in XAI." Technologies (Mar 2025).
6. MDPI, "Explainable AI in Computer Vision: Grad-CAM, RISE, SmoothGrad" (2025).
7. "Systematic Review: LIME and SHAP in Alzheimer's Detection." Brain Informatics (Apr 2024).
8. Tahir, H. A., Alayed, W., Hassan, W.U., Haider, A. "Hybrid XAI in Autonomous Vehicles: LIME–SHAP Integration." Sensors (2024).
9. Wikipedia: "Explainable artificial intelligence" (Regulation, concept taxonomy) (2025).
10. "Evaluating post hoc explanations using transparent models." Int. J. Data Sci. Analytics (2023).

Harshita Sharma

FUN FACTS



#ACMFunPage #CoderLife

#CollegeChronicles

CREDITS

Editorial Mentor Board

Dr. Sunil K. Singh
(Mentor)
Professor and HoD
Department of CSE

Dr. Sudhakar Kumar
(co-mentor)
Professor
Department of CSE

Sahil Garg
CASC Student
Chairperson
(2024-2025)

Ayushi
CASC-W Student
Chairperson
(2024-2025)

Jaiveer Singh
CASC Student
Chairperson
(2025-2026)

Ritika Kalia
CASC-W Student
Chairperson
(2025-2026)

Lead Editors

Vanshika Singla
2023

Eshmeet Singh Bhachu
2023

Content Editors

Aarushi
2023

Aanshi Bansal
2023

Feature Editors

Khushi
2023

Bhavya
2023

Anshika Goyal
2024

Tanvi
2024

Shiven Sharma
2024

CASC Board

Jaiveer Singh
Chairperson

Satvik Pathak
Vice-Chairperson

Sanatan
Secretary

Shivam Vats
Membership Chair

Dhruv Bali
Treasurer

Abhay
Webmaster

Aarushi
Design Head

Kritin
External Member Head

Vanshika Singla
Editorial Head

Sahil Kumar
Social Media Manager

Maanit
PR Head

Aditya
Event Manager

Japjot
Domain Director(Web & DevOps)

Hitesh
Domain Director
(Competitive Programming)

Anshul
Domain Director
(Android)

Jasvir
Marketing Head

Jasjeet
Domain Director
(AI & Ml)

CASC-W Board

Ritika Kalia
Chairperson

Samriti Sharma
Vice-Chairperson

Simar Atwal
Secretary

Mehak Negi
Membership Chair

Khushi
Treasurer

Bhavya
Webmaster

Eshmeet Singh Bachu
Design Head

Ravina Mittal
Executive Member Head

Aanshi Bansal
Editorial Head

Bhumika Bijlwan
Social Media Manager

Harshita
PR Head

Sargun
Event Manager

Shreya
Domain Director(Web & DevOps)

Hitesh
Domain Director
(Competitive Programming)

Anshul
Domain Director
(Android)

Anshika Goyal
Marketing Head

Jasjeet
Domain Director
(AI & Ml)

Engineering is the closest thing to magic that exists in the world.

~ Elon Musk

- acmccet@gmail.com
- /acmccet
- https://ccet.acm.org
- CCET ACM Student Chapter
- /acmccet
- ccet-acm-student-chapter