DIGITAL OUTLET

VOLUME 5- ISSUE 3

MAY- JUNE 2024

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

TABLE OF CONTENTS

Cover Page	1
Index	2
Mentors	3
Team	4 - 5
Vision/Mission	6
Articles	7-14
Credits	15
Last Page	16

A NOTE FROM OUR MENTORS

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

Dr. Manpreet Singh
Principal CCET (Degree Wing)

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

Dr. Sunil K. Singh
Professor and HOD, CSE | Faculty Mentor

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

Dr. Sudhakar Kumar
Assistant Professor, CSE | Faculty Sponsor

CCET ACM Student chapter is a group of people with similar interests and goals in computer science. Together, this platform focuses on the growth and development at not only personal but professional level also as it has a unique learning environment.

Saket Sarin
UG Scholar, 5th Semester, CSE | Chairperson, CASC

ACM-W Student Chapter of CCET aims to promote women in technology. As a member of this community, you will have the opportunity to collaborate with others who share similar interests and explore different areas of computing in order to advance in them.

Aishita
UG Scholar, 5th Semester, CSE | Chairperson, CASC-W

CCET ACM STUDENT CHAPTER

Research and Development

Student Speaker Program

Competitive Coding

Designing & Digital Art

Internship and Career Opportunity

ABOUT ACM

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 fields. 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 field, so we started an ACM-W Chapter to increase the morale of women. CASC launched an app which aimed at maintaining decorum of reading among CS members and sharing their ideas.

CCET ACM-W STUDENT CHAPTER

- Research and Development
- Student Speaker Program
- Competitive Coding
- Designing & Digital Art
- Internship and Career Opportunity

ABOUT ACM-W

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 community 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 professionally. 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.

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/scholars, 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

ADVANCING TRANSPARENCY AND TRUST THROUGH EXPLAINABLE AI (XAI)

Divyansh Manro [CSE 2022]

Introduction:
Explainable Artificial Intelligence (XAI) has become essential in addressing the increasing use of machine learning (ML) models in different industries due to the significant hurdles caused by the lack of transparency in decision-making processes. AI models, particularly deep learning architectures, can handle large data sets and generate highly precise predictions. Yet, their intrinsic complexity, commonly known as the "black box" nature, has raised concerns about their clarity and reliability. In industries like healthcare, finance, and autonomous systems, it is crucial to understand how AI algorithms reach decisions in order to establish trust and accountability, keeping in mind their significant real-world consequences. This article explores the evolution of XAI techniques such as Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), and how they contribute to increasing transparency in AI, tackling ethical issues, and promoting broader adoption of AI.

XAI Techniques and their Uses:
The complexity of recent AI models has resulted in the creation of advanced techniques such as LIME and SHAP, which offer insights into model predictions while maintaining their precision. LIME operates by tweaking input data to see how small alterations impact model predictions, approximating the behaviour of a complicated model with a more straightforward, understandable one near a particular instance. This method of interpretability at a local level helps users understand the reason behind a specific decision, even if the overall model is still opaque. On the contrary, SHAP uses cooperative game theory to allocate a contribution value to each feature depending on its significance to the model's output. SHAP provides a reliable and theoretically supported explanation for feature importance through the calculation of Shapley values, making it a valuable tool in a diverse range of AI applications. The mechanisms of LIME and SHAP are demonstrated in Fig 1(a) and Fig 1(b) respectively.

XAI methodologies are being used in vital sectors, proving their significance in guaranteeing transparency and dependability in AI systems. In the field of healthcare, the importance of being able to provide explanations for why a specific diagnosis or treatment plan is recommended is crucial, especially as AI becomes more prevalent in areas such as diagnostics, treatment suggestions, and personalized medicine. XAI builds trust with clinicians and detects biases in the model, enhancing patient safety. LIME can help explain the factors affecting a diagnosis, like medical history, symptoms, or imaging data, while SHAP reveals the importance assigned to these factors, giving a clear understanding of the decision-making process.

Transparency is crucial in the financial industry for adhering to regulations, managing risks, and identifying fraudulent activities. Algorithmic trading systems, using AI for rapid decision-making, need to be understandable to prevent manipulation in the market and maintain accountability. XAI methods like LIME and SHAP can shed light on the impact of specific market indicators or past patterns on a given trade, enabling regulators and stakeholders to understand the justification behind decisions made by AI. Likewise, explainable models in credit risk assessment can uncover the factors impacting a borrower's risk score, guaranteeing impartial and equitable decision-making. XAI also aids financial institutions in meeting regulations that require transparency in AI systems, lowering the chance of legal action and improving stakeholder trust.

In the realm of autonomous systems such as self-driving vehicles and robotics, XAI is crucial. Autonomous systems commonly have complicated decision-making procedures that rely on live data from sensors, cameras, and GPS. It is essential for safety and public trust that these decisions are made transparent. XAI techniques can provide explanations for why an autonomous vehicle performed a specific action, such as braking or changing lanes, by identifying the input features that played a role in the decision. XAI in robotics can offer explanations for tasks like object manipulation or navigation, helping engineers and users comprehend the system's actions. Having transparency is crucial for debugging, enhancing system performance, and guaranteeing the safe operation of autonomous systems in real-world situations.

Challenges in XAI:

Even with the notable progress in XAI, a major challenge still remains in balancing model accuracy with interpretability. Complex models like deep neural networks and ensemble methods frequently deliver strong accuracy levels, but their interpretation can be challenging, leading users to doubt their decisions. However, simpler models such as decision trees and

linear regression are easier to understand but may not have the precision required for complex tasks. Finding the perfect equilibrium between these two opposites is essential for the effective implementation of XAI in practical scenarios. Methods like simplifying models by removing unnecessary features and providing post-hoc explanations like LIME and SHAP help manage this balance by offering localized interpretations of complex models without significantly reducing accuracy.
Another difficulty lies in the ability of XAI solutions to grow in size. As AI systems become more advanced and are used with bigger sets of data, providing instant explanations for model predictions becomes costly in terms of computation. This is especially troublesome in areas like self-driving cars and financial markets, where quick decisions are crucial. Improving XAI algorithms to be more efficient on extensive datasets and work in real-time is a crucial focus of ongoing research. Moreover, it is a challenge to guarantee that explanations are strong and applicable to various datasets and situations, since localized explanations may not always reflect the overall performance of the model.

Ethical considerations are also vital in the advancement and implementation of XAI. AI systems are frequently utilized in critical areas that require fairness, accountability, and transparency. XAI can assist in tackling issues regarding bias and discrimination by offering transparent explanations of decision-making processes and pinpointing potential sources of bias in the training data or model architecture. This is especially crucial in industries such as finance and healthcare, as prejudiced AI models can greatly impact marginalized groups. Moreover, XAI can support regulatory compliance by guaranteeing that AI systems adhere to legal requirements for transparency and fairness, especially in areas with strict data protection laws like the European Union's General Data Protection Regulation (GDPR).

Future Prospects in XAI:
In the future, the importance of XAI will grow as AI systems become more advanced and integrated into daily routines. The latest developments in XAI research involve incorporating cognitive science principles to generate explanations that are more human-centric and in line with how individuals perceive information. This requires creating AI models that not just offer explanations but do it in a manner that is easy to grasp by non-professionals. A potentially promising research area is the creation of interactive visualization tools that enable users to dynamically explore AI decisions. These resources can assist individuals in comprehending the real-time decision-making process, offering more in-depth understanding of how AI systems operate.

In conclusion, Explainable AI is a significant progression in the artificial intelligence sector, meeting the increasing need for openness, reliability, and responsibility in AI platforms. XAI helps different industries' stakeholders better comprehend, have confidence in, and make use of AI technologies by enhancing the interpretability of complex models. Nevertheless, issues concerning scalability, balancing accuracy with interpretability, and ethical concerns persist. As XAI research progresses, it will become more crucial in making sure AI systems are not only strong but also transparent, fair, and in line with human values. This will be vital for the extensive acceptance of AI technologies in crucial sectors, ultimately advancing the field of artificial intelligence entirely.

MOJO PROGRAMMING LANGUAGE: REVOLUTIONIZING AI AND HIGH-PERFORMANCE COMPUTING

Sahil Garg[CSE 2022]

In recent years, programming languages have evolved to meet the growing demands of diverse domains, including artificial intelligence (AI), high-performance computing (HPC), and machine learning (ML). One of the emerging stars in this space is Mojo, a new programming language designed to address the performance and productivity challenges faced by developers working in these fields. In this article, we'll explore what Mojo is, its key features, and why it is becoming a game-changer in AI and HPC.

What is Mojo?

Mojo is a modern, high-performance programming language that combines the ease of use of Python with the efficiency of low-level languages like C++ and Rust. It is specifically tailored for applications that require massive computational power, such as AI, ML, and data-intensive tasks. The language is designed to be accessible to both beginner programmers and seasoned developers, making it ideal for rapid prototyping and high-performance computing alike.

Mojo was created by Modular, a company founded by Chris Lattner, who also designed LLVM and Swift, which makes the language inherently innovative and purpose-driven. It provides a single environment where developers can write code that performs well on both CPUs and specialized hardware like GPUs and TPUs, which is critical in today's AI ecosystem.

Key Features of Mojo

Python Compatibility Mojo is deeply integrated with Python, allowing developers to use Python code, libraries, and tools seamlessly within Mojo projects. This compatibility makes it an easy

transition for Python developers, who can enjoy the benefits of Mojo's speed and efficiency while continuing to use their familiar workflows.

Performance-Oriented Mojo is designed to perform at a level that rivals low-level languages. It achieves this through automatic parallelization, hardware-aware optimizations, and advanced compilation techniques. The language allows developers to write high-level code that the Mojo compiler transforms into optimized machine code, making it suitable for demanding applications like deep learning and scientific computing.

Multicore and Heterogeneous Computing Mojo provides built-in support for multicore processing and heterogeneous computing environments. This means that developers can write code that scales efficiently across CPUs, GPUs, and specialized accelerators without requiring manual hardware-specific optimizations. This capability is especially crucial for AI and ML workloads, where different types of hardware are often used together to achieve better performance.

Memory Safety Like Rust, Mojo emphasizes memory safety, reducing the risk of common programming errors like null pointer dereferencing, buffer overflows, and data races. This ensures that even in performance-critical applications, developers can avoid subtle bugs that may arise from manual memory management.

Metaprogramming and Optimization Mojo features metaprogramming capabilities, allowing developers to write code that can generate other code, thus automating repetitive tasks and enabling complex optimizations at compile time. The language also supports advanced type inference and compile-time evaluation, ensuring that high-level abstractions don't come at the cost of performance.

Hardware-Aware Programming Mojo includes built-in support for hardware-specific optimizations. Developers can write code that is aware of the underlying hardware and can take full advantage of its capabilities, such as vectorized operations or custom instruction sets. This level of control is essential for maximizing the performance of AI and HPC workloads.

Extensibility and Modularity Mojo is designed to be extensible, meaning that developers can write their own domain-specific optimizations and tools to further improve the performance of their applications. Its modular design allows it to integrate easily with other languages, tools, and libraries, making it a flexible choice for diverse programming environments.

Mojo for AI and Machine Learning One of Mojo's standout applications is in the field of AI and ML, where performance and scalability are paramount. AI workloads, especially deep learning, require vast computational resources to train models on large datasets. Mojo's automatic parallelization, hard-

ware-aware optimizations, and GPU/TPU support make it an excellent choice for these tasks.

In contrast to Python, which often relies on external libraries like TensorFlow or PyTorch to handle the heavy lifting, Mojo is designed to provide high-performance execution natively. This can lead to significant performance improvements, reducing the time it takes to train models or perform inference on large datasets.
Moreover, Mojo allows developers to fine-tune their AI algorithms with greater precision and efficiency than traditional Python-based environments. By providing direct access to hardware-level instructions, developers can create models that run faster and use fewer resources, an essential advantage as AI systems become more complex and data-hungry.

Mojo's Impact on High-Performance Computing

Beyond AI, Mojo's performance features make it a powerful tool for a wide range of HPC applications, including scientific simulations, data analysis, and real-time computing. Traditional HPC languages like Fortran and C++ offer excellent performance, but they can be difficult to work with, especially when it comes to modern software engineering practices like modularity and code reuse.

Mojo addresses these concerns by offering the high-level abstractions that make languages like Python popular, while still providing low-level control when needed. This hybrid approach makes it easier for HPC developers to write scalable, efficient code without sacrificing productivity. Additionally, Mojo's focus on multicore and heterogeneous computing means that HPC applications can be optimized to run on the latest hardware, including GPUs and specialized accelerators.
Why Mojo Stands Out
Mojo's combination of ease of use, performance, and flexibility sets it apart from other languages that are either too low-level for rapid development or too high-level to achieve maximum performance. Its close integration with Python also makes it attractive for the vast number of developers who are already proficient in Python but need more performance for their applications.
In short, Mojo aims to fill the gap between high-level, productivity-focused languages like Python and performance-centric languages like C++ and Rust. By doing so, it promises to democratize access to high-performance computing, making it easier for developers to build next-generation AI, ML, and HPC applications without having to compromise on speed or productivity.

Conclusion

The Mojo programming language is set to become a cornerstone for developers

working in AI, machine learning, and high-performance computing. By combining Python's simplicity with the power and efficiency of low-level languages, Mojo opens up new possibilities for creating performant, scalable, and efficient software. Whether you're building AI models, optimizing hardware for HPC, or working on cutting-edge data science projects, Mojo offers a unified solution that bridges the gap between productivity and performance.

As the language matures, it's likely we will see Mojo playing an increasingly important role in both academic research and industry, empowering developers to tackle the most challenging problems of our time with greater speed and efficiency.

Credits

Editorial Mentor Board

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

Dr. Sudhakar Kumar
(Co-Mentor)
Assistant Professor
Department of CSE

Saket Sarin
CASC Student Chairperson
(2023 - 2024)

Aishita
CASC-W Student Chairperson
(2023 - 2024)

Akash Sharma
CASC Student Chairperson
(2022 - 2023)

Anureet Chhabra
CASC-W Student Chairperson
(2022 - 2023)

Lead Editors

Kanishk Nagpal
CSE 2021

Japan Ajit Singh
CSE 2021

Content Editor

Manya
CSE 2021

Ayushi
CSE 2022

Feature Editors

Ayushi
CSE 2022

Saksham Arora
CSE 2022

CASC Board

Saket Sarin
Chairperson

Kanishk Nagpal
Vice Chair

Shivam Goyal
Secretary

Saksham Arora
Membership Chair

Kartik
Treasurer

Tushar Singh
Webmaster

Japan Ajit Singh
Design Head

Palvasha Bansal
External PR Head

Manya
Editorial Head

Utkarsh Chauhan
Executive Head

Briti Singla
Social Media Mnager

Vanshika Chilkoti
Event Manager

CASC-W Board

Aishita
Chairperson

Mehak Preet
Vice Chair

Vanshika Bhardwaj
Secretary

Sahil Garg
Membership Chair

Harkiran Kaur
Treasurer

Ruchika Thakur
Webmaster

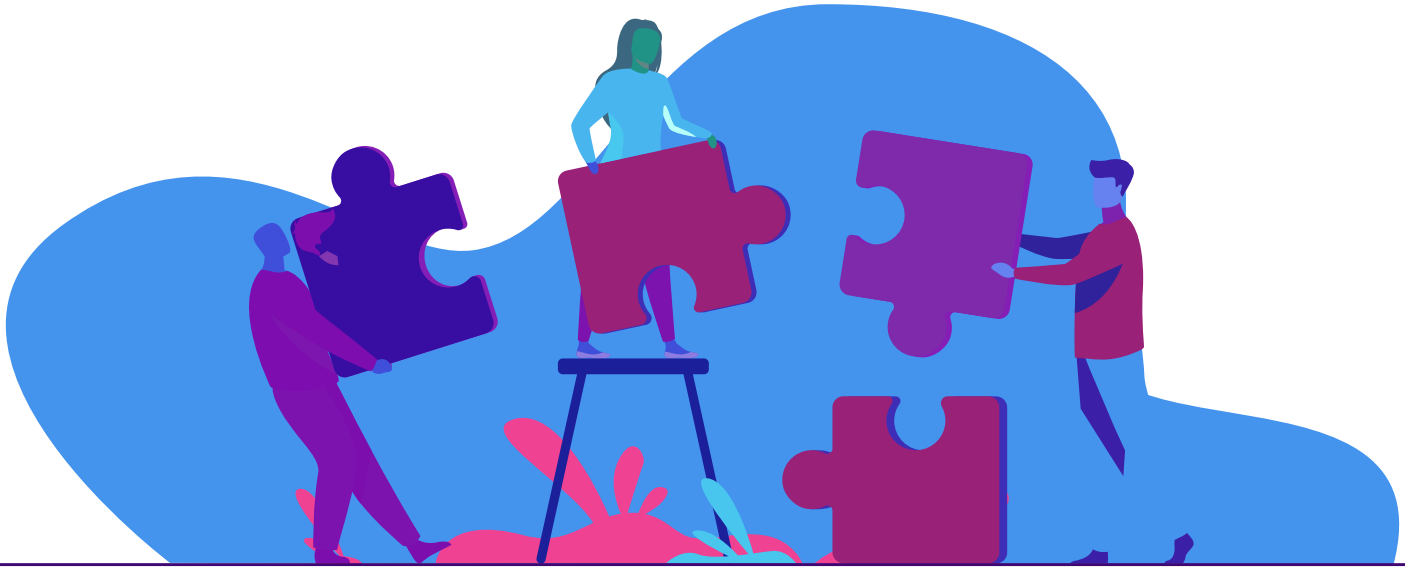
Priyanshu
Design Head

Ritika Gupta
External PR Head

Ayushi
Editorial Head

Avneet Kaur
Social Media Manager

Simran Jaggi
Event Manager



"Scientists explore the mysteries of what exists, while engineers bring to life what once only existed in dreams."

Theodore von Kármán
Pioneering Aerospace Engineer and Physicist

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

CCET Details
Department of CSE
CCET, Degree Wing
Sector - 26, Chandigarh

Contact Us
For general submissions
and feedback, contact us.
Website: www.ccet.ac.in