



Hari Rohit Bhavsar
Computer Science & Engineering
Indian Institute of Technology Bombay

22B0934
B.Tech.
Gender: Male
DOB: 28/12/2004

Examination	University	Institute	Year	CPI / %
Graduation	IIT Bombay	IIT Bombay	2026	9.27
Intermediate	Maharashtra HSC	Pace Junior Science College	2022	92.17%
Matriculation	ICSE	Lilavatibai Podar High School	2020	97.83%

Pursuing **Honors** in **Computer Science and Engineering**

SCHOLASTIC ACHIEVEMENTS

- Achieved an **AP** grade in Advanced Computer Architecture, given to the top 2 out of 105 students [2024]
- Achieved an All India Rank of **461** in the JEE Advanced exam among over **160,000** candidates [2022]
- Performed within the top **10** students from the institute in the **Trade-a-thon** conducted by Optiver [2024]
- Received **99.96** percentile in the Maharashtra Common Entrance Test among **600,000+** candidates [2022]
- Performed within **top 1%** of the Class XII Boards and was awarded scholarship under **INSPIRE** [2022]
- Secured an All India Rank of **24** in the **IISER** Aptitude Test out of over **22,000** candidates [2022]

INTERNSHIP EXPERIENCE

Tracking Dependencies for Memoized Functions

[May 2025 - July 2025]

D.E. Shaw India

- Generated and stored the dependencies of **disk** and **RAM memoized** python functions using python's **coverage** module for code-level dependencies and linux's **strace** tool for file-level dependencies
- Identified the files directly modified by a **git commit** and generated **abstract syntax trees** for python files to extract the specific code classes and functions that had been affected by the commit
- Used the **stored dependencies** along with the list of code components and files modified by a git commit to update the values of affected **memoized** python functions, thus removing stale values present on disk
- Added **multiple modes** for ease of use and successfully tested the tool on a production use case

RESEARCH EXPERIENCE

Countering Cache-Based Side Channel Attacks

[May 2024 - December 2024]

Principal Instructor: Professor Biswabandan Panda

Accepted to USENIX Security 2025

- Investigated **conflict based** cache side channel attacks that exploit the set associativity of Last Level Caches to obtain **hidden information** from victim processes and studied designs to mitigate these attacks
- Researched **MIRAGE**, a cache design which decouples **tag storage** and **data storage**, thus providing the **performance** of set-associative caches along with the additional **security** of fully-associative caches
- Conducted **occupancy** attacks on **MIRAGE**, to predict the memory utilization of a victim process which is then used for **website fingerprinting**, thus proving that MIRAGE is not immune to such attacks

Cache Hierarchy for Graph Workloads

[December 2024 - Present]

Principal Instructor: Professor Biswabandan Panda

- Extended **valgrind** to model a third level of cache revealing that common graph algorithms such as sssp and page-rank on average observe 80% more **last level cache (LLC) misses** than other algorithms
- Observed the **power law** of graphs and that 80% of LLC accesses came from the top 20% of vertices having the most edges ("hot vertices") suggesting that retaining these vertices could boost performance
- Discovered that cache blocks containing hot vertices obtain more **level two cache hits** and used this to build a replacement policy which obtained a speedup of **14.6%** while requiring **4KB** less space

KEY PROJECTS

Hypervisor Design and Linux Kernel Modules | *Course Project: CS695*

[Spring 2024-25]

Topics in Virtualization and Cloud Computing

Guide : Professor Purushottam Kulkarni

- Used the **IOCTL** syscall to create a **device driver** that provided interfaces for reassigning child and parent processes, given process IDs, as well as one to terminate all child processes of a given process
- Utilized the Linux KVM to create a **Paravirtualized Guest Operating System** with separate modes for memory management, and with **hypercalls** to perform memory translations and file IO handling
- Implemented a method to keep track of process stats using the **procfs** and **sysfs** Linux Kernel Modules

- Implemented a **scanner** and a **parser** for a simplified version of C containing arithmetic operations, function calls and conditional statements, using an LALR(1) parser from **bison** and **flex** for scanning
- Conducted **semantic analysis** of the language by constructing an **abstract syntax tree** for type-checking and generating **three address code** for the language, ensuring the language was strongly typed
- Ensured support for **global variables** by using a multi-layered symbol table for book-keeping of variables

- Implemented a lossy image compression framework for grayscale images by computing the **Discrete Cosine Transforms** of the quantized images, then using Huffman Encoding and RLE for lossless compression
- Incoorporated downsampling in the chrominance components of coloured images and provided APIs to specify the quality factor from 0 to 255, achieving a compression ratio of **6:1** at a quality factor of 10

- Processed **order books** received via **sockets** in real time, **estimated market prices** and made efficient algorithms for detecting **arbitrage opportunities** in markets by **pruning** quantities of orders placed
- Simulated a market using **multithreading** in C++ to receive and execute orders from several traders concurrently and developed strategies such as **median trading** and **statistical arbitrage** to make profit

- Used **Principal Component Analysis** to greatly reduce the dimensionality of a dataset of handwritten characters, preserving only the **essential features** of the dataset and thus allowing for faster computation
- Implemented a **Recurrent Neural Network** from scratch using numpy with **backpropogation** and **forward pass** and tested it to predict the next character in a given text sequence with over **90%** accuracy

- Created an online hybrid game based on Minesweeper and Cricket using **HTML**, **CSS** and **Javascript**
- Implemented intricate game mechanics such as **wickets**, **multiple runs**, and **powerups** using JavaScript
- Gave players the ability to customize the **grid size** and the **number of wickets** using CSS through JS
- Implemented mechanics such as **wide balls**, **no balls**, and **free hits** using random numbers in javascript

- Created intricate game mechanics such as **gravity** and **collision** detection using **SDL2** and **C++**
- Implemented a five stage platformer game with various different obstacles such as **cannons** and **lasers**
- Rendered **images** and **sprites** onto various elements of the game while maintaining **high framerates**
- Created interactive objects such as **collectable coins** and timer based events like **locked buttons**

POSITIONS OF RESPONSIBILITY

- Taught a group of **22** people about the fundamentals of **parallel programming**, such as **SIMD** instructions and **multi-threading** as well as how one could exploit these techniques for performance gains
- Released content **weekly** and kept meetings to ensure that mentees were able to cope with the workload
- Guided mentees through learning the **Open Computing Language** in order to code on the **GPU**
- Provided mentees with naive implementations of a **SAT Solver** and a **Machine Learning** library, which they were able to significantly optimize using the aforementioned techniques to get large performance gains

TECHNICAL SKILLS

Languages	Python, C, C++, Java, Javascript, MIPS, x86, Bash, SED, AWK, VHDL, Verilog
Data Science	NumPy, Matplotlib, OpenCL, Pandas, Pytorch
Software	MATLAB, Git, Fusion 360, Jupyter, $\LaTeX$, Doxygen, GDB, make

EXTRA-CURRICULAR ACTIVITIES

- Represented Hostel 6 in the prestigious **English Creative Writing General Championship**
- Secured **first** place and was awarded a certificate of merit in an inter-school quiz competition
- Obtained **third** place in an inter-school mono-acting competition and was given a merit certificate