INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

SEMESTER I						
Sl. No.	**Subject Code**	**Subject Name**	**L**	**T**	**P**	**C**
1.	CS5101	Design and Analysis of Algorithms	3	1	0	4
2.	CS5102	Foundations of Computer Systems	3	0	0	3
3.	CS61XX	DE-I	3	0	0	3
4.	CS61XX	DE-II	3	0	0	3
5.	HS51PQ	Technical Writing and Soft Skill	1	2	2	4
6	CS5103	Computing Lab-1	0	1	2	2
7	CS6109	IDE-I (Drone Data Processing & Analysis)	3	0	0	3
	TOTAL					**22**

SEMESTER II						
Sl. No.	**Subject Code**	**Subject Name**	**L**	**T**	**P**	**C**
1.	CS5201	Advanced Artificial Intelligence	3	0	0	3
2.	CS5202	Theoretical Computer Science	3	0	0	3
3.	CS62XX	DE-III	3	0	0	3
4.	CS62XX	DE-IV	3	0	0	3
5.	CS5204	Computing Lab-2	0	1	2	2
6	CS5205	Advanced Artificial Intelligence Lab	0	1	2	2
7.	RM6201	Research Methodology	3	1	0	4
8.	IK6201	IKS	3	0	0	3
	TOTAL					**23**

SEMESTER III						
Sl. No.	**Subject Code**	**Subject Name**	**L**	**T**	**P**	**C**
1.	CS6198	Summer Internship/Mini Project*	0	0	12	3
2.	CS6199	Project I	0	0	30	15
	TOTAL					**18**

* Evaluation of Summer Internship done between 2nd & 3rd Semester

Note: Summer internship (*) period of at least 60 days duration begins in the intervening vacation between semester 6 and 7 that may be done in industry/R & D /Academic institutions including IIT Patna. The evaluation would comprise combined grading based on host supervisor evaluation, project internship report after plagiarism check, and presentation evaluation by the parent department with equal weightage of each component.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

SEMESTER IV						
Sl. No.	**Subject Code**	**Subject Name**	**L**	**T**	**P**	**C**
1.	CS6299	Project II	0	0	42	21
	TOTAL					**21**

Total credit: 84

M.Tech CS Elective List

Bucket -1	**Semester -1 Subject List for DE-1**
Course Code	Course Name
CS6121	Advanced Blockchain Technology
CS6122	Advanced Cyber Security
CS6123	Advanced Pattern Recognition
CS6124	Formal Methods in Program Analysis and Verification
CS6125	Cyber Physical Systems

Bucket -2	**Semester -1 Subject List for DE-2**
Course Code	Course Name
CS6127	Advanced Cloud Computing
CS6128	Advanced Edge Computing
CS6129	Advanced Computational Data Analysis
CS6131	Reinforcement Learning
CS6132	Advanced Graph Machine Learning
CS6133	Advanced Time Series Analysis

Bucket -3	**Semester -2 Subject List for DE-3**
Course Code	Course Name
CS6221	Artificial Internet of Things
CS6222	Game Theory
CS6223	Quantum Machine Learning
CS6224	Text Mining & Analytics
CS6225	Knowledge Distillation

Bucket -4	**Semester -2 Subject List for DE-4**
Course Code	Course Name
CS6226	Quantum Cyber Security
CS6228	Selected Topics in Wireless Networks
CS6229	Advanced Big Data Analytics
CS6231	Selected Topics in Cryptography
CS6232	High Performance Computing
CS6233	Selective Topics in Generative AI

IDE from CSE							
	Semester	Course Code	Course Name	L	T	P	C
IDE-I	Semester-I	CS6109	Drone Data Processing & Analysis	3	0	0	3

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course number	CS5101
Course Credit	3-1-0-4
Course Title	**Design and Analysis of Algorithms**
Learning Mode	offline
Learning Objectives	The objective of this course is to equip students with a solid understanding of data structures and algorithms, enabling them to design, analyze, and implement efficient algorithms to solve complex computational problems. The course covers fundamental topics such as data structures, complexity analysis, sorting and searching techniques, problem-solving strategies, graph algorithms. By the end of the course, students will have developed the skills to critically analyze algorithm efficiency and apply advanced algorithms in practical scenarios.
Course Description	This course will provide understanding of aadvanced methods to solve problems on computers. It will also provide an overview to analyze those theoretically.
Course Outline	Fibonacci heap, unionfind, splay trees. Amortized complexity analysis Randomized algorithms Reducibility between problems and NPcompleteness: discussion of different NP-complete problems like satisfiability, clique, vertex cover, independent set, Hamiltonian cycle, TSP, knapsack, set cover, bin packing, etc. Backtracking, branch and bound Approximation algorithms: Constant ratio approximation algorithms. Application areas(i)Geometric algorithms: convex hulls, nearest neighbor, Voronoi diagram, etc.(ii)Algebraic and number-theoretic algorithms: FFT, primality testing, etc.(iii)Graph algorithms: network flows, matching, etc.(iv)Optimization techniques: linear programming
Learning Outcome	By the end of this course, students will be able to solve problems that are computationally intractable
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Mark Allen Weiss, "Data Structures and Algorithms in C++", Addison Wesley, 2003.
- Adam Drozdek, "Data Structures and Algorithms in C++", Brooks and Cole, 2001.
- Aho, Hopcroft and Ullmann, "Data structures and Algorithm", Addison Welsey, 1984.
- Introduction to Algorithms Book by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen
- Sanjoy Dasgupta, Christos H. Papadimitriou and Umesh V. Vazirani, Algorithms, Tata McGraw-Hill, 2008.
- Steven Skiena, The Algorithm Design Manual, Springer
- Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
- Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
- Udi Manber, Algorithms – A Creative Approach, Addison-Wesley, Reading, MA, 1989.
- Tim Roughgarden, Algorithms Illuminated

Course Number	CS5102
Course Credit	3-0-0-3
Course Title	**Foundations of Computer Systems**
Learning Mode	Offline
Learning Objective	The objective of the course is to provide a conceptual and theoretical understanding of computer architecture and operating systems.
Course Description	Foundations of computer systems is a review of two fundamental subjects of computer science viz., computer architecture and operating systems.
Course Outline	Computer architecture**:** Performance measures, Memory Location and Operations, Addressing Modes, Instruction Set, A Simple Machine, Instruction Mnemonics and Syntax, Machine Language Program, Assembly Language Program with examples. Processing Unit Design: Registers, Datapath, CPU instruction cycle, Instructions and Micro-operations in different bus architectures, Interrupt handling, Control Unit Design: Control signals, Hardwired Control unit design, Microprogram Control unit design. Pipelining and parallel processing, Pipeline performance measure, pipeline architecture, pipeline stall (due to instruction dependancy and data dependancy), Methods to reduce pipeline stall. RISC and CISC paradigms, I/O Transfer techniques, Memory organization: hierarchical memory systems, cache memories, virtual memory. Operating systems: Process states, PCB, Fork, exec system call, Threads, Process scheduling, Concurrent processes, Monitors, Process Synchronization, Producer Consumer Problem, Critical section, semaphore, Various process synchronization problems. Deadlock, Resource Allocation Graph, Deadlock prevention, Deadlock Avoidance: Banker's Algorithm and Safety Algorithm.

	Memory management techniques, Allocation techniques, Paging, Page Replacement Algorithms, Numericals.
Learning Outcome	This course will revisit two fundamental subjects of computer science viz., computer architecture and operating systems, thereby enabling the students to pursue more advanced problems in computer science based on these topics.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, 7th Ed, John Wiley and Sons, 2004.
- M. Singhal and N. Shivratri, Advanced Concepts in Operating Systems, McGraw Hill, 1994.
- David A Patterson and John L Hennessy, Computer Organisation and Design: The Hardware/Software Interface, Morgan Kaufmann, 1994. ISBN 1-55860-281-X.

Course Number	CS5103
Course Credit	0-1-2-2
Course Title	**Computing Lab-1**
Learning Mode	Offline
Learning Objective	The course aims to develop students' analytical and practical skills in designing efficient algorithms and understanding the complexities of operating systems. Students will learn to analyze the efficiency of algorithms, understand various algorithmic strategies, and implement them to solve complex problems. In the Operating Systems segment, students will explore the core concepts, including process management, memory management, file systems, and concurrency. By the end of the course, students will be proficient in both designing algorithms and managing operating system resources, preparing them for advanced studies and professional careers in computer science.
Course Description	This lab course is structured to provide an in-depth understanding of both algorithm design and operating system concepts. The Design and Analysis of Algorithms section covers fundamental topics such as sorting, searching, dynamic programming, greedy algorithms, and graph algorithms. Students will learn to critically evaluate the efficiency and applicability of different algorithms. The Operating Systems section delves into process scheduling, memory management techniques, file systems, and synchronization mechanisms. Through a series of hands-on labs and projects, students will apply theoretical knowledge to practical scenarios, reinforcing their understanding and problem-solving abilities.
Course Outline	The course begins with an introduction to basic algorithmic concepts and techniques, progressing through various algorithm design paradigms such as divide-and-conquer, dynamic programming, and greedy methods. Concurrently, students will explore the architecture and functionalities of operating systems, starting with process management and memory management, then advancing to file systems, I/O systems, and concurrency control. The course will include practical lab sessions where students will implement and test algorithms, as well as design and manage operating system components. The course culminates in a comprehensive project that integrates both algorithm

	design and operating system principles to solve complex computing problems.
Learning Outcome	Upon completing this course, students will have a solid grasp of both algorithm design and analysis, as well as operating system functionalities. They will be able to design, analyze, and implement efficient algorithms to address computational problems. Additionally, students will gain practical experience in managing operating system resources, including process scheduling, memory management, and file systems. This dual expertise will equip students with the skills necessary for tackling advanced topics in computer science and pursuing careers in software development, system administration, and research.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 4th Edition
- "Algorithms" by Robert Sedgewick and Kevin Wayne, 4th Edition
- "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, 10th Edition
- "Modern Operating Systems" by Andrew S. Tanenbaum and Herbert Bos, 4th Edition
- "The Algorithm Design Manual" by Steven S. Skiena, 3rd Edition

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6121
Course Credit	3-0-0-3
Course Title	**Advanced Blockchain Technology**
Learning Mode	Offline
Learning Objectives	The objective of this course is to cover a number of popular blockchain platforms and smart contract language paradigms. This course makes the learners familiar with various (a) research challenges, such as interoperability, scalability, security vulnerabilities, functional/non-functional correctness proof, etc., and their possible solutions, (b) synergizing machine Learning and blockchain, and (c) development of secure blockchain-based decentralized applications using Ethereum and Hyperledger.
Course Description	This course will start with a quick introductory background of blockchain technology and its working principle. The primarily focus of this course is to provide a detailed information about the state-of-the-art blockchain platforms and their supported smart contract languages. In particular, syntax, semantics, and paradigms of various smart contract languages will be discussed. In this perspective, blockchain-oriented software development life cycle and decentralized application development will be discussed. Following this, the course will cover two importantdirections: addressing various research challenges in blockchain and AI/machine learning for blockchain (and vice-versa).
Course Outline	Introduction to Blockchain Technology**:** A Quick Tour Different Blockchain Platforms and Smart Contract Languages: Bitcoin, Ethereum, Hyperledger, Solidity, GoLang. Consensus Mechanisms: PoW Vs. PoS, Alternative Consensus Synergizing Machine Learning and Blockchain: Transaction Analysis, Smart Contract Code Analysis, AI-driven Blockchain Applications, Blockchain for AI, Decentralized Learning. Research Challenges in Blockchain: Scalability, Interoperability, Security, Privacy, Decentralized Identity, Smart Contract Vulnerabilities and Detection, Real case studies on developing DApps, Metaverse, Some ongoing relevant research topics.
Learning Outcome	• Gain proficiency in blockchain technology and software engineering of developing decentralized applications. • An overview of the state-of-the-art blockchain platforms and their supported smart contract languages. • Know about the paradigms of various smart contract languages.

	• Understand how AI/machine learning brings benefits to blockchain technology and vice-versa. • Identify various research challenges and opportunities, such as scalability, interoperability
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Don Tapscott and Alex Tapscott: Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World, Portfolio (May 2016).
- Andreas M. Antonopoulos, Gavin Wood: Mastering Ethereum: Building Smart Contracts andDapps, O′Reilly, first edition (Dec 2018).
- Nitin Gaur, Luc Desrosiers, Venkatraman Ramakrishna, Petr Novotny, Salman A. Baset, Anthony O'Dowd: Hands-On Blockchain with Hyperledger, Packt Publishing, first edition (June2018).
- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction, Princeton University Press (2016).
- Stijn Van Hijfte: Blockchain Platforms: A Look at the Underbelly of Distributed Platforms, Morgan & Claypool Publishers, first edition (July 2020)
- Relevant Research Papers and Study Materials available online.

Course Number	CS6122
Course Credit	3-0-0-3
Course Title	**Advanced Cyber Security**
Learning Mode	Offline
Learning Objectives	To have a clear understanding of security and privacy issues in various aspects of computing, including: Programs, Operating systems, Networks, Web Applications
Course Description	The course covers. security and privacy issues in various aspects of computing, including: Programs, Operating systems, Networks, Web Applications
Course Outline	Introduction to Computer Security and Privacy: security and privacy; types of threats and attacks; methods of defense Basics of cryptography, Authentication & key agreement, Authorization and access control Program Security: nonmalicious program errors; vulnerabilities in code, Secure programs; malicious code; Malware detection Internet security: IPSEC, TLS, SSh, Email security Wireless security: WEP, WPA, Bluetooth security, Web Security: XSS attack, CSRF attack, SQL Injection, DoS attack & defense
Learning Outcome	After completion of this course a student will have • Understanding of security issues in computer and networks, • Understanding and analysis of internet security protocols • Understanding and analysis of web security protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Computer Security: Principles and Practice: Dr. William Stallings and Lawrie Brown, Pearson
- O'Reilly Web Application Security by Andrew Hoffman

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6123
Course Credit	3-0-0-3
Course Title	**Advanced Pattern Recognition**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the advanced topics of pattern recognition, including classification and clustering methods. (b) To understand the advanced topics of feature selection, multi-label classification. (c) Apply advanced pattern recognition algorithms to practical applications in image processing, speech recognition, and data mining.
Course Description	This course on advanced pattern recognition aims to equip students with the advanced topics of classification, clustering, and feature selection. By focusing on advanced topics, students will develop the ability to implement and evaluate various pattern recognition algorithms. Students will enhance their understanding of advanced topics of classification, clustering, statistical methods, and data preprocessing techniques through interactive lectures, exercises, and projects. Upon completion, students will be proficient in designing and applying advanced pattern recognition systems for applications such as image processing, text mining, speech recognition, and data mining, thereby enhancing their analytical and problem-solving capabilities in diverse domains.
Course Outline	Introduction and motivation of advanced pattern recognition Modern Classification Methods, Random fields, Pattern recognition based on multidimensional models Contextual classification, Hidden Markov models, Multi-classifier systems Advanced parameter estimation methods, Advanced Unsupervised classification, Modern methods of feature selection. Data normalization and invariants, Benchmarking. Analysis and synthesis of image information. Applications od pattern recognition in Text Processing and Healthcare.
Learning Outcome	• Mastery of advanced concepts in pattern recognition. • In-depth understanding of various advanced algorithms across different pattern recognition paradigms.

	• Comprehensive knowledge of advanced aspects of classification, clustering, feature selection, feature extraction, and projection techniques. • Ability to apply advanced pattern recognition algorithms to real-world projects
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall,1982.
- R. Duda and P. Hart and D.G. Stork, Pattern Classification, J. Wiley, 2001.
- Webb, Statistical Pattern Recognition, J. Wiley, 2002.
- S.Theodoridis, K.Koutroumbas, Pattern Recognition, Elsevier, 2003.
- S. Z. Li, Markov Random Field Modeling in Image Analysis, Springer, 2009.

Research papers will be provided on various topics

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6124
Course Credit	3-0-0-3
Course Title	**Formal Methods in Program Analysis and Verification**
Learning Mode	Offline
Learning Objectives	Formal methods are mathematically rigorous techniques to facilitate in building high-confidence critical systems with stringent quality requirements, such as safety and security. It provides a systematic guidance for specification, development, and verification of software and hardware systems. Examples for such systems are banking software, avionics software, medical device software, software used to control industrial plants/cars, etc. This course will provide necessary background on formal methods and their role in software engineering practice. A range of formal methods will be introduced along with practical case studies of their use. Students will learn how these methods can be used to build reliable software, hardware, and security protocols.
Course Description	This course will start with the fundamentals of set theory, relation and function, lattice theory, propositional and predicate logic, and proof techniques. In order to demonstrate how to analyze and verify a software system, this course will discuss the following three formal approaches using suitable examples: (a) Abstract Interpretation Theory, (b) Temporal Logic and Model Checking, and (c) Deductive Reasoning. In this context, formalism of syntax and semantics of programming languages will be explained considering a simple imperative language WHILE. All these approaches will be illustrated using real-life examples, such as microwave oven, mutual exclusion problem, etc.
Course Outline	Introduction: Introduction to critical systems, Introduction to formal methods and its role, Dependability, Testing Vs. Verification Formal Syntax and Semantics: the WHILE Language, Syntax Vs. Semantics, Formal Program Semantics - Operational, Denotational, Axiomatic Formal Program Analysis: Program Slicing, Dataflow Analysis, Fixpoint Algorithm, Abstract Interpretation Framework Formal Program Verification: Deductive Reasoning; Predicate Abstraction and CEGAR, Temporal Logic and Model Checking, Role of some other formal methods in software engineering New Research Directions: Recent trends on the application of formal methods in Machine Learning and Blockchain Tools: Introduction to various state-of-the-art Analyzers and Verifiers (e.g., NuSMV, UPPAAL, SPIN, ASTREE, CBMC, etc.)
Learning Outcome	• Gain proficiency in formal methods and their role in critical systems. • Understanding formal tools and techniques for analysis and verification of software source codes. • Learning how to define semantics of a software formally, and

	how to abstract its semantics at different levels of precision in order to capture its run-time behavioral properties of interests. • Learning temporal logic to express system's time-varying behaviors. • Applying automatic software verification tools based on model checking and deductive reasoning. • Hands-on experience with NuSMV, Uppaal, Z3 SMT solver, etc. • Application of formal methods in cutting edge research domains including Robotics, IoT, Blockchain Smart Contracts.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Flemming Nielson, Hanne R. Nielson, Chris Hankin. Principles of Program Analysis, Springer, 1999.
- Edmund M. Clarke, Orna Grumberg, Doron A. Peled. Model Checking, The MIT Press, 1999.
- Glynn Winskel. The formal semantics of programming languages: an introduction, The MIT Press, 1993.
- José Bacelar Almeida, Maria João Frade, Jorge Sousa Pinto, Simão Melo de Sousa. Rigorous Software Development: An Introduction to Program Verification. Springer- Verlag London, 2011
- Recent Research Papers relevant to the course.

Course number	CS6125
Course Credit	3-0-0-3
Course Title	**Cyber Physical Systems**
Learning Mode	offline
Learning Objectives	To learn how to model and design the joint dynamics of software, networks, and physical processes., To develop the skills to realize embedded systems that are safe, reliable, and efficient in their use of resources., To learn to think critically about technologies that are available for achieving such joint dynamics.
Course Description	This course will provide an overview of modeling, building, analyzing methods for cyber physical systems.
Course Outline	Models of computation: finite state machines, threads, ordinary differential equations, hybrid systems, actors, discrete-events, data flow Basic analysis, control, and systems simulation: Bisimulations, reachability analysis, controller synthesis, approximating continuous-time systems. Interfacing with the physical world: sensor/actuator modeling and calibration, concurrency in dealing with multiple real-time streams, handling numerical imprecision in software Mapping to embedded platforms: real-time operating systems, execution time analysis, scheduling, concurrency Distributed embedded systems: Protocol design, predictable networking, security
Learning Outcome	• Basic understanding of cyber physical systems • To develop the skills to realize embedded systems that are safe, reliable, and efficient in their use of resources., • To learn to think critically about technologies that are available for achieving such joint dynamics.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Introduction to Embedded Systems - A Cyber-Physical Systems Approach, Second Edition, by E. A. Lee and S. A. Seshia, 2015
- Vahid, F. and T. Givargis (2010). Programming Embedded Systems - An Introduction to Time-Oriented Programming, UniWorld Publishing.
- Schaumont, P. R. (2010). A Practical Introduction to Hardware/Software Codesign, Springer.

- E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*, Addison-Wesley, 2003.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6127
Course Credit	3-0-0-3
Course Title	**Advanced Cloud Computing**
Learning Mode	Offline
Learning Objectives	This course aims to help the students understand (a) how and why cloud systems work and the cloud technologies that manifest these concepts, such as those from Amazon AWS and Microsoft Azure; (b) distributed systems concepts like virtualisation, data parallelism, CAP theorem, and performance analysis at scale; (c) Big Data programming patterns such as Map-Reduce (Hadoop), Vertex-centric graphs (Giraph), Continuous Dataflows (Storm), and NoSQL storage systems to build Cloud applications; (d) Cloud native computing and micro-services.
Course Description	This course provides an in-depth understanding of cloud computing, virtualisation, and distributed systems. It covers foundational concepts, advanced techniques, and real-world applications. Students will explore various aspects of cloud infrastructure, virtualisation technologies, distributed algorithms, and cloud-native computing. By the end of the course, students will be equipped with the knowledge and skills to design, implement, and manage cloud-based solutions and distributed systems effectively.
Course Outline	Cloud computing features and categories. Virtualization: Virtualization Models, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para virtualization, Device virtualization. Virtual Machine: Live VM Migration Stages, Virtual Machine Migration for Enterprise Data Centers, Data Center Workloads, Provisioning methods, Resource provisioning. Geo-distributed Clouds: Server Virtualization, Network Virtualization, Approaches for Networking of VMs: Hardware approach: Single-root I/O virtualization (SR-IOV), Software approach: Open vSwitch, Mininet and its applications. Software Defined Network for Multi-tenant Data Centers: Network virtualization, Case Study: VL2, NVP

	Geo-distributed Cloud Data Centers: Inter-Data Center Networking, Data center interconnection techniques: MPLS, Google's B4 and Microsoft's Swan. Leader Election algorithms in Cloud. Google's Chubby and Apache Zookeeper. Time and Clock Synchronization in Cloud Data Centers, Datacenter time protocol (DTP. Consensus, Paxos and Recovery in Clouds. Cloud Storage: Key-value stores/NoSQL,Design of Apache Cassandra, HBase. Peer to Peer Systems in Cloud Computing. Cloud application: MapReduce Examples. Advances in Cloud Computing with decentralization and Edge Computing.
Learning Outcome	• Cloud Computing as a Distributed Systems: Explain and contrast the role of Cloud computing within this space. • Cloud Virtualization, Abstractions and Enabling Technologies: Explain virtualisation and their role in elastic computing. Characterise the distinctions between Infrastructure, Platform and Software as a Service (IaaS, PaaS, SaaS) abstractions, and Public and Private Clouds, and analyse their advantages and disadvantages. • Programming Patterns for "Big Data" Applications on Cloud: Demonstrate using Map-Reduce, Vertex-Centric and Continuous Dataflow programming models. • Application Execution Models on Clouds: Compare synchronous and asynchronous execution patterns. Design and implement Cloud applications that can scale up on a VM and out across multiple VMs. Illustrate the use of NoSQL Cloud storage for information storage. • Performance, scalability and consistency on Clouds: Explain the distinctions between Consistency, Availability and Partitioning (CAP theorem), and discuss the types of Cloud applications that exhibit these features.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Distributed and Cloud Computing From Parallel Processing to the Internet of Things; Kai Hwang, Jack Dongarra, Geoffrey Fox Publisher: Morgan Kaufmann, Elsevier, 2013.
- Cloud Computing: Principles and Paradigms; Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski Publisher: Wiley, 2011.
- Distributed Algorithms Nancy Lynch Publisher: Morgan Kaufmann, Elsevier, 1996.
- Cloud Computing Bible Barrie Sosinsky Publisher: Wiley, 2011.
- Cloud Computing: Principles, Systems and Applications, Nikos Antonopoulos, Lee Gillam Publisher: Springer, 2012.

Course Number	CS6128
Course Credit	3-0-0-3
Course Title	**Advanced Edge Computing**
Learning Mode	Offline
Learning Objectives	Upon successful completion of this course, students will be able to: (a) understand the fundamental concepts and limitations of cloud computing and identify the advantages of edge computing; (b) describe various edge computing architectures and differentiate them from traditional cloud models; (c) comprehend the principles of distributed systems as they apply to edge computing environments; (d) explore the functionalities of edge data centers and lightweight edge clouds; (e) deploy and manage containerized applications using Docker and Kubernetes in edge computing contexts; and (f) implement and evaluate edge storage systems and end-to-end edge pipelines utilising MQTT and Kafka, as well as investigate advanced edge computing technologies for real-world applications.
Course Description	This course delves into the emerging field of edge computing, providing a comprehensive understanding of its architectures, systems, and technologies. Students will explore the limitations of traditional cloud computing and learn about the advantages and applications of edge computing. The course covers key concepts in distributed systems, edge data centers, and lightweight edge clouds and includes hands-on experience with Docker, Kubernetes, and edge storage systems. Additionally, students will gain insights into end-to-end edge pipelines using MQTT and Kafka and examine advanced edge computing technologies. By the end of the course, students will be equipped with the knowledge and skills to design, implement, and manage edge computing solutions.
Course Outline	Cloud Computing Basics.Edge Computing basics. Edge Computing Use-Cases, Benefits. Different Types of Edge. Edge Deployment Modes. Edge Computing in 5G, Multi-access Edge Computing (MEC) and Mobile Edge Computing.
Learning Outcome	• Critically evaluate advanced edge computing architectures, such as hierarchical, mesh, and hybrid models, considering their suitability for specific use cases and environments.

	• Analyses emerging technologies and trends in advanced edge computing, such as edge AI, blockchain, and serverless computing, and assess their potential impact. • Design and implement innovative edge computing solutions that leverage advanced techniques, such as federated learning, edge caching, and dynamic resource allocation. • Evaluate the performance and scalability of advanced edge computing systems using benchmarking, simulation, and experimentation. • Investigate advanced techniques for ensuring security, privacy, and data integrity in edge computing ecosystems, such as secure enclaves, encryption, and access control mechanisms. • Explore specialised applications of advanced edge computing in domains such as healthcare, smart cities, and autonomous systems, analysing their requirements and challenges.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya (Editor), Satish Narayana Srirama (Editor), Wiley, 2019
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
- Cloud and Distributed Computing: Algorithms and Systems, Rajiv Misra, Yashwant Patel, Wiley 2020.
- Besides these books, we will provide Journal papers as references.

Course Number	CS6129
Course Credit	3-0-0-3
Course Title	**Advanced Computational Data Analysis**
Learning Mode	Offline
Learning Objective	In this subject, the students will be trained with the knowledge of various advanced data analytics techniques encountered in real life.
Course Description	Current Physical systems/devices are highly complex and fast and operate with high data acquisition and generation capabilities. Data generated from such systems require advanced level of analytics for apprehension and further usage. This course aims to give a broad understanding what is advanced level data analytics techniques and how they play a critical role in analysing modern day physical systems acquired data.
Course Outline	Introduction, Operation of physical systems and data generation, Complexity, Drawbacks and Challenges in data generation from physical devices. Requirement of advanced data analytics. Foundations of advanced data analytics principles, mathematical models, probabilistic models, optimization models, deep learning and machine learning models. Role of advanced data analytics in data apprehension and compression, curve-based approximation techniques, interpolation techniques, machine learning models for data interpretation. Statistical models to advanced data analytics, data analytics for 2D and 3D data processing and data manipulation, application of advanced data analytics to real life cases, problem solving.
Learning Outcome	• Gain understanding on data generation systems and the role of advanced data analytics. • Apply the Mathematical models of advanced data analytics to real time • Understand the utilities of statistical models and ML models for advanced data analytics.

Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Signal Processing: A Mathematical Approach, Charles L. Byrne, Second Edition, Chapman & Hall, 2014.
- Digital Functions and Data Reconstruction: Digital-Discrete Methods, Li M Chen, Springer, 2013.
- Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, Bernhard Mehlig, Cambridge University Press, 2021
- Signal Processing and Machine Learning with Applications, Michael M. Richter, Sheuli Paul, Veton Këpuska, Marius Silaghi, Springer Cham, 2022
- Data Compression: The Complete Reference, David Solomon, 4th Edition, Springer, 2007

Course Number	CS6131
Course Credit	3-0-0-3
Course Title	**Reinforcement Learning**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the foundational concepts and mathematical frameworks of reinforcement learning. (b) Gain proficiency in key reinforcement learning algorithms, including dynamic programming, Monte Carlo methods, and temporal-difference learning (c) Apply deep reinforcement learning techniques to solve complex problems using methods such as deep Q-networks and policy gradient algorithms. (d) Explore recent advancements and applications of reinforcement learning, including multi-agent systems and ethical considerations.
Course Description	This specialized course on reinforcement learning aims to give students a deep understanding of the algorithms and methodologies used to train agents to make decisions through trial and error. Students will learn to develop and implement reinforcement learning models by focusing on foundational theories and practical applications. Students will explore key concepts such as Markov decision processes, policy gradients, Q-learning, and deep reinforcement learning through a mix of theoretical lectures, coding exercises, and project-based learning. Upon completion, students will be equipped to design and apply reinforcement learning solutions to complex problems in fields such as robotics, game development, and autonomous systems, enhancing their expertise in this dynamic area of artificial intelligence.
Course Outline	Foundations: Basics of machine learning and reinforcement learning (RL) terminology. Probability Concepts: Axioms of probability, random variables, distributions, and correlation. Markov Decision Process: Introduction to MDPs, Markov property, and Bellman equations. State and Action Value Functions: Concepts of MDP, state, and action value functions. Tabular Methods and Q-networks: Dynamic programming, Monte Carlo, TD learning, and deep Q-networks. Policy Optimization: Policy-based methods, REINFORCE algorithm, and actor-critic methods. Recent Advances and Applications: Meta-learning, multi-agent RL, ethics in RL, and real-world applications.

Learning Outcome	• Mastery of fundamental principles and mathematical frameworks of reinforcement learning. • Proficiency in implementing key reinforcement learning algorithms and techniques. • Ability to apply deep reinforcement learning methods to complex, real-world problems. • Understanding of recent advancements in reinforcement learning and their ethical implications.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto, The MIT Press (1 January 1998).
- Deep Reinforcement Learning Hands-On by Maxim Lapan, Packt Publishing Limited (21 June 2018).
- Algorithms for Reinforcement Learning by Csaba Szepesvari, Morgan and Claypool Publishers (2010)
- Deep Reinforcement Learning: Fundamentals, Research and Applications by Hao Dong, Springer Verlag (2020)

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6132
Course Credit	3-0-0-3
Course Title	**Advanced Graph Machine Learning**
Learning Mode	Offline
Learning Objectives	Several real world systems can be represented as a network of entities that are connected to each other through some relations. Often the number of entities is immensely large, thus forming a very large network. Typical examples of such large networks include network of entities in knowledge graphs, co-occurrence graph of the keywords in natural languages, interaction graph of users in social networks, protein-protein interaction graphs and the network of routers in Internet to name a few. Study of these networks is often needed for relational learning tasks, as well as for developing frameworks for representing the intrinsic structure of the data. This course will mainly deal with both the traditional as well as current state of the art machine learning techniques to be applied on Graphs for different downstream tasks.
Course Description	The course will provide knowledge on the representation and statistical descriptions of large networks, along with traditional machine learning and deep learning techniques applied on graphs. Several use cases of Graph Machine Learning across different domains including Natural Language Processing, Social Network Analysis and Computational Biology would be studied.
Course Outline	Introduction and background knowledge of graphs; Network Measures and Metrices; Spectral Analysis of Graphs and its applications; Random Networks; Properties of Random Networks; Overview of machine learning applications on graphs; Feature based learning on graphs, Shallow embedding and deep Learning techniques for generating node and graph representations – Graph Neural Networks, Graph Attention Networks, Graph Transformers; Graph Neural Networks Pretraining techniques; Generative models for graphs; Models for scale-free and small-world networks; Temporal networks, Modeling temporal networks;

Learning Outcome	Course training via lectures & tutorial sessions to • Represent and analyze the structure of graphs • Discover recurring and significant patterns of interconnections in your data with network motifs and community structure. • Gain Knowledge on traditional machine learning techniques applied on graphs • Leverage graph-structured data to make better predictions using graph neural networks • Understand the problems in dealing with large graphs for machine learning tasks and learn how to improvise • Analyze temporal and dynamic graphs • Scaling neural networks with generative models for graphs.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- M.E.J. Newman, Networks - An introduction , Oxford Univ Press, 2010.
- Yao Ma and Jilian Tang, Deep Learning on Graphs, Cambridge University Press, 2021
- Goyal, Palash and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey." *Knowl.-Based Syst.* 151 (2018): 78-94.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6133
Course Credit	3-0-0-3
Course Title	**Advanced Time Series Analysis**
Learning Mode	Offline
Learning Objectives	• This course on advance time series will teach both the fundamental concepts time series analysis, as well as recent trends in time series analysis. • Students will learn to design successful time series data applications with sequential Neural Networks. • Deploy Nonlinear Auto-regressive Network with Exogenous Inputs • Adapt Deep Neural Networks for Time Series Forecasting and classification
Course Description	This course provides advanced concepts in time series analysis including some fundamentals of time series, data pre-processing, feature selection, Variety of modeling techniques, Anomaly Detection in Time Series and forecasting financial series using statistical, econometric, machine learning, and deep learning approaches and Practical Applications and Deployment of models.
Course Outline	Introduction to classical time series methods, time series Virtualization Univariate Stationary Processes; Granger Causality; Vector Autoregressive Processes Nonstationary Processes; Cointegration; Cointegration in Single Equation Models: Representation,Estimation and Testing. Applied Predictive Modeling Techniques; Autoregressive Conditional Heteroskedasticity. Finance and Algorithmic trading: Machine Learning and Deep Learning in Stock Price

	Prediction Machine Learning, Deep Learned Time series Analysis, Risk and Portfolio Management Practical Applications and Deployment of models; applications of convolutional neural network (CNN) and long-and-short-term memory (LSTM) network architectures; designing predictive models for financial time series data Stock Price Prediction using Deep Learning and Natural Language Processing
Learning Outcome	At the end of the course, students will have achieved the following learning objectives. • problems relating to obtaining, cleaning, simulating, and storing time series data. • Variety of modeling techniques that can be used for recent time series analysis • techniques of financial time series analysis and forecasting financial series using statistical, econometric, machine learning, and deep learning approaches. • Apply more recently developed methods, such as machine learning and neural network, to time series data, highlighting the challenges of data processing and data layout when time series data is used for fitting models
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Kirchgässner, Gebhard, Jürgen Wolters, and Uwe Hassler. *Introduction to modern time series analysis*. Springer Science & Business Media, 2012.
- Lazzeri, F. (2020). *Machine learning for time series forecasting with Python*. John Wiley & Sons.
- Jaydip, Sen, and Mehtab Sidra. *Machine Learning in the Analysis and Forecasting of Financial Time Series*. 2022.
- Gharehbaghi, Arash. *Deep Learning in Time Series Analysis*. CRC Press, 2023.

Course Number	CS5201
Course Credit	3-0-0-3
Course Title	**Advanced Artificial Intelligence**
Learning Mode	Offline
Learning Objectives	• To understand the principles of Artificial Intelligence and the nature of intelligent agents. • To learn various problem-solving techniques, including informed search and exploration. • To gain proficiency in handling constraint satisfaction problems and adversarial search. • To develop a solid foundation in knowledge representation, first-order logic, and propositional logic. • To learn to plan and act effectively in real-world AI applications. • To grasp the concepts of uncertain knowledge and probabilistic reasoning. • To make informed decisions using simple and complex decision-making models. • To acquire skills in learning from observations and applying statistical learning methods. • To explore advanced AI techniques and their practical applications.
Course Description	This course offers an in-depth exploration of advanced concepts in Artificial Intelligence (AI). Students will delve into the theoretical underpinnings and practical applications of AI, examining intelligent agents, the nature of environments, and advanced problem-solving techniques. The curriculum covers informed search and exploration, constraint satisfaction problems, adversarial search, and knowledge representation. Students will also explore reasoning with first-order and propositional logic, planning and acting in real-world scenarios, and handling uncertainty through probabilistic reasoning. The course concludes with statistical learning methods and advanced AI techniques, providing a comprehensive understanding of AI's capabilities and applications.
Course Outline	Introduction and motivation Artificial Intelligence, intelligent agents, nature of environments, Problem-solving by searching, informed search and exploration, constraint satisfaction problem, adversarial search, Knowledge and reasoning, first order logic, inference and propositional logic, knowledge representation,

	Planning and acting in real world of AI agent Uncertain knowledge and reasoning, uncertainty, probabilistic reasoning, making simple and complex decisions Learning from observations and knowledge, statistical learning methods, Some advanced techniques of AI and its applications
Learning Outcome	Upon completing this course, students will be able to: • Analyze and implement intelligent agents in various environments. • Apply informed search techniques to solve complex problems. • Formulate and solve constraint satisfaction problems and engage in adversarial search strategies. • Represent and reason with knowledge using first-order and propositional logic. • Develop and execute plans in real-world AI scenarios. • Manage uncertainty and employ probabilistic reasoning to make sound decisions. • Utilize statistical learning methods to derive insights from data. • Implement advanced AI techniques in real-world applications. • Demonstrate a comprehensive understanding of advanced AI concepts and their implications.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

<u>Suggested Reading</u>

- Russell, S. J., & Norvig, P. (2016). Artificial intelligence: A modern approach. Pearson.
- Poole, D. L., & Mackworth, A. K. (2010). Artificial Intelligence: foundations of computational agents. Cambridge University Press.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: Springer.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS5202
Course Credit	3-0-0-3
Course Title	**Theoretical Computer Science**
Learning Mode	Offline
Learning Objectives	• Explore advanced topics in theoretical computer science, including computational complexity, automata theory, and algorithms. • Develop rigorous mathematical reasoning and problem-solving skills applicable to theoretical computer science. • Understand foundational concepts such as Turing machines, formal languages, and computational models. • Engage in independent research and scholarly exploration of theoretical computer science topics. • Apply theoretical insights to practical problems in computer science and related fields.
Course Description	This course offers an advanced study of theoretical computer science, focusing on formal models of computation, computational complexity theory, and algorithmic analysis. Students will delve into abstract concepts and mathematical techniques essential for understanding the limits and capabilities of computing systems. Topics covered include formal languages, automata theory, computability theory, complexity classes, and advanced algorithms. Through lectures, seminars, and research projects, students will develop a deep understanding of theoretical frameworks and their implications for solving real-world computational problems.

Course Outline	Introduction to Theoretical Computer Science , Historical overview and scope of theoretical computer science , Mathematical foundations in computer science , Overview of formal models of computation Automata Theory , Finite automata and regular languages , Context-free grammars and pushdown automata , Turing machines and computability Formal Languages and Computability , Formal language definitions and properties , Decidability and undecidability , Church-Turing thesis and implications Computational Complexity , Time and space complexity classes , NP-completeness and beyond , Complexity hierarchies and reductions Advanced Topics in Algorithms , Design and analysis of algorithms , Approximation algorithms and randomized algorithms , Advanced data structures Logic in Computer Science , Propositional and first-order logic , Model theory and logical reasoning , Applications of logic in computing Theory of Computation , Formal models beyond Turing machines , Quantum computing and computational models , Complexity aspects of quantum computation Cryptography and Computational Complexity , Basics of cryptography and cryptanalysis , Complexity-theoretic foundations of cryptography , Applications of cryptography in secure computation

Learning Outcome	• Demonstrate advanced knowledge of theoretical computer science principles and methodologies. • Apply mathematical reasoning and formal methods to analyze computational problems. • Evaluate the computational complexity of algorithms and problems using theoretical frameworks. • Conduct independent research in theoretical computer science and contribute to scholarly discourse. • Translate theoretical insights into practical solutions for complex computational challenges.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- "Introduction to the Theory of Computation" (3rd Edition) by Michael Sipser
- "Computational Complexity: A Modern Approach" (1st Edition) by Sanjeev Arora and Boaz Barak
- "Automata Theory, Languages, and Computation" (3rd Edition) by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman
- "Algorithm Design" (1st Edition) by Jon Kleinberg and Éva Tardos
- "Introduction to Algorithms" (3rd Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

Course Number	CS5204
Course Credit	0-1-2-2
Course Title	**Computing Lab-2**
Learning Mode	Offline
Learning Objectives	The primary objective of this course is to equip students with practical skills in Shell Scripting and Android Programming. Students will gain proficiency in writing and debugging shell scripts to automate tasks in Unix-based systems. They will also develop Android applications, understanding the core principles of mobile app development. By the end of the course, students will be able to apply scripting techniques to streamline workflows and create functional Android applications that meet industry standards.
Course Description	This lab course is designed to provide a dual focus on Shell Scripting and Android Programming, offering students a robust understanding of both domains. The Shell Scripting component covers the fundamentals of scripting in Unix-like environments, including file manipulation, process control, and automation of repetitive tasks. The Android Programming section introduces students to the Android development environment, covering topics such as user interface design, event handling, data storage, and network communication. Through a series of hands-on lab exercises and projects, students will develop the skills necessary to write efficient shell scripts and build user-friendly Android applications.
Course Outline	The course begins with an introduction to Unix-based systems and basic shell commands, progressing to more advanced scripting techniques such as control structures, functions, and error handling. Concurrently, students will be introduced to the Android Studio IDE, basic components of Android apps, and fundamental programming concepts. As the course advances, students will delve deeper into both subjects, working on complex shell scripts and developing feature-rich Android applications. The course will culminate in a final project

	where students integrate their knowledge from both areas to solve real-world problems.
Learning Outcome	By the end of this course, students will have a thorough understanding of both Shell Scripting and Android Programming. They will be able to write and debug complex shell scripts to automate system tasks, enhancing productivity and efficiency. In addition, students will be proficient in developing Android applications, from conceptualization to deployment, equipped with the knowledge to design intuitive user interfaces and implement backend functionalities. This dual skill set will prepare students for careers in system administration, software development, and mobile application development.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- "The Linux Command Line: A Complete Introduction" by William E. Shotts, 2nd Edition
- "Learning the bash Shell: Unix Shell Programming" by Cameron Newham, 3rd Edition
- "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano, 4th Edition
- "Head First Android Development: A Brain-Friendly Guide" by Dawn Griffiths and David Griffiths, 2nd Edition
- "Unix and Linux System Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley, 5th Edition

Course Number	CS5205
Course Credit	0-1-2-2
Course Title	**Advanced Artificial Intelligence Lab**
Learning Mode	Offline
Learning Objectives	• To implement the techniques and algorithms learnt in Advance Artificial Intelligence theory • To analyze advanced AI techniques and their practical applications.
Course Description	This course offers an in-depth exploration and practical implementation of advanced concepts in Artificial Intelligence.
Course Outline	Practical implementation of algorithms and techniques learnt in Advance Artificial Intelligence theory
Learning Outcome	Upon completing this course, students will be able to: • Analyze and practically implement the advanced concepts in Artificial Intelligence. • Demonstrate a comprehensive understanding of advanced AI concepts and their implications in real world.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Course Number	CS6221
Course Credit	3-0-0-3
Course Title	**Artificial Internet of Things**
Learning Mode	Offline
Learning Objectives	• Gain a comprehensive understanding of the convergence of Artificial Intelligence (AI) and Internet of Things (IoT), including basic concepts, architectures, and applications. • Learn various AI techniques and their applications in IoT, including machine learning, deep learning, and data analytics. • Develop skills in designing and implementing IoT systems, integrating sensors, and managing data flow. • Understand the processes for collecting, storing, processing, and analyzing IoT data using AI techniques. • Identify and mitigate security risks and privacy concerns in AIoT systems. • Analyze various real-world applications of AIoT in industries such as healthcare, smart cities, agriculture, and manufacturing. • Understand the regulatory and ethical considerations related to AIoT technologies and their deployment.
Course Description	This course provides an in-depth exploration of the convergence of Artificial Intelligence (AI) and the Internet of Things (IoT), known as AIoT. It covers the fundamental principles and technologies of both AI and IoT, demonstrating how they can be integrated to create intelligent, autonomous systems. Students will learn about IoT architecture, AI algorithms, machine learning, data analytics, and the implementation of AI-driven IoT solutions. Through hands-on projects and real-world case studies, students will gain practical experience in developing smart applications for various domains such as smart cities, healthcare, industrial automation, and smart homes.

Course Outline	Introduction to AIoT, Intersection of AI and IoT, Benefits and challenges of AIoT Fundamentals of IoT, IoT Architecture and Protocols, Layers of IoT architecture, Communication protocols and standards, IoT Devices and Sensors Fundamentals of Artificial Intelligence, Machine Learning and Deep Learning, Overview of AI tools and frameworks AIoT System Architecture, Components and Designing AIoT, Edge Computing in AIoT, Edge vs. cloud computing, AI Models for IoT Data Management in AIoT, Data Processing and Analysis, Handling large-scale IoT data, Big data technologies and platforms AIoT Applications and Use Cases: Smart Homes and Buildings, Healthcare and Wearables, Industrial IoT (IIoT), Smart Cities and Transportation AIoT Platforms and Tools: AI Development Tools, Case Studies of AIoT Solutions, AIoT Project Development, Future Trends and Innovations in AIoT
Learning Outcome	At the end of course, students will learn: • Students should grasp the foundational concepts of AI and IoT, including machine learning algorithms, data analytics, sensor technologies, and network protocols. • Ability to integrate AI algorithms with IoT devices and platforms to create intelligent systems capable of data collection, analysis, and decision-making in real-time. • Proficiency in developing AI-driven IoT applications, including sensor data processing, predictive analytics, anomaly detection, and automation. • Awareness of security challenges and solutions in AIoT systems, including data privacy, authentication, encryption, and intrusion detection. • Knowledge of optimization techniques for AIoT systems to enhance performance, scalability, and energy efficiency.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Olivier Hersent, David Boswarthick, and Omar Elloumi, The Internet of Things: Key Applications and Protocols, Wiley
- Maciej Kranz, Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry, Wiley
- John Paul Mueller and Luca Massaron, Machine Learning for the Internet of Things: Practical Guide, Packt

Course Number	CS6222
Course Credit	3-0-0-3
Course Title	**Game Theory**
Learning Mode	Offline
Learning Objectives	• Learn the principles of decision theory and its relevance to game theory. • Understand and analyze extensive form games, including game trees and backward induction. • Identify and compute pure and mixed strategy Nash equilibria. • Analyze matrix games, specifically two-player zero-sum games. • Understand Bayesian games and apply Bayesian equilibrium concepts to games with incomplete information. • Analyze and compute subgame perfect equilibria in dynamic games. • Explore coalitional games, including the core and the Shapley value. • Explore auction theory and its various models and applications. • Utilize game theory concepts in practical applications such as IoT, wireless networks, and cloud computing.
Course Description	This course aims to establish a solid foundation in both game theory and mechanism design, enabling participants to apply these principles rigorously to solve problems. By the end of the course, students will be equipped to model real-world scenarios using game theory, analyze these scenarios with game-theoretic concepts, and design effective and robust solutions, including mechanisms, algorithms, and protocols suitable for rational and intelligent agents.
Course Outline	Non-cooperative Game Theory: Decision theory, Extensive Form Games, Strategic Form Games, Dominant Strategy Equilibria, Pure Strategy Nash Equilibrium, Mixed Strategy Nash Equilibrium, Computation of Nash Equilibrium, Complexity of Computing Nash Equilibrium, Matrix Games (Two Player Zero-sum Games), Bayesian Games, Subgame Perfect Equilibrium. Cooperative Game: Correlated Strategies and Correlated Equilibrium, Two Person Bargaining Problem, Coalitional Games, Core, Shapley Value.

	Mechanism Design: Introduction to Mechanism Design, Social Choice Functions and their properties, Incentive Compatibility, Auction theory and its variants. Applications: IoT, Wireless Networks, Cloud Computing
Learning Outcome	By the end of this course, students will be able to: • Describe the principles of decision theory and its importance in game theory. • Formulate and solve strategic form games, identifying dominant strategy equilibria and Nash equilibria. • Analyze and solve matrix games, particularly two-player zero-sum games. • Formulate Bayesian games and determine Bayesian equilibria for games with incomplete information. • Compute subgame perfect equilibria for dynamic games using appropriate techniques. • Apply the concepts of correlated strategies and correlated equilibria in cooperative settings. • Analyze and solve two-person bargaining problems. • Analyze social choice functions and their properties, focusing on incentive compatibility. • Utilize game theory concepts to address practical problems in IoT, wireless networks, and cloud computing.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbook:

- M. Osborne, An Introduction to Game Theory, Oxford University Press.
- Y. Narahari. Game Theory and Mechanism Design. IISc Press and the World Scientific.

Reference Book:

- M. Maschler, E. Solan, and S. Zamir, Game Theory. Cambridge University Press
- D. Niyato, & W. Saad. Game theory in wireless and communication networks. Cambridge University Press.

Course Number	CS6223
Course Credit	3-0-0-3
Course Title	**Quantum Machine Learning**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) proficiency in implementing and applying classical machine learning algorithms, including classification, regression, gradient descent, and neural networks. (b) grasp the foundational principles of quantum computing, quantum states, qubits, and basic quantum operations.(c) advanced quantum algorithms and their applications in machine learning and computational tasks. (d) gain practical experience in implementing quantum algorithms and simulating quantum processes.
Course Description	This course offers a comprehensive exploration of machine learning (ML) and quantum computing (QC) principles, preparing students to navigate the intersection of classical and quantum computational paradigms. Students will master classical ML techniques including classification, regression, neural networks, and optimization methods like gradient descent. In the quantum computing segment, foundational concepts such as quantum states, qubits, and basic quantum operations (e.g., Hadamard gates) will be covered, alongside encoding classical data on quantum systems and implementing basic quantum algorithms. Advanced topics include variational quantum algorithms, quantum support vector machines, the HHL algorithm for linear systems, and quantum neural networks. Through lectures, practical exercises using quantum programming frameworks, and real-world applications, students will develop a dual proficiency in classical ML and quantum computing, equipping them for roles in research, development, or applications across industries leveraging emerging quantum technologies.
Course Outline	Overview of Machine Learning, Quantum Circuit, Variational quantum algorithm, Quantum Neural Network

Learning Outcome	• Understanding of Machine Learning and Quantum Computing Fundamentals. • Apply the concept of feature vectors, encode data in Quantum computing. • Analysis of Variational quantum algorithms to solve complex problems. • Implementation and analysis of advanced quantum machine learning algorithms.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Nielsen, M.A. and Chuang, I.L., 2010. *Quantum computation and quantum information*.
- Schuld, M. and Petruccione, F., 2021. *Machine learning with quantum computers* (Vol. 676). Berlin.
- Relevant research articles.

Reference books:

- Kasirajan, V., 2021. *Fundamentals of quantum computing*.
- Quantum Machine Learning,
  Link: http://sites.iiserpune.ac.in/~santh/course/QML/qml.html

Course Number	CS6224
Course Credit	3-0-0-3
Course Title	**Text Mining and Analytics**
Learning Mode	Offline
Learning Objectives	• To understand the fundamental principles and scope of text mining and analytics. • To acquire skills in data collection, cleaning, and integration for text data. • To learn text preprocessing techniques including tokenization, stemming, stopword removal, and normalization. • To construct knowledge graphs by linking entities and extracting relationships. • To identify and mine frequent patterns and apply advanced pattern mining techniques. • To extract features from text data and apply clustering and classification methods. • To implement practical applications such as sentiment analysis and text summarization. • To utilize advanced techniques for enhanced text data analysis and mining.
Course Description	This course provides a comprehensive understanding of the principles and techniques of text mining and analytics. Students will learn about data collection, cleaning, integration, and preprocessing methods essential for handling text data. The course covers knowledge graph construction, pattern mining, feature extraction, and advanced text clustering and classification techniques. Practical applications such as sentiment analysis and text summarization are also explored. By the end of the course, students will be prepared to tackle real-world challenges in data mining and text analytics.

Course Outline	Text mining and analytics introduction: Overview, motivation, scope, Data Collection and Pre-processing: Techniques for collecting data from various sources, Text data cleaning and integration, descriptive analytics Text preprocessing: tokenization, stemming, stopword removal, and normalization Knowledge graph construction: Basics of graphs, entity linking, relationship extraction Concepts of frequent patterns, closed patterns, max-patterns, and association rules, mining frequent patterns: apriori algorithm, pattern-growth approach. Advanced: mining sequential patterns Feature extraction, Bag-of-Words, TF-IDF, word embeddings Clustering and classifying text data, Expectation-maximization (EM) algorithm for text data, Latent Dirichlet Allocation (LDA) for topic modeling, and some advanced techniques Some applications: sentiment analysis, text summarization, etc. Some advanced topics and project
Learning Outcome	• By the end of this course, students will be able to: • Understand the core principles and scope of text mining and analytics. • Collect, clean, and integrate text data from various sources. • Apply text preprocessing techniques such as tokenization, stemming, and normalization. • Construct and utilize knowledge graphs for entity linking and relationship extraction. • Identify and mine various patterns in text data, including frequent, closed, and sequential patterns. • Extract features from text data using methods like Bag-of-Words, TF-IDF, and word embeddings. • Perform text clustering and classification using algorithms such as EM and LDA.

	• Implement practical text analytics applications such as sentiment analysis and text summarization. • Utilize advanced techniques for enhanced text data analysis and mining.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Srivastava, A. N., & Sahami, M. (Eds.). (2009). Text mining: Classification, clustering, and applications. CRC press.
- Chakraborty, G., Pagolu, M., & Garla, S. (2014). Text mining and analysis: practical methods, examples, and case studies using SAS. SAS Institute.
- Sarkar, D. (2016). Text analytics with python (Vol. 2). New York, NY, USA:: Apress.
- Witten, I. H., Frank, E., Hall, M. A., Pal, C. J., & Data, M. (2005, June). Practical machine learning tools and techniques. In Data mining (Vol. 2, No. 4, pp. 403-413). Amsterdam, The Netherlands: Elsevier.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6225
Course Credit	3-0-0-3
Course Title	**Knowledge Distillation**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) understand and apply knowledge distillation techniques; (b) master deep neural network compression methods; (c) deploy ML/DNN models on edge devices like Raspberry Pi and others; (d) analyze and optimize model performance in resource-constrained environments; (e) identify the research opportunity in the domain of knowledge distillation and DNN compression on resource-constrained devices.
Course Description	This course delves into advanced techniques for enabling machine learning on resource-constrained devices. Beginning with an introduction to on-device training, students will explore the principles and methods of knowledge distillation and deep neural network (DNN) compression. The course covers practical strategies for implementing machine learning and deep neural networks on devices with limited computational resources. Additionally, students will learn to combine knowledge distillation and compression techniques to optimise performance, making sophisticated machine-learning models viable on edge devices.
Course Outline	Introduction to on-device training: Overview of resource-constrained edge devices and their significance, possibilities of enabling machine learning (ML) and deep neural networks (DNN) models on resource-constrained devices, applications and use cases of ML/DNN on edge devices. Knowledge Distillation: Concept and principles of knowledge distillation, Teacher-student model framework, Applications and benefits of knowledge distillation. Advanced techniques in knowledge distillation, Implementation of knowledge distillation in various frameworks, and Practical exercises on distilling models.

	Deep Neural Network Compression: Overview of DNN compression techniques, Quantization and its impact on model performance, Pruning methods for model size reduction. Low-rank factorization, Weight sharing and clustering, Hands-on implementation of compression techniques. ML/DNN on resource-constrained devices: Introduction to edge devices: Raspberry Pi, NVIDIA Jetson, etc, Setting up an AI development environment on Raspberry Pi, Case study: Running a pre-trained model on Raspberry Pi. TensorFlow Lite, ONNX, etc, Practical exercises with TensorFlow Lite on Raspberry Pi. Combining Knowledge Distillation and Compression: Integrating knowledge distillation and compression for optimal performance, Strategies for balancing accuracy and efficiency, Real-world examples and case studies.
Learning Outcome	• Explain and implement knowledge distillation techniques. • Apply DNN compression methods such as quantisation and pruning. • Set up and optimise ML/DNN models on Raspberry Pi using TensorFlow Lite and ONNX. • Evaluate and enhance ML/DNN model performance on edge devices. • Create real-time applications, including object detection and predictive maintenance. • Plan, develop and present comprehensive projects that may lead to publication.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

<u>Suggested Reading</u>

- Deep Learning for Edge AI" by John Doe
- Knowledge Distillation: Principles, Methods and Applications" by Jane Smith
- Official documentation and tutorials for TensorFlow Lite, ONNX, and edge devices
- "Knowledge Distillation: A Survey" Jianping Gou, Baosheng Yu, Stephen John Maybank, Dacheng Tao
- K. Nan, S. Liu, J. Du and H. Liu, "Deep model compression for mobile platforms: A survey," in Tsinghua Science and Technology, vol. 24, no. 6, pp. 677-693, Dec. 2019, doi: 10.26599/TST.2018.9010103.
- Mishra et al.. "A survey on deep neural network compression: Challenges, overview, and solutions."

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6226
Course Credit	3-0-0-3
Course Title	**Quantum Cyber Security**
Learning Mode	Physical
Learning Objectives	To have a clear understanding of Quantum technology and that brings on the security and privacy of communication and computation
Course Description	The course covers various effects that developing quantum technologies will have on cyber security.
Course Outline	Quantum information concepts: qubits, mixed states, operations, distance measures, quantum circuits, quantum algorithms (factoring, discrete logarithms, search). Classical Cryptography, encryption, authentication and key distribution protocols, Security analysis, quantum cryptography, quantum-key-distribution protocols, Security and implementation aspects. classical protocols and their security under quantum attacks, general quantum attacks (superposition attacks)
Learning Outcome	After completion of this course a student will have • Understanding of the quantum technologies. • Demonstrate their understanding of the power of quantum algorithms and be able to use the basic mathematical formalism for quantum information and quantum cryptography • Test whether a classical cryptosystem is secure against a range of quantum attacks • Use security notions for quantum information, such as encryption and authentication, in quantum cryptographic protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Quantum Computation and Quantum Information by Nielsen and Chuang
- Cryptography: Theory and Practice by D.R. Stinson

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6228
Course Credit	3-0-0-3
Course Title	**Selected Topics in Wireless Networks**
Learning Mode	Offline
Learning Objectives	In this subject, the students will be trained with the knowledge of 802.11 wireless networks, including protocol knowledge and the associated security vulnerabilities.
Course Description	In the consumer, industrial, and military sectors, 802.11-based wireless access networks have been widely used due to their convenience. This application, however, is reliant on the unstated assumptions of availability and anonymity. The management and media access protocols of 802.11 may be particularly vulnerable to malicious denial-of-service (DoS) and various security attacks. This course analyzes these 802.11-specific attacks, including their applicability, effectiveness, and proposed low-cost implementation improvements to mitigate the underlying vulnerabilities.
Course Outline	Introduction to Wireless Networks: Basic principles, types of wireless networks (Wi-Fi, Bluetooth, cellular), and network topologies. Wireless Communication Fundamentals: Radio frequency, signal propagation, modulation techniques, and interference management. Network Protocols and Standards: IEEE 802.11 (Wi-Fi), IEEE 802.15 (Bluetooth), and cellular standards (2G, 3G, 4G, 5G). Network Design and Architecture: System design, frequency reuse, and resource allocation. Mobility and Handoff: Techniques for managing mobility, handoff processes, and roaming. Security in Wireless Networks: Security protocols, encryption, and threat mitigation. Emerging Technologies: Overview of 6G, IoT, in-network caching

Learning Outcome	On successful completion of the course, students should be able to: • Understand the fundamentals of 802.11 wireless networks • Describe the WLAN services-association, disassociation, re-association, distribution, integration, authentication, de authentication and data delivery services • Comprehend the vulnerabilities associated with 802.11 protocol.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books and References:

- "Wireless Communications: Principles and Practice" by Theodore S. Rappaport (2nd Edition)
- "802.11 Wireless Networks: The Definitive Guide" by Matthew S. Gast (2nd Edition)
- "Wireless Communications & Networks" by William Stallings (2nd Edition)
- "Wireless Communications: Principles and Practice" by Andreas F. Molisch (2nd Edition)
- "Fundamentals of Wireless Communication" by David Tse and Pramod Viswanath (1st Edition)
- "Next Generation Wireless LANs: 802.11n and 802.11ac" by Eldad Perahia and Robert Stacey (2nd Edition)
- "Wireless Networking: Understanding Internetworking Challenges" by Anurag Kumar, D. Manjunath, and Joy Kuri (1st Edition)
- "Wireless Communications: Principles and Practice" by Kaveh Pahlavan and Prashant Krishnamurthy (1st Edition)

Course Number	CS6229
Course Credit	3-0-0-3
Course Title	**Advanced Big Data Analytics**
Learning Mode	Offline
Learning Objectives	The primary objective of this course is to equip students with advanced knowledge and skills in big data analytics. By the end of the course, students will be able to understand and apply advanced data analysis techniques, develop and implement big data solutions, and leverage big data technologies for strategic decision-making. Additionally, students will gain proficiency in using big data tools and platforms, enhance their ability to handle and analyze large datasets, and develop critical thinking skills for solving complex data-driven problems.
Course Description	This course provides an in-depth exploration of advanced big data analytics, focusing on the theoretical foundations, practical techniques, and cutting-edge technologies in the field. Students will learn about various aspects of big data, including data acquisition, storage, processing, and analysis. The course covers advanced topics such as machine learning algorithms for big data, real-time data processing, and big data visualization. Emphasis will be placed on the use of big data tools and platforms such as Hadoop, Spark, and NoSQL databases. Through hands-on projects and case studies, students will develop the skills needed to design and implement big data solutions for a variety of applications.
Course Outline	Introduction to Big Data Analytics: Definition and characteristics (Volume, Velocity, Variety, Veracity, and Value), Importance and challenges of Big Data. Big Data Ecosystem: Components and architecture, Key players and technologies in Big Data (e.g., Hadoop, Spark). Big Data vs. Traditional Data: Differences in processing and analysis, Applications of Big Data Analytics- Industry-specific applications and Case studies Data Acquisition and Storage: Structured, semi-structured, and unstructured data and Data generation and collection methods. Distributed file systems (e.g., HDFS), NoSQL databases (e.g.,

	MongoDB, Cassandra), and Cloud storage options, ETL (Extract, Transform, Load) processes, Data pipelines and workflow automation, Insuring data integrity and accuracy, Data privacy and security considerations Data Processing Frameworks: Hadoop MapReduce architecture and workflow, Advantages and limitations, Apache Storm, Apache Flink, and Kafka Streams, Real-time data processing and its significance, Apache Spark architecture and RDDs (Resilient Distributed Datasets), Spark SQL, Spark Streaming, and MLlib Machine Learning for Big Data: Introduction to Machine Learning, Machine Learning Algorithms, Machine Learning Tools and Libraries, Training and evaluating models on large datasets, Scalability and performance optimization Real-Time Data Processing: Importance and applications of real-time analytics, Apache Kafka, Apache Flink, and Apache Storm, Designing and implementing real-time data workflows, Industry examples and best practices Big Data Visualization: Making data comprehensible and actionable, Visualization Tools and Techniques, Building user-friendly and interactive data dashboards, Intersection of data science and big data analytics, Integrating AI techniques with big data analytics, Processing and analyzing IoT-generated data, Distributed computing at the edge of networks, and Industry-Specific Case Studies- Healthcare, finance, retail, and other industries.
Learning Outcome	• Understand the key concepts and significance of big data analytics. • Acquire, store, and manage large datasets using appropriate big data technologies. • Apply advanced data processing techniques using Hadoop and Spark. • Implement machine learning algorithms for big data applications. • Perform real-time data processing and analysis. • Utilize big data visualization tools to interpret and present data insights. • Develop and implement comprehensive big data solutions for various industry applications. • Critically evaluate and solve complex data-driven problems using advanced analytics techniques.

Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Marz, N., & Warren, J. (2015). Big Data: Principles and Best Practices of Scalable Real-Time Data Systems (1st ed.). Manning Publications.
- White, T. (2015). Hadoop: The Definitive Guide (4th ed.). O'Reilly Media.
- Karau, H., & Warren, R. (2017). High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark (1st ed.). O'Reilly Media.
- Gulla, U., Gupta, S., & Kumar, V. (2020). Practical Big Data Analytics: Hands-on Techniques to Implement Enterprise Analytics and Machine Learning Using Hadoop, Spark, NoSQL and R (2nd ed.). Packt Publishing.
- Zikopoulos, P. C., Eaton, C., & deRoos, D. (2012). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data (1st ed.). McGraw-Hill Education.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6231
Course Credit	3-0-0-3
Course Title	**Selected topics on Cryptography**
Learning Mode	physical
Learning Objectives	To have a clear understanding of design and analysis of different advanced cryptographic protocols
Course Description	The course covers design and analysis of advanced cryptographic protocols
Course Outline	Mathematical Background: Modular Arithmetic, Finite Fields, Elliptic Curves over Finite Fields Crypto basics: SKE. PKE Zero-knowledge proofs, and protocols, Secret sharing, Commitment, Oblivious transfer, Secure multiparty computation, Homomorphic encryption, obfuscation, Post quantum cryptography
Learning Outcome	After completion of this course a student will have • Understanding and analysis of symmetric key cryptography • Understanding and analysis of asymmetric key cryptography • Designing and analysis of advanced cryptographic protocols • Understanding and analysis of post quantum cryptographic protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Doug Stinson, Cryptography: Theory and Practice, Chapman and Hall/CRC

Course Number	CS6232
Course Credit	3-0-0-3
Course Title	**High Performance Computing**
Learning Mode	Offline
Learning Objectives	The course is designed to provide basic understanding of structure, and function of various building blocks of high performance Computing System. Students will be able to design various functional units and components and to identify the elements of modern GPUs and their impact on processor/GPU/TPU and parallel architecture design including memory
Course Description	Using a set of fundamental techniques and technologies, the high performance systems theme broadly explains how computing platforms work at various levels of abstraction, including both software and hardware. The course introduces HPS architecture with focus on parallel architectures
Course Outline	Computer types, Structure with basic computer components - instruction sets of some common CPUs/GPUs; Parallel Processing Concepts: a) Levels of parallelism (instruction, transaction, task, thread, memory, function) Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc) c) Architectures: N-wide superscalar architectures, multi-core, multi-threaded Parallel Programming with CUDA: a) Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: Fundamental Design Issues in Parallel Computing: a) Synchronization b) Scheduling, c) Job Allocation d) Job Partitioning, e) Dependency Analysis,f) Mapping Parallel Algorithms onto Parallel Architectures g) Performance Analysis of Parallel Algorithms Power-Aware Computing and Communication: a) Power-aware Processing Techniques Advanced Topics:(a) Petascale Computing,(b) Optics in Parallel Computing,(c) Quantum Computers,(d) Recent developments in Nanotechnology and its impact on HPC

Learning Outcome	The student will be able to : • Appreciate understanding of the HPC blocks, key terminology, and current industry trends in high performance computer architecture. • Understand parallel programming and evaluate and compare the architectural features of the state of the art high performance commodity hardware platforms. • Understand the processor (CPU/GPU/TPU) subsystem. • Employ concepts of the HPS memory subsystem and hierarchy
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books:

- "Highly Parallel Computing", by George S. Almasi and Alan Gottlieb
- "Advanced Computer Architecture: Parallelism, Scalability, Programmability", by Kai Hwang, McGraw Hill 1993
- "Parallel Computer Architecture: A hardware/Software Approach", by David Culler Jaswinder Pal Singh, Morgan Kaufmann, 1999.
- "Scalable Parallel Computing", by Kai Hwang, McGraw Hill 1998.
- "Principles and Practices on Interconnection Networks", by William James Dally and Brian Towles, Morgan Kauffman 2004.
- GPU Gems 3 --- by Hubert Nguyen (Chapter 29 to Chapter 41)
- Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, © 2003.
- Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, © 2007.

Course Number	CS6233
Course Credit	3-0-0-3
Course Title	**Selective Topics in Generative AI**
Learning Mode	Offline
Learning Objectives	To gain a comprehensive understanding of advanced AI architectures, particularly in the context of Generative AI. To develop proficiency in implementing and evaluating a variety of Generative AI techniques and models. To understand the principles and applications of Generative Pre-trained Transformers and other application-specific architectures. To explore and address ethical considerations and biases in Generative AI, emphasizing the importance of explainability. To engage with advanced topics and apply knowledge through hands-on projects.
Course Description	This course provides an in-depth exploration of Generative AI (GenAI), focusing on advanced AI architectures such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Generative Pre-trained Transformers (GPT). Students will learn about hybrid and emerging models, application-specific architectures, and the ethical considerations and biases in Generative AI. The course includes hands-on projects to design, implement, and evaluate sophisticated generative AI models, emphasizing innovation and practical problem-solving skills.
Course Outline	Introduction to advanced AI, overview of advanced AI architectures and Generative AI (GenAI) Generative Adversarial Network (GAN): various GAN architectures, DCGAN Advanced Variational AutoEncoder (VAE): hierarchical VAEs, Semi-supervised VAE

	Hybrid and emerging models: Energy-based models, diffusion models, autoregressive and flow-based models, attention mechanism in generative models Generative Pre-trained Transformers (GPT): architectural details and variations Advanced application-specific architecture: Models for Image-to-Text generation, Text-to-Image generation, Prompt engineering, Multimodality Ethical consideration and bias in Generative AI, Explainability Some advanced topics and project
Learning Outcome	• Master various Generative AI architectures, including GANs, VAEs, and emerging models. • Demonstrate proficiency in implementing and evaluating advanced Generative AI techniques, such as hierarchical VAEs and energy-based models. • Understand the design principles and applications of Generative Pre-trained Transformers (GPT) and application-specific architectures. • Analyze and address ethical considerations and biases in Generative AI, emphasizing the importance of explainability. • Explore advanced topics in Generative AI and apply acquired knowledge through hands-on projects, fostering innovation and practical problem-solving skills.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Foster, D. (2022). Generative deep learning: : Teaching Machines to Paint, Write, Compose, and Play. " O'Reilly Media, Inc.".
- Valle, R. (2019). Hands-On Generative Adversarial Networks with Keras: Your guide to implementing next-generation generative adversarial networks. Packt Publishing Ltd.

- Research Papers and Articles from Journals such as JMLR, IEEE Transactions on Neural Networks and Learning Systems, etc., and Conference Proceedings from NeurIPS, ICML, and CVPR,etc.

INDIAN INSTITUTE OF TECHNOLOGY PATNA

Department of Computer Science & Engineering

Program: M. Tech CSE

Curriculum and Syllabus-2024

Course Number	CS6109 (IDE-1)
Course Credit	3-0-0-3
Course Title	**Drone Data Processing & Analysis**
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) understand the integration of various sensors and platforms, including optical, thermal, LiDAR, multispectral, and hyperspectral sensors; (b) identify and analyze the use of drones in civilian and remote sensing applications; (c) to learn the importance and techniques of sensor calibration and boresighting to ensure data accuracy and reliability; (d) comprehend the operational requirements for UAVs and develop a Concept of Operation (CONOP) including risk assessment for safe and effective drone missions; (e) gain proficiency in using advanced data processing software tools to generate high-quality digital products from drone data; and (f) to evaluate data quality using accuracy metrics and understand the latest mapping standards to ensure high-precision geospatial data.
Course Description	This course provides an in-depth exploration of advanced drone systems, focusing on integrating and applying various sensors, including optical, thermal, LiDAR, multispectral, and hyperspectral. Students will learn how to integrate these sensors with different drone platforms and apply them in various fields such as agriculture, construction, environmental monitoring, and urban planning. The course covers using drones in remote sensing for resource management, disaster response, and scientific research, emphasising the importance of sensor calibration, boresighting methods, and operational best practices. Additionally, students will gain hands-on experience with leading software tools for drone data processing and understand the latest standards for geospatial data accuracy. Regulatory compliance, safety, security, and privacy issues will also be addressed. Practical applications and industry case studies will be analysed to illustrate successful drone data processing projects.
Course Outline	Importance of Calibration, Methods and best practices for boresighting to align sensors accurately, Key operational requirements and best practices, Developing and implementing a Concept of Operation (CONOP) for drone missions, Risk assessment Introduction to leading software tools for drone data processing (e.g., Pix4D, Agisoft Metashape, DroneDeploy), Steps from data acquisition to final product generation Understanding the latest standards for geospatial data accuracy, Techniques for identifying and correcting errors in drone data, Creating digital elevation models (DEMs) and digital terrain models (DTMs)

	Current Rules and Regulations in India, Compliance and Certification, Comparison with global regulatory standards, UAV Safety Issues, Security Concerns, Privacy Issues Practical Applications and Case Studies, Analysis of successful drone data processing projects in various industries
Learning Outcome	• Evaluate and apply drone technology in diverse civilian and remote sensing scenarios, identifying the benefits and challenges of each application. • Execute proper calibration and boresighting techniques to ensure the accuracy and reliability of sensor data. • Create and implement an effective CONOP for drone missions, including risk assessment and mitigation strategies. • Efficiently use leading data processing software tools to process drone data and generate high-quality digital products. • Assess data quality using established accuracy metrics and apply the latest mapping standards to ensure high-precision geospatial data. • Navigate and comply with current drone regulations in India and understand international regulatory frameworks.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Barnhart, R., Michael, M., Marshall, D., and Shappee, E. ed. 2016. Introduction to Unmanned Aircraft Systems, 2nd edition. Boca Raton. CRC Press.
- Fahlstrom, P. and Gleason, T. 2012. Introduction to UAV Systems. 4th edition. United Kingdom. John Wiley & Sons Ltd.
- Wolf, P., DeWitt, B., and Wilkinson, B. 2014. Elements of Photogrammetry with Applications in GIS, 4th edition. McGraw-Hil
- Introduction to UAV Systems, Paul G. Fahlstrom and Thomas J. Gleason
- Drone Technology in Architecture, Engineering, and Construction, Daniel Tal and Jon Altschuld

- UAV or Drones for Remote Sensing Applications, edited by Felipe Gonzalez Toro and Antonios Tsourdos