
PES
UNIVERSITY

B.TECH IN COMPUTER SCIENCE AND ENGINEERING(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

PROGRAM EDUCATIONAL OBJECTIVES

- Prepare and train students in theoretical foundations to work with cutting edge computing technologies and design solutions to complex engineering problems, making them ready to work in industrial environment.
- Develop all round skills such as team building, inter-personal skills, and leadership qualities in order to effectively communicate with engineering community and with society at large.
- Promote research culture through internships, research assistantships, research-oriented projects, sponsored and collaborative research and enable them to pursue higher studies in computer science and related fields.
- To inculcate social concern meeting the requirements of prospective employers and to develop an ability to innovate efficient computing solutions for a better society.
- Create professionally superior and ethically strong globally competent employees and entrepreneurs.

PROGRAM OUTCOMES

- Apply mathematical and theoretical principles in the modeling and design of high-quality computer-based systems using state-of-the-art computer technology.
- Conduct in-depth study of research literature in the area of Computer Science, analyse problems in order to arrive at substantiated conclusions using first principles of mathematics, and allied sciences.
- Design, implement and evaluate Computer Systems, programs and processes that meet partial/complete specifications with concern for society, environment, and culture.
- Design and conduct experiments, collect data, analyze, and interpret the results to investigate complex engineering problems in the field of Computer Science.
- Apply state-of-the-art techniques and modern computer-based tools in prediction, comparison, and modeling of complex engineering activities.
- Have a sound understanding of professional, legal, security and social issues and responsibilities in engineering activities involving Computer Science.
- Understand societal and environmental concerns and demonstrate responsibility in sustainable development of computer-based solutions.
- Be aware of ethical and professional responsibilities in engineering situations; make informed judgments regarding intellectual property and rights in relation to computer-based solutions in global, economic, environmental and societal contexts.
- Able to function effectively in teams to establish goals, plant asks, meet deadlines, manage risk, and produce high-quality technical solutions.
- Contribute and communicate effectively with the society, be able to write effective reports and design documents by adhering to appropriate standards, make effective presentations, give, and receive clear instructions.
- Apply skills in clear communication, responsible teamwork, and time management by, for example, managing a team or project and communicating with external stakeholders.
- Recognize the need for and demonstrate an ability to engage in continuing professional development in its broadest sense.

B.TECH IN COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

I SEMESTER (2025-29 BATCH)

Sl. No	Course Code	Course Title	Hours per week				Credits	Tools/Languages	Course Type
			L	T	P	S	C		
1	UE25CS151A	Python for Computational Problem Solving	4	0	2	5	5	Python interpreter 3.8 and above. IDLE and any IDE like Jupyter. Copilot aided teaching - Not copilot driven course Chatgpt	FC- Lab Integrated

II SEMESTER (2025-29 BATCH)

Sl. No	Course Code	Course Title	Hours per week				Credits	Tools/Languages	Course Type
			L	T	P	S	C		
1	UE25CS151B	Problem Solving with C	4	0	2	5	5	C Programming Language GCC Compiler, GDB Debugger AI Tools- Code 5, DeepCode, Codium, AIcode Helper	FC- Lab Integrated

III SEMESTER (2024-28 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	AI Tools/ Tools/Languages	Course Type
			L	T	P	S	C		
1	UE24CS251A	Digital Design and Computer Organization	4	0	2	5	5	Logisim- for circuit simulation, Logically- for circuit simulation	CC- Lab Integrated
2	UE24CS252A	Data Structures and its Applications	4	0	2	5	5	C-Programming language Visu Algo (Interactive Visualizations), Algorithm Visualizer (AI Explanations)	CC-Lab Integrated
3	UE24CS241A	Mathematics for Computer Science and Engineering	4	0	0	4	4	Jupyter Notebook, Python, Pandas, Matplotlib, Scipy, Seaborn, BeautifulSoup, Numpy, Scikit learn. R (R Studio) and Python, Orange.	CC- Independent
4	UE24CS242A	Web Technologies	4	0	0	4	4	HTML, CSS, JavaScript, MERN Technologies. GitHub Copilot and Tabnine	CC- Independent
5	UE24CS243A	Automata Formal Languages and Logic	4	0	0	4	4	PLY, JFLAP AI Tool - https://www.opentrain.ai/ glossary/automata-theory https://www.yeschat.ai/ gp ts-9t55kXLEUfB-AutomataExpert	CC- Independent
6	UE24EC231A	CIEL1	2	0	0	2	2		CC- Independent
7	UE25MA231A	Bridge Course Mathematics–I (Applicable for The Lateral Entry Students)	2	0	0	2	0		FC- Independent
Total			**22/ 24**	**0**	**4**	**24/ 26**	**24**		

IV SEMESTER (2024-28 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	AI Tools/ Tools/Languages	Course Type
			L	T	P	S	C		
1	UE24CS251B	Microprocessor and Computer Architecture	4	0	2	5	5	ARM Simulator, Arduino Microcontroller kit, Para Cache simulator. Compute intensive Application are run on GPU systems	CC-Lab Integrated
2	UE24CS252B	Computer Networks	4	0	2	5	5	Wireshark, Python. Advanced-Computer Networking-Tutor Computer Network Professor (yeschat.ai)AI	CC-Lab Integrated
3	UE24CS241B	Design and Analysis of Algorithms	4	0	0	4	4	C-Programming language Visu Algo (Interactive Visualizations), Algorithm Visualizer (AI Explanations)	CC-Independent
4	UE24CS242B	Operating Systems	4	0	0	4	4	C, Linux/ Unix OS for system call implementation	CC-Independent
5	UE24MA241B	Linear Algebra and its Applications	4	0	0	4	4	Python IDE, MathGPT, Deepseek	CC-Independent
6	UE24EC221B	CIEL2	2	0	0	2	2		CC-Independent
7	UE25MA221B	Bridge Course Mathematics–II (Applicable to Lateral Entry Students)	2	0	0	2	0		FC-Independent
Total			**24/26**	**0**	**4**	**24/ 26**	**24**		

V SEMESTER (2023-27 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	AI Tools/ Tools/Languages	Course Type
			L	T	P	S	C		
1	UE23CS351A	Database Management System	4	0	2	5	5	MySQL Work bench, Python, ERwin, Any other tool for ER modeling Bytebase SQL Editor, Chat2DB	CC-Lab Integrated
2	UE23CS352A	Machine Learning	4	0	2	5	5	NetworkX for statistical features of graphs, Tensor flow, Keras, and Scikit Learn for traditional graph ML, and Pytorch Geometric for Graph Neural Networks. Google cloud Vertex AI(end-to-end ML platform), Amazon Sagemaker, IBM Watson studio, Auto ML tools (H2O.ai (H2O AutoML), Google AutoML, Hugging Face, Haystack etc	CC-Lab Integrated
3	UE23CS341A	Software Engineering	4	0	0	4	4	Git, GitHub, docker, Jenkins, gtest, pytest, Jira/OpenProject, SonarQube.	CC-Independent
4	UE23AM342AAX	Elective I	4	0	0	4	4		EC
5	UE23AM343ABX	Elective II	4	0	0	4	4		EC
6	UE23AM320A	Capstone Project Phase-I	0	0	8	4	2		PW
Total			**20**	**0**	**12**	**24**	**24**		
Elective–I									
7	UE23AM342AA1	Advanced Foundations for ML	4	0	0	4	4	Pytorch, SkLearn, Keras, Tensorflow Google cloud Vertex AI (end to-end ML platform).	EC-Independent
8	UE23AM342AA2	Active Learning	4	0	0	4	4	Pytorch, SkLearn, Keras, Tensorflow Google cloud Vertex AI (end to-end ML platform)..	EC-Independent
9	UE23AM342AA3	Social Computing	4	0	0	4	4	NetworkX, along NLTK, Spacy, PyTorch, etc..	EC-Independent

10	UE23AM342AA4	AI for Smart Manufacturing	4	0	0	4	4	OracleERP, Salesforce CRM, SCM applications and tools, MLtools	EC-Independent
11	UE23AM342AA5	AI for Medical Imaging	4	0	0	4	4	Python, Tensorflow/PyTorch, ImageJ, MATLAB.	EC-Independent
12	UE23CS342AA3	Internet of Things	4	0	0	4	4	Python,Embedded-C, CloudPlatforms, SingleBoard Computers	EC-Independent
13	UE23CS342AA4	Applied Cryptography	4	0	0	4	4	SEED labs, Python ProgrammingLanguage	EC-Independent
14	UE23CS342AA5	Virtual and Augmented Reality	4	0	0	4	4	C/C++/JAVA/ Python using OpenGL. Unit3D, ThreeJS, Blender, Cloud Compare, OpenGL, Unity ML Agents	EC-Independent
15	UE23CS342AA7	Topics in Wireless Networks and 5G	4	0	0	4	4	Cisco Packet Tracer, OMNET++	EC-Independent
Elective–II									
16	UE23AM343AB1	Advanced Data Analytics	4	0	0	4	4	Python and related libraries for Machine learning and Causal Inference.	EC-Independent
17	UE23AM343AB2	Privacy Preserving Machine Learning	4	0	0	4	4	Python tools and librariesused for PPML, such as PySyft.	EC-Independent
18	UE23AM343AB3	Optimizations and Metaheuristics	4	0	0	4	4	scikit-opt and scikitlearn, pyswarm, scipy, DEAP, etc	EC-Independent
19	UE23AM343AB4	Supply Chain Optimization using AI	4	0	0	4	4	Python and various search and optimization libraries.	EC-Independent
20	UE23AM343AB5	Machine Learning with Business Data	4	0	0	4	4	Python as ScikitLearn, XGBoost,Light GBM, CatBoost,TabNet, Pytorch Tabular.	EC-Independent
21	UE23CS343AB5	Advanced Computer Networks	4	0	0	4	4	GNS3, Cisco Packet Tracer, Mininet.	EC-Independent
22	UE23CS343AB6	Computer Network Security	4	0	0	4	4	SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.	EC-Independent
23	UE23CS343AB7	ROS for autonomous system	4	0	0	4	4	C++,Python,Java, ROS,Rviz, Gazebo and Webots.	EC-Independent

ELECTIVES TO BE OPTED FOR SPECIALIZATION			
Sl. No.	**SPECIALIZATION**	**ELECTIVE–I**	**ELECTIVE–II**
A	Applied and Advanced Machine Learning (AAML)	UE23AM342AA1, UE23AM342AA2, UE23AM342AA3.	UE23AM343AB1, UE23AM343AB2, UE23AM343AB3.
B	AI for Industry (AII)	UE23AM342AA4, UE23AM342AA5.	UE23AM343AB4, UE23AM343AB5.
C	Cyber Security & Connected Systems (CSCS)	UE23CS342AA3, UE23CS342AA4, UE23CS342AA5. UE23CS342AA7.	UE23CS343AB5, UE23CS343AB6, UE23CS343AB7

VI SEMESTER (2023-26 BATCH)

Sl. No	Course Code	Course Title	Hours per week				Credits	AI Tools/ Tools/Languages	Course Type
			L	T	P	S	C		
1	UE23CS351B	Cloud Computing	4	0	2	5	5	Amazon AWS (or equivalent), AWS Skill Builder, AWS Educate, Qwiklabs, Docker, Kubernetes, Jenkins, Zookeeper, Github, NoSQL database, Flask, Python, Go Lang. AWS, Sagemaker	CC
2	UE23CS352B	Object Oriented Analysis and Design	4	0	2	5	5	Star UML, Object Oriented Programming Language(Java/C++)	CC
3	UE23CS341B	Compiler Design	4	0	0	4	4	Lex/flex and YACC/Bison. Compiler Explorer + AI Explanation	CC
4	UE23AM342BAX	Elective III	4	0	0	4	4		EC
5	UE23AM343BBX	Elective IV	4	0	0	4	4		EC
6	UE23AM320B	Capstone Project Phase - II	0	0	8	2	2		PW
Total			**20**	**0**	**12**	**24**	**24**		
Elective–III									
7	UE23AM342BA1	Interdisciplinary Deep Learning on Graphs	4	0	0	4	4	Pytorch Geometric for Graph Neural Networks	EC-Independent
8	UE23AM342BA2	Large language Models and Their Applications	4	0	0	4	4	Python as Scikit Learn, XGBoost,Light GBM.	EC-Independent
9	UE23AM342BA3	Explainable AI	4	0	0	4	4	Python Tools and Libraries.	EC-Independent
10	UE23AM342BA4	Deep Learning for Life Sciences	4	0	0	4	4	Python Tools and Libraries.	EC-Independent
11	UE23AM342BA5	Intelligent Systems with Knowledge Graphs	4	0	0	4	4	Python, Neo4J, Networkx, NLP Libraries.	EC-Independent
12	UE23CS342BA5	BlockChain	4	0	0	4	4	Solidity, Ganache, Meta mask. RemixID, Ganache tool	EC-Independent

13	UE23CS342BA6	Digital Forensics and Incident Response	4	0	0	4	4	Open source tools on Forensics. Cellebrite,Deepfake Detection Tools	EC-Independent
14	UE23CS342BA7	Digital Twin	4	0	0	4	4	C/C++/JAVA/Python using OpenGL. Luma Labs Genie, 3D CSM, Skybox Blockade Labs, SIMENS Digital Twin Tool.	EC-Independent
15	UE23CS342BA8	Cloud Security	4	0	0	4	4	AWS Security Services	EC-Independent
Elective–IV									
16	UE23AM343BB1	Natural Language Processing with Deep Learning	4	0	0	4	4	Python, Pytorch, and NLP libraries such as Spacy, NLTK, Open AI, andHuggingface	EC-Independent
17	UE23AM343BB2	Deep Reinforcement Learning	4	0	0	4	4	Pytorch, Python Libraries and tools.	EC-Independent
18	UE23AM343BB3	Deep Learning for Images	4	0	0	4	4	TensorFlow, PyTorch, OpenCV, Jupyter Notebooks.	EC-Independent
19	UE23AM343BB4	Machine Learning for Finance	4	0	0	4	4	TensorFlow, PyTorch, OpenCV, Jupyter Notebooks.	EC-Independent
20	UE23AM343BB5	Large Language Models Agent	4	0	0	4	4	HuggingFace, LangChain, CrewAI, OpenAI, FAISS, DevOps: Docker, GitHub Actions, Deployment: Streamlit, FastAPI.	EC-Independent
21	UE23CS343BB6	Information Security	4	0	0	4	4	SEED Labs VM, Scapy, Burp Suite, Metasploit, Nmap,etc	EC-Independent
22	UE23CS343BB7	Mobile Autonomous Robotics	4	0	0	4	4	Wireshark, Yersinia, VoIP Hopper, Bettercap,aircrack-ng	EC-Independent
23	UE23CS343BB8	Security for IoT	4	0	0	4	4	Wireshark, Yersinia, VoIP Hopper, Bettercap, aircrackng	EC-Independent

24	UE23CS343BB9	Applied ML in IoT	4	0	0	4	4	Arduino Nano 33 BLE sense Arduino IDE Google Colab	EC-Independent

ELECTIVES TO BE OPTED FOR SPECIALIZATION

Sl. No.	SPECIALIZATION	ELECTIVE–III	ELECTIVE–IV
A	Applied and Advanced Machine Learning – (AAML)	UE23AM342BA1, UE23AM342BA2, UE23AM342BA3, *UE23AM342BA4	UE23AM343BB1, UE23AM343BB2, UE23AM343BB3. **UE23AM342BA5
B	AI for Industry (AII)	*UE23AM342BA4 **UE23AM342B5	UE23AM343BB4, UE23AM343BB5
C	Cyber Security & Connected Systems (CSCS)	UE23CS342BA5, UE23CS342BA6, UE23CS342BA7, UE23CS342BA8.	UE23CS343BB6, UE23CS343BB7, UE23CS343BB8, UE23CS343BB9.

Note : * and ** are offered both as part of AAML and AII

VII SEMESTER (2022-26 BATCH)

Sl No.	Course Code	Course Title	Hours/week				Credits	Course Type
			L	T	P	S	C	
1	UE22AM461A	Capstone Project Phase-III	0	0	16	4	4	PW
2	U22AM421AX	Special Topic/ Directed Independent Study - Swayam/MOOC	2/4/4			2	6	ST
Total			**2/4/4**	**2/4/4**	**20**	**6**	**10**	

VIII SEMESTER (2022-26 BATCH)

Sl. No.	Course Code	Course Title	Hours/week				Credits	Course Type
			L	**T**	**P**	**S**	**C**	
1	UE22AM421B	Capstone Project Phase-4	0	0	8	2	2	PW
2	UE22AM421XB	Special Topic/ Directed Independent Study– Swayam/MOOC.	2	0	0	2	2	ST
3	UE22AM461XB	Internship	0	0	12	6	6	PW
Total			**2**	**0**	**20**	**10**	**10**	

<table>
<tr><td>Course Code</td><td>UE25CS151A</td><td>Course Title</td><td colspan="5">Python For Computational Problem Solving</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>2</td><td>5</td><td>5</td></tr>
<tr><td>Semester</td><td>1</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Python interpreter 3.8 and above.
IDLE and any IDE like Jupyter.
Copilot aided teaching</td><td>Desirable Knowledge</td><td colspan="5">-</td></tr>
<tr><td>Prelude</td><td colspan="6">Python is an easy to learn, general-purpose, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">• Understand the syntax and semantics of Python programming language.
• Apply the process of structuring the data using lists, tuples, sets and dictionaries.
• Utilize the use of built-in functions to navigate the file system.
• Explore various paradigms of programming and implement the Object-Oriented Programming concepts in Python.</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Introduction
Computer, Computational Problem Solving, Algorithm, Computer Hardware &Software(Application Software, System Software - Operating System),Syntax, semantics and program translation. Introduction to Python Programming Language, IDLE Python Development Environment, Output function, variables, types and id, input function, operators and expressions, Control structures. Standards and Guidelines for pythonic solution.
14 Hours
Unit 2: Collections & Functions
Lists, Tuples, Dictionaries, Sets, Strings, and text file manipulation: reading and writing files. Functions: Definition, call, Positional and keyword parameter, Default parameters, Variable number of arguments.
14 Hours
Unit 3: Functions, GUI, Modules, Testing and Debugging
Recursion, Call-backs, Closure, Decorators, generators. Graphical User Interface with WxPython, import mechanisms. Introduction to modules – Numpy, PyImage, nltk, sklearn, Regular Expressions – re, Testing-Pytest , Function testing with Doctest, pdb debugger commands, Debugging using IDLE.
14 Hours
Unit 4: Functional & Object Oriented Programming
Lambda function, Map, filter,and reduce,max,min,Zip, listcomprehension.
Classes and objects-inheritance, polymorphism, iterators,Error handling & Exceptions-try,except and raise, exception propagation.
14 Hours</td></tr>
</table>

Laboratory	• Exploring IDLE, • Programs on Input Output Functions, Operators and Expressions and Usage of Libraries • Programs on Control Structures. • Programs on Collections(Lists,Tuples,Sets,Dictionaries,Strings). • Programs on Files and File manipulations. • Programs on Functions, Recursion and Callback • Programs on Functional Programming. • Programs on Object Oriented Programming. • Programs on Pytest and pdb and one application module. • GUI programming (using wxPython)
TextBook(s):	1. Allen B.Downey,"Think Python:How to Think Likea Computer Scientist",2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf (Chapters 13, 15, 16, 17, 18) (Download pdf/html files from the above link). 2. Automate the Boring Stuff with Python", Al Sweigart,1 st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/). 3. "Python Programming using Problem Solving approach", Reema Thareja, Second Edition, Oxford University Press,2017. 4. "Fundamentals of Python Programming" by Richard L. Halterman, Course Technology ptr, 2018.
Reference Book(s):	1:"Introduction to Computer Science Using Python: A Computational Problem-Focus", Charles Dierbach, Wiley India Edition,John Wiley, 2015. 2. "Learn python Programming", Fabrizio Romano, 2nd Edition,Packet Publishing,2018. 3. "Fundamentals of Python: First Programs",Kenneth A.Lambert,Cengage,2019. 4. "Introduction to Computation and Programming Using Python: With Application to Understanding Data",John V. Guttag, MIT Press, MIT with Library of Congress Cataloguing- in-Publication Data, 2016. 5. "WxPython in action",By NOEL RAPPIN ROBIN DUNN.
Course Outcome	• Develop efficient programs using the Python language. • Demonstrate the ability to create and manipulate lists, tuples and dictionaries. • Perform common file system operations using built-in Python functionalities. • Apply different paradigms of programming and interpret the concepts of Object-Oriented Programming as used in Python.

Course Code	UE25CS151B	Course Title	Problem Solving with C				
Category	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	2	5	5
Semester	2	Type of Course	Core				
AITools /Tools/Languages	C Programming Language GCC Compiler, GDB Debugger AI Tools- Code 5, Deep Code, Codium, AIcode Helper	Desirable Knowledge	-				
Prelude	The Problem Solving with C course introduces students to fundamental techniques for solving Common computational problems. Students will learn to analyse problem statements, design appropriate algorithms, and implement efficient solutions. Emphasis is placed on using C language constructs to develop well-structured and maintainable programs, thereby laying a strong foundation in both problemsolving and programming skills.						
Course Objectives:	• Acquire Develop the ability to solve relevant and logical problems using a computing machine through algorithmic thinking. • Translate algorithmic solutions into programs by effectively utilizing C programming language constructs. • Acquire a solid understanding of C language syntax, semantics, and its supporting ecosystem. • Recognize and appreciate the nuances of C standards and understand their implications on program behaviour.						
ourse Contents	**Unit1: Problem Solving Fundamentals** Introduction to Programming, Salient Features of 'C', Program Structure, Variables, DataTypes & range of values ,Qualifiers, Operators and Expressions, Control Structures, Input/Output Functions, Language Specifications-Behaviors, Single character input and output, Coding standards and guidelines. **14 Hours** **Unit2: Counting, Sorting and Searching** Arrays–1D and 2D,Pointers, Pointer to an array, Array of pointers, Functions, Callback, Storage classes, Recursion, Searching, Sorting, Functions, GUI, Modules, Testing and Debugging. **14 Hours** **Unit3:Text Processing and User-Defined Types** Strings, String Manipulation Functions & Error handling, Command line arguments, Dynamic Memory Management functions & Error handling, Structures, #pragma, Array of Structures, Pointer to structures, Passing Structure and Array of structure to a function, Bit fields, Unions, Enums, Lists, Stack, Queue, Priority Queue. **14 Hours** **Unit4: File Handling and Portable Programming** File IO using redirection, File Handling functions of C, Searching, Sorting, Headerfiles, Comparison of relevant User defined and Built-in functions, Variable Length Arguments, Environment variables, Preprocessor Directives, Conditional Compilation. **14 Hours**						

Laboratory	• Programs on IO, Operators and Control structures. • Programs on Arrays and Pointers, Functions using arrays and pointers. • Programs on Strings, Structures and Dynamic Memory Management functions • Programs on Structures and Array of structures. • Programs on Inclusion of files using redirection operator. • Programs using File handling functions in C –Sorting and Searching using Array of pointers to structures. • Implementation of Ordered List • Implementation of Priority based scheduling. • Programs on unions, enums and Preprocessor directives.
TextBook(s):	1:"The C Programming Language", Brian Kernighan and Dennis Ritchie, Prentice Hall PTR, 2nd Edition, 1988.
Reference Book(s):	1: "How To Solve It By Computer", R G Dromey, Pearson, 2011. 2: "C Programming: A Modern Approach",2nd Edition by K.N.King, W.W.Norton & Company. 3: "Learn C the Hard Way": Zed Shaw's Hard Way Series, 1st Edition. 4: "C Puzzle Book" by Alan R. Fever, Pearson Education. 5. "Expert C Programming: Deep C Secrets" by Peter Van Der Linden, Pearson Education.
Course Outcome	• Understand and apply algorithmic solutions to counting and other basic computational Problems using appropriate C construct. • Understand, analyse, and implement fundamental sorting and searching techniques. • Apply string manipulation and text processing methods using arrays, pointers, and functions. • Design and implement user-defined data types using structures, unions, and other mechanisms in C, and demonstrate the ability to read from and write to secondary storage in a portable manner.

<table>
<tr><td>Course Code</td><td>UE24CS251A</td><td>Course Title</td><td colspan="5">Digital Design & Computer Organization</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>2</td><td>5</td><td>5</td></tr>
<tr><td>Semester</td><td>3</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AITools /Tools/Languages</td><td>Icarus Verilog Simulator, GTK Wave waveform viewer.(Open–Source Tool)
Logisim- for circuit simulation, Logically- for circuit simulation</td><td>Desirable Knowledge</td><td colspan="5">Basic Electronics</td></tr>
<tr><td>Prelude</td><td colspan="6">This course focuses on the structure, design, and operation of a computer system at different levels of abstraction. The digital design part of the course describes low level digital logic building blocks while the computer organization part explains the structure and operation of microprocessors.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● Fundamental (combinational and sequential) building blocks of digital logic circuits.
● Design of more complex logic circuits such as adders, multipliers and register files.
● Design of Finite State Machines based on problem specification.
● Construction using above logic circuits, of a microprocessor, and its functioning at the clock cycle level.</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Gate-Level Minimization and Combinational logic-1
Introduction, The map method Four variable K-map, Product of Sums simplification, Don’t Care conditions, NAND and NOR implementation, Combinational circuits, Analysis procedure Design Procedure, Combinational logic-1: Binary Combinational logic: Adder- Subtractor, Decimal Adder, Binary multiplier, Magnitude comparator Decoders Encoders, Multiplexers.
14 Hours

Unit 2: Synchronous Sequential Logic-I
Synchronous Sequential Logic: Introduction, Sequential circuits, Storage elements: Latches, Flip flops, Analysis of clocked sequential circuits, State reduction and assignment, Design procedure Registers and counters: Registers, Shift register, Ripple counters, Synchronous counters, Other counters.
14 Hours
Unit 3: Basic structure of computers, Standard I/O interface, Interrupts, Memory System
Computer Types, Functional Units: Input Unit, Memory Unit, ALU, Output Unit, Control Unit, Basic operational concepts, Number representation and arithmetic Operations, Character representation, Memory locations and addresses, Memory Operations, Instruction and instruction sequencing ,Addressing modes, Assembly Languages , Accessing I/O Devices, Interrupts, Standard I/O Interfaces. Semi conduct or RAM memories.
14 Hours

Unit 4: Arithmetic Processing Unit and Control Unit Design Arithmetic:
Multiplication of Positive numbers, Signed operand Multiplication, Fast multiplication, Integer division, floating point numbers operation and Architecture, Some fundamental concepts, Execution of a complete instruction, Multiple Bus Organization, Hardwired control , Single- cycle, Case Study: Multi-cycle processor data path and control.
14 Hours</td></tr>
</table>

Laboratory	• Implementation of Basic gates. • Implementation of Half adder and full adder. • Implementation of Decoder, encoder and Priority encoder. • Implementation of 4-to-1 multiplxer and 1-to-4 demultiplexer. • Implementation of 4-bit magnitude comparator • Implementation of BCD adder. • Implementation of Flip-Flops (JK,T and D). • Implementation of counters (Ripple,updown). • Implementation of counters (Ring,Jonson). • Implementation of Universal Shift register. • Implementation of 4-bit serial adder. • Implementation of Sequential Binary multiplier.
TextBook(s):	1. "Digital Design",M Morris Mano, Michael D Ciletti, Pearson, 5th Edition, 2012. 2. "Computer Organization", Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGrawHill, 5th Edition,2002.
Reference Book(s):	1. Digital Design & amp; Computer Architecture, David Money Harris, Sarah L.Harris, 2nd Edition, Elsevier, 2013. 2. Computer Organization and Design, David A.Patterson, JohnL.Hennessey 5th Edition, Elsevier.
Course Outcome	• Perform analysis of given synchronous digital logic circuit. • Design and implement small to medium scaled at a path logic circuits from given specification. • Design and implement control logic using Finite State Machines. • Understand hardware level microprocessor operation, providing a foundation for the higher layers.

<table>
<tr><td>CourseCode</td><td>UE24CS252A</td><td>CourseTitle</td><td colspan="5">Data structures & its Applications</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>2</td><td>5</td><td>5</td></tr>
<tr><td>Semester</td><td>3</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>C-Programming language Visu Algo (Interactive Visualizations), Algorithm Visualizer (AI Explanations)</td><td>Desirable Knowledge</td><td colspan="5">Problem Solving with C</td></tr>
<tr><td>Prelude</td><td colspan="6">This course provides a comprehensive introduction to fundamental data structure concepts, emphasizing both their theoretical foundations and practical applications. It begins with abstract representations of data and explores how these abstractions facilitate effective problem solving. The course then bridges the gap between theory and practice by demonstrating how these concepts can be concretely implemented using a programming language. Equal importance is given to understanding the underlying principles, their implementation techniques, and their real-world applications, enabling students to not only grasp the concepts but also apply them effectively in software development.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● To introduce the fundamental concepts and methodologies for analyzing and designing data structures, fostering problem-solving skills for efficient data storage, retrieval, and manipulation.
● To explore and utilize various data structures such as arrays, linked lists, stacks, queues, trees, heaps, and graphs, emphasizing their suitability for different computational tasks and operations.
● To implement core operations—insertion, deletion, searching, and modification—across a variety of data structures in both linear and non-linear forms.
● To design and develop applications using appropriate data structures, demonstrating the ability to select and apply the most suitable structure to solve real-world problems effectively.</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Linked List and Stacks
Review of C, Static and Dynamic Memory Allocation. Linked List: Doubly Linked List, Circular Linked List – Single and Double, Multilist: Introduction to sparse matrix (structure). Skip list Case study: Dictionary implementation using skip list Stacks: Basic structure of a Stack, Implementation of a Stack using Arrays & Linked list. Applications of Stack: Function execution, Nested functions, Recursion: Tower of Hanoi. Conversion & Evaluation of an expression: Infix to postfix, Infix to prefix, Evaluation of an Expression, Matching of Parenthesis.
14 Hours

Unit 2: Queues and Trees
Queues & Dequeue: Basic Structure of a Simple Queue, Circular Queue, Priority Queue, Dequeue and its implementation using Arrays and Linked List. Applications of Queue: Case Study – Josephus problem, CPU scheduling- Implementation using queue (simple /circular). General: N-ary trees, Binary Trees, Binary Search Trees (BST) and Forest: definition, properties, conversion of an N-ary tree and a Forest to a binary tree. Traversal of trees: Preorder, Inorder and Postorder.
14 Hours

Unit 3: Application of Trees and Introduction to Graphs
Implementation of BST using arrays and dynamic allocation: Insertion and deletion operations, Implementation of binary expression tree., Threaded binary search tree and its implementation. Heap: Implementation using arrays. Implementation of Priority Queue using heap - min and max heap. Applications of Trees and Heaps: Implementation of a dictionary / decision tree (Words with their meanings). Balanced Trees: definition, AVL Trees, Rotation, Splay Tree, Graphs: Introduction, Properties, Representation of graphs: Adjacency matrix, Adjacency list. Implementation of graphs using adjacency matrix and lists. Graph traversal methods: Depth first search, Breadth first search techniques. Application: Graph representation: Representation of computer network topology. 14 Hours</td></tr>
</table>

	Unit 4: Applications of Graphs, B-Trees, Suffix Tree and Hashing Application of BFS and DFS: Connectivity of graph, finding path in a network. Suffix Trees: Definition, Introduction of Trie Trees, Suffix trees. Implementations of TRIE trees, insert, delete and search operations. Hashing: Simple mapping / Hashing: hash function, hash table, Collision Handling: Separate Chaining & Open Addressing, Double Hashing, and Rehashing. Applications: URLs decoding, Word prediction using TRIE trees / Suffix Trees. **14 Hours**
Laboratory	1: Linked List and advanced operations. 2: Stack and applications based on it. 3: Queue and applications based on it. 4: BinaryTree, BinarySearch Tree and applications based on it. 5: Graph Data structure and applications based on it. 6: Hashing Techniques.
TextBook(s):	1:"DataStructures using C/C++",Langsum Yedidyah, MosheJ Augenstein, Aaron MTenenbaum Pearson Education Inc, 2nd edition,2015.
Reference Book(s):	1:"Data Structures and Program Design in C",Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019
Course Outcome	• Select and apply appropriate data structures for solving problems in a variety of application domains. • Implement fundamental data structures and their operations using suitable programming constructs and techniques. • Utilize data structures effectively in competitive programming to design optimal and efficient solutions. • Design and develop efficient software systems by leveraging an in-depth understanding of data structure principles and their applications.

<table>
<tr><td>Course Code</td><td>UE24MA241A</td><td>Course Title</td><td colspan="5">Mathematics for Computer Science Engineers</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>3</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Jupyter Notebook, Python, Pandas, Matplotlib, Scipy, Seaborn, BeautifulSoup, Numpy, Scikit learn. R (R Studio) and Python, Orange</td><td>Desirable Knowledge</td><td colspan="5">-</td></tr>
<tr><td>Prelude</td><td colspan="7">This course covers both Descriptive statistics to understand the data and the inferential statistics which seeks to infer something about a population on the basis of a statistical sample and build a simple linear regression model. This course will also cover engineering optimization techniques like unconstrained, constrained, and discrete optimization techniques.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Provide insights about the basic roles of a Data Scientist. Develop a greater understanding of the importance of summary statistics and Random Variables and their Distributions.
● Provide students with knowledge of Confidence Intervals and their importance. Make inferences about the population parameters using sample data and test it to draw meaningful conclusions.
● Provide an understanding on the importance and techniques of predicting a relationship between the two sets of data and determine the goodness of fit model.
● Provide an understanding on the importance and techniques of engineering optimization.</td></tr>
<tr><td>Course Contents</td><td colspan="7">Unit 1: Descriptive Statistics and Sampling Distributions
Sampling and Descriptive Statistics: Introduction, Motivating Examples and Scope. Statistics: Introduction, Types of Statistics, Types of Data, Types of Experiments –Controlled and Observational study, Sampling: Sampling Methods, Sampling Errors, Case Study.
Probability and commonly used Distributions: Chebyshev's inequality, Linear function of random variables, Normal Probability Plots, Introduction to Generation of Random Variates and mention the types, Acceptance- Rejection method, Sampling Distribution, The Central Limit Theorem and Applications.
Self-Learning: Generation of Random Variates -Inverse Transform Method.
Applications:
1. Poisson distribution, calculation of number of calls received in a specified time duration in call centers.
2. Variance, standard deviation, identifying the customer satisfaction in online shopping.
3. Central limit theorem, Load Balancing in distributed systems and internet traffic prediction
4. Sampling mean, Estimating database query response times.
14 Hours

Unit 2: Point Estimation, Confidence Intervals and Hypothesis Testing
Commonly used Distributions: Principles of Point Estimation - Mean Squared Error for Bernoulli, Binomial, Poisson, Normal, Maximum Likelihood Estimate for Bernoulli, Binomial, Poisson, Normal and Case Study. Introduction to multivariate normal distribution, MAP distribution.
Confidence Intervals: Interval Estimates for Mean of Large and Small Samples, Student's t Distribution, Interval Estimates for Proportion of Large and Small Samples, Confidence Intervals for the Difference between Two Means, Interval Estimates for Paired Data. Factors affecting Margin of Error,</td></tr>
</table>

	Hypothesis Testing: Introduction, Large-Sample tests for a Population Mean, Drawing conclusions from the results of Hypothesis tests, Tests for population proportion, Small-Sample tests for a Population Mean, Case Study. **Self-Learning**: Confidence interval for difference between two proportions. **Applications:** 1. t-distribution, confidence interval, students' performance analysis based on hours of study. 2. z-test, application form processing in banking system. 3. Hypothesis testing, randomly trained students' placement into tier-I and tier-II companies. **14 Hours** **Unit 3: Hypothesis Testing and Multiple Linear Regression** Hypothesis Testing: Distribution Free Tests, Chi-squared Test,Fixed Level Testing, TypeI and TypeII Errors, Power of a Test, Factors Affecting Power of a Test. Correlation and Simple Linear Regression: Introduction, Correlation, the Least Square Lines, Predictions using regression models - Uncertainties in Regression Coefficients, Checking Assumptions and transforming data. Multiple regression: Introduction to the Multiple Regression Model, Case Study. Self-Learning: F test for equality of Variance. **Applications:** 1. Linear regression, stock market prediction 2. Using Chi-Square Test, Analyzing the association between vaccination and recovery of the patients considering COVID data. 3. Chi-Square Test and Test of Independence, Analyzing the relationship between gender and preference for a product purchase. 4. Identifying Type1 and Type2 Errors in Spam mail classification. **14 Hours** **Unit 4: Engineering optimization** Introduction to optimization, Types of optimization, Mathematical concepts of objective function, constraints and decisions, Constrained Linear Optimization- LPP – Formulation, Graphical Method, The Simplex Method, Discrete variable optimization - Integer Programming Problem, Branch and Bound Method, Constrained Non-linear Optimization-Lagrangian Multipliers method, Kuhn Tucker Conditions, Unconstrained non-linear optimization -Gradient Descent Method, Evolutionary computation -Genetic Algorithms. **Self-Learning:** Newton Rapson Method. **Applications:** 1. Minimize a Loss functions in Neural Networks using Batch gradient descent (Unconstrained Optimization) 2. Lagrange Multipliers to find local maxima and minima of a function subject to equations constrains (Constrained Optimization) 3. Case study on Bayesian Optimization with Discrete Variables (Discrete Variable optimization) 4. Use Genetic Algorithms to optimize Production Scheduling in a manufacturing environment, focusing on minimizing total production costs while meeting job deadlines and machine constraints. Evaluate the GA's effectiveness against traditional scheduling methods. **14 Hours**
TextBook(s):	1. "Statistics for Engineers and Scientists", William Navidi, McGraw Hill Education, India, 4th Edition, 2015. 2. "Engineering Optimization Theory and Practice", Singiresu S. Rao, John Wiley & Sons, Inc, 4th Edition,2009.

Reference Book(s):	1. "Fundamentals of Data Science", Sanjeev J.Wagh, Manisha S. Bhende, and Anuradha D. Thakare, First edition, CRC Press, 2022. 2. "Sampling-Design and Analysis", Sharon L.Lohr, 2nd edition (stats), Cengage,2010. 3. "Data Science from Scratch", Joel Grus, O' Reilly, 1st Edition,2015. 4. "A Hands-On Introduction to Data Science", Chirag Shah, Cambridge University Press, First edition, 2020.
Course Outcome	● Use Python and other tools to extract, clean and analyze data from several data sources(files, web) analyze an extremely large dataset and perform exploratory data analysis to extract meaningful insights. ● Analyze a real-world problem and solve the same with the knowledge gained from various distribution studies. ● Develop and test a hypothesis about the population parameters to draw meaningful conclusions and fit a regression model to data and use it for prediction. ● Analyze a real-world problem and solve the same with the knowledge gained from various engineering optimization techniques.

Course Code	UE24CS241A	Course Title	Web Technologies				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	3	Type of Course	Core				
AI Tools /Tools/Languages	HTML, CSS, JavaScript, MERN Technologies GitHub Copilot and Tabnine	Desirable Knowledge	-				
Prelude	Web Technologies course demonstrates an in-depth understanding of the technologies necessary for designing and developing a rich web application in an efficient way.						
Course Objectives:	• Basic web technologies and building blocks of a website using HTML, CSS, JavaScript and Advanced JavaScript. • The core concepts of HTML5, JQuery and AJAX, MERN (MongoDB, ExpressJS, ReactJS and NodeJS) stack and build an UI of the application using React JS. • Building a multi-tier application by interfacing UI to NodeJS. • Integrate database MongoDB through ExpressJS Framework and Web services.						
Course Contents	**Unit 1: HTML, CSS and Client-Side Scripting** Introduction to Web Architecture and Web protocols (HTTP Request Response Formats, URLs), Basic Mark-ups & syntax, HTML elements & attributes, Web Form, HTML5 (New Tags, Inputs, Elements and Controls), CSS3.0 - Styles and Style sheets, Selectors, Style properties, Box Model, JavaScript Basics (variables, scope, Built-in Objects), JavaScript objects, DOM Manipulations, Events and Event Handling in JavaScript **14 Hours** **Unit 2: HTML5 and ReactJS** HTML5 (APIs), Audio, Video and Progress, Geolocation, Callbacks & Promises, Single Page Application, XML Vs JSON, , Async/Await, ReactJS - MERN Introduction, JSX, rendering of Elements, Set-up React Installation, React Component and JSX, React Component Styling, Properties, States and Context – (Arrays), How state works in functional component. **14 Hours** **Unit 3: ReactJS and NodeJS** State Management – Complex components, Keys,Event Handling, React Forms, React Hook – useState, useRefs and UseEffect, React Hook – useContext and useReducer, React Router, Introduction to NextJS, Understanding Node JS Architecture, callbacks, Node Modules, Buffers and Streams, File system, Axios API,Nodejs. **14 Hours** **Unit 4: MongoDB** MongoDB-Documents, Collections, Reading and Writing to MongoDB, MongoDB NodeJS Driver, running a react application on NodeJS, React Router. ExpressJS – Introduction to Web services and REST API's, Express Framework Overview, Routing and URL building, Error Handling, Express Middleware, Form Data and File Upload. **14 Hours**						

TextBook(s):	1. "Learning PHP, MySQL & JavaScript", Robin Nixon.,5th edition, O'Reilly Media, Inc.ISBN: 9781491978917, 2018. 2. "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", Vasan Subramanian, Apress, 2017.
Reference Book(s):	1. "Beginning Node.js, Express & MongoDB Development", GregLim, July 2019. 2. "Learning React, Functional Web Development with React and Redux", Alex Banks and Eve Porcello, O'Reilly Media, May 2017.
Course Outcome	● Understand basic web technologies like HTML, CSS and JavaScript. ● Achieve rich user experience by implementing HTML5 features and Asynchronous communication using MERN stack layers (MongoDB, ExpressJS, ReactJS and NodeJS) and Create rich User Interface using React JS ● Understand and Integrate the UI with NodeJS. ● Create RESTful Web services using ExpressJS and MongoDB database.

<table>
<tr><td>Course Code</td><td>UE24CS242A</td><td>Course Title</td><td colspan="5">Automata Formal Languages & Logic</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>3</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AITools /Tools/Languages</td><td>PLY,JFLAPAI Tool-https://www.opentrain.ai/glossar y/automata-theory https://www.yeschat.ai/gpts-9t55kXLEUfB-Automata-Expert</td><td>Desirable Knowledge</td><td colspan="5">-</td></tr>
<tr><td>Prelude</td><td colspan="6">The course introduces fundamental concepts in Automata and Formal Languages and their application to Logic. The course covers the notions of Finite State Automaton, Regular expression, Push down Automaton, Context Free Languages and Turing Machines. These abstract and formal models and their usage in Propositional and First Order Predicate Logic, allow for solving problems in Formal language Generation and Recognition.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● Teach students to construct basic machines like DFA, NFA which represent Regular Languages, Regular Expressions, and Regular Grammars and to identify Non – Regular Languages.
● To familiarize students to construct Context Free Languages, to construct Push down Automata which represent Context Free Languages, to conver the given grammar to various normal forms and to make use of Membership Algorithm.
● Teach students to understand closure properties of Context Free Languages, to identify Non – Context Free Languages and to construct Turing Machines and
● To familiarize students with concepts like Recursively Enumerable languages, Recursive Languages, Undecidable Problems. And to familiarize notions of mathematical logic: logical notations (syntax) and how to assign meaning to them (semantics).</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Introduction
Mathematical Preliminaries and Notation, Three Basic Concepts. Finite Automata: Deterministic Finite Accepters,Non-DeterministicFiniteAccepters,EquivalenceofDeterministicandNon-DeterministicFinite Accepters, Reduction of the number of states in Finite Automata. Regular Expressions, Connection between Regular Expressions and Regular Languages, Usage of Regex in practical programming/Regex in practice, Regular Grammars.
14 Hours
Unit 2: Regular Languages and Context Free Languages
Properties of Regular Languages: Closure Properties of Regular Languages, Elementary Questions about Regular Languages, Identifying Non-Regular Languages. Definitions of PDA and CFL, Deterministic Pushdown Automata, Non-Deterministic Pushdown Automata, Pushdown Automata and Context Free Languages, Context Free Grammars.
14 Hours
Unit 3: Properties of Context Free Languages and Turing Machine
Parsing and Ambiguity: Parse tree, ambiguous grammars, removing ambiguity, inherently ambiguous grammars. Simplification of Context–Free Grammars and Normal Forms: Methods for Transforming Grammars, Two Important Normal Forms, A Membership algorithm for Context Free Grammar. Properties of Context-Free Languages: Closure Properties and Questions about Context–Free Languages, Pumping Lemma for Context–Free Languages. Turing Machines: The Standard Turing Machine, Constructing Turing Machines, Combining Turing Machines for Complicated Tasks, Turing's Thesis. 14Hours</td></tr>
</table>

	Unit4: Undecidability and design of a declarative language. Hierarchy of Formal Languages and Automata: Recursive and Recursively Enumerable Languages, the Chomsky Hierarchy. Limits of Algorithmic Computation: Some Problems that cannot be solved byTuring Machines, Undecidable Problem for Recursively Enumerable Languages, idea of reduction, Rice's theorem, its proof and its applications. A very simple Logic, Syntax, Semantics, A simple knowledge Base, A simple inference procedure. Propositional Theorem Proving: Inference and Proofs, Proof by Resolution, Conjunctive Normal Form, A resolution algorithm. Syntax and Semantics of First Order Logic: Models for First Order Logic Symbols and interpretations, Terms, Atomic Sentences, Complex Sentences Quantifiers, Equality, Numbers, sets and Lists. Example - The electronic circuits' domain **14 Hours**
TextBook(s):	1:"An Introduction to Formal Languages and Automata", PeterLinz, Jones and Bartlett,New Delhi, India, 6th Edition, 2016. 2:"Artificial Intelligence – A Modern Approach", Stuart Russell and Peter Norvig, Pearson, 3rd Edition(Paperback), 2016.
Reference Book(s):	1: "Theory of Computation", Michael Sipser, Cengage Learning, New Delhi, India, 2008. 2: "Introduction to Automata Theory, Languages, and Computation", John E Hopcroft, Rajeev Motwani, Jeffrey DUllman, Pearson Education, New Delhi, India, 3rd Edition, 2009. 3:"Theory of Computation: A Problem–Solving Approach", Kavi Mahesh, Wiley India, NewDelhi, 2012.
Course Outcome	• Design simple machines like DFA, NFA, convert NFA to DFA and minimize a given DFA, construct regular expressions for different languages, verify that some languages are regular and some are not. • Analyze the difference between Regular Languages and ContextFree Languages, design Push Down automata, construct Context Free Grammars, and convert one form of the grammar to another form. • Enumerate the properties of ContextFree Grammars, verify that some languages are context free and some are not, design Turing Machines, and analyze the difference between acceptability and decidability. • Analyze the difference between Recursive and Recursively Enumerable Languages, Decidable Languages, Turing –Recognizable and Co-Turing –Recognizable, some problems that cannot be solved by Turing Machines, reduce one Undecidable Problem to another, Undecidable Problems for Recursively Enumerable Languages. And make use of Propositional Logic and Predicate Logic in knowledge representation and truth verification.

Course Code	UE24CS251B	Course Title	Microprocessor & Computer Architecture				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	2	5	5
Semester	4	Type of Course	Core				
AI Tools /Tools/Languages	ARM Simulator, Arduino Microcontroller kit, Para Cache simulator. Compute intensive Application are run on GPU systems.	Desirable Knowledge	UE24CS251A-Digital Design and Computer Organization				
Prelude	This course will give you an in-depth understanding of the inner-workings of modern digital computer systems and trade-offs present at the hardware-software interface. The course focuses on key topics in microprocessor such as the system architecture, low level programming aspects and interface with other key components. Also, the course will help in understanding the core computer architecture concepts such as multilevel in memory hierarchies, pipelining, and super scalar techniques. A desirable knowledge of Digital Design and Computer organization is required.						
Course Objectives:	• Introduce concepts of basic processor architecture and its design. • Understanding the concept of pipeline architecture and hazards. • Study of memory hierarchy, cache memory and its optimizations. • Introduce advanced concepts in processor architecture like multi-core/manycore Processor architectures.						
Course Contents	**Unit1: Architecture and Assembly Language Programming** Introduction, ISA Classification – RISC and CISC, Memory Addressing, Operands - Types and Size, Instruction Set-Operations, Control Flow, Instruction Encoding, Case Study-ARM/MIPS/x86Processor. **16 Hours** **Unit2: Pipelining** 3-Stage Pipelining, 5-Stage Pipelining, Pipeline Datapath and Control, Data Hazards –Forwarding vs. Stalling, Control Hazards, Branch Prediction Mechanisms and Exceptions, Performance Metrics. **12 Hours** **Unit3: Basics of Cache and Cache Optimization** Basics of Caches-Fully Associative, Direct Mapped and Set Associativity, Cache Performance, Basic Cache Optimization- Reduce in Miss Rate. Basic Cache Optimization- Reduce Miss Penalty, Reduce Hit Time. **14 Hours** **Unit4: Advances in Architecture** Introduction to Parallel Computing, Parallel Computer, Memory Architecture and Flynn's Taxonomy, Parallel Computing Concepts and Terminology, Design Issues & Constraints, performance in parallel computing: Amdahl's Law & Gustafson's Law , Multicore Processors and OpenMP programming, Realization of ML operations on Modern Architecture, Introduction to Heterogenous Parallel Computing, Introduction to GPU Computing. **14 Hours**						

Laboratory	● Introduction to ARM Simulator and sample programs. ● Implementation of Data Processing Instructions, Programs on usage of addressing modes. ● Programs on functions and software interrupts, Matrix Operations–Addition and Multiplication. ● Introduction to PARACACHE simulator–Direct Mapping, Associative Mapping. ● Introduction to Plugins interface. ● Project work.
TextBook(s):	1:"Computer Organization and Design",Patterson,Hennessey,5th Edition, Morgan Kaufmann, 2014. 2. "Computer Organization and Design –ARM Edition", Patterson, Hennessey, 4th Edition, Morgan Kaufmann, 2010. 3. "ARM System-on-Chip Architecture", Steve Furber,2nd Edition, Pearson India,2015. 4. "Programming Massively Parallel Processors A Hands-on Approach", Third Edition David B. Kirk Wen-mei W. Hwu, Morgan Kaufmann. 5. "Multicore and GPU Programming An Integrated Approach",Gerassimos Barlas, second edition, Morgan Kaufmann.
Reference Book(s):	1:"Computer Architecture: A Quantitative Approach", Hennessey, Patterson, 5th Edition, Morgan Kaufmann, 2011. 2:"The Definitive Guide to the ARM Cortex-M0 and CortexMO + processors", JosephYiu, 2nd Edition, Newnes, 2015.
Course Outcome	● Demonstrate ability to understand the design of different instruction sets like RISC/CISC And their addressing modes. ● Demonstrate the ability to understand the design of a pipelined processor and its Challenges. ● Demonstrate the use of tools to analyses the performance of programs on different Architectures. Design alternative memory hierarchy layouts and optimizations. ● Demonstrate and appreciate modern trends in architecture such as multicore Architectures.

Course Code	UE24CS252B	Course Title	Computer Networks				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	2	5	5
Semester	4	Type of Course	Core				
AI Tools /Tools/Languages	Wireshark, Python. Advanced-Computer Networking-Tutor Computer Network Professor (yeschat.ai)AI	Desirable Knowledge	-				
Prelude	This is a foundation course on Computer Networking which focuses on building blocks of the Internet. We trace the journey of messages sent over the Internet from an application residing on one host machine (source) to another (Destination) using a top-down layered approach. The course contents are organized based on TCP/IP Protocol stack.						
Course Objectives:	• To provide a comprehensive understanding of the fundamental principles and architectures of computer networks and the Internet. • To enable students to design, implement, and analyze network applications using socket programming and understand the role of application layer protocols. • To equip students with the knowledge to analyze and evaluate transport layer protocols, including their mechanisms for reliable data transfer and congestion control, and to introduce them to the concepts of Software Defined Networking (SDN). • To familiarize students with link layer protocols, error detection/correction techniques, and routing algorithms, enabling them to understand the operational aspects of local area networks and network routing.						
Course Contents	**Unit 1: Computer Networks and the Internet, Application Layer** Introduction to Computer Networks, Internet: A Nuts–and–Bolts Description, A Services Description, Protocol,TheNetworkEdge:AccessNetworks,PhysicalMedia,Introductiontophysicallayerdevices,The Network Core, Packet Switching, Circuit Switching, A Network of Networks, Delay, Loss, and Throughput in Packet–Switched Networks, Overview of Delay in Packet Switched Networks – Queuing Delay and Packet Loss, End–to–End Delay, Throughput in Computer Networks, The OSI Model and the TCP/IP Protocol Suite, Protocol Layers, The OSI Model, TCP/IP Protocol Suite. Network Application Principles: Network Application Architectures, Processes Communication, Transport Services available to Applications, Transport Services provided by Internet. **14 Hours** **Unit 2: Application Layer, Transport Layer – UDP** The Web, HTTP and HTTPS, Nonpersistent and persistent connection, HTTP Message Format, User Server Interaction: Cookies, Web Caching. DNS, The Internet's Directory Service: Services provided by DNS; How DNS works: DNS Records and messages; Peer to peer Applications; Socket Programming with TCP and UDP; Other Application Layer Protocols: FTP, SMTP, SNMP, Telnet, SSH. Introduction to Transport Layer Services: Relationship Between Transport and Network Layer, Overview of the Transport layer in the Internet, Multiplexing and Demultiplexing; Connectionless Transport UDP: UDP Segment Structure, UDP Checksum. Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocol. **14 Hours** **Unit 3: Transport Layer–TCP, Network Layer and Internet Protocol** Go–Back–N Protocol, Selective–Repeat; Connection Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Flow Control, TCP Connection Management, TCP Congestion Control. Numerical on TCP congestion control mechanisms–TCP Tahoe, Reno. Overview of Network Layer: Forwarding and routing, what's Inside a Router? Introduction to SDN, Components and Architecture of SDN, The Internet Protocol(IP) IPV4: Datagram Format, Fragmentation,						

	Addressing, NAT. **14 Hours** **Unit 4: Network Layer and Internet Protocol, Link Layer and LAN** Introduction to Network layer Protocols: DHCP,ICMP; IPv6 Protocol: Packet Format, Transition from IPv4 to IPv6; Introduction to Routing Algorithms: Link State: Dijkstra's algorithm and Distance Vector: Bellman–Ford Algorithm. Link layer – Error–Detection and Correction techniques, Parity checks, Internet Checksum, Cyclic Redundancy Check,and Multiple Access Protocols: CSMA/CD,CSMA/CA;SwitchedLAN: Link layer addressing and ARP, Ethernet: Link–layer switches. Retrospective: A Day in the Life of a Web Page Request. Physical Layer – Purpose, Signals to Packets, Transmission media. Wireless LANs: IEEE 802.11 LAN architecture. **14 Hours**
Laboratory	● Designing and Simulation of Network Topology using Cisco Packet Tracer. This experiment introduces students to Cisco Packet Tracer, enabling them to create basic network topologies, configure devices, and visualize data flow within a network. ● Understanding Persistent and Non-persistent HTTP Connections. Students use Wireshark to analyze HTTP traffic and understand the differences between persistent and non-persistent connections, focusing on real-world scenarios like web page loading. ● Exploring UDP with DNS and Sockets using Wireshark. This week is divided into two parts: analyzing DNS queries and responses to understand UDP traffic and implementing UDP socket communication for practical understanding. ● Congestion Window Plotting using NS2. Students simulate TCP congestion control using NS2 and plot the congestion window, learning how TCP adapts to network conditions. ● Understanding Hardware Components in Networking. Students explore physical networking components, such as routers,switches,and cables,gaining hands-on experience with hardware setup and troubleshooting.
TextBook(s):	1:"Computer Networking: A Top –Down Approach", James F. Kurose, Keith W. Ross, 7th Edition, Pearson Publication, 2017.
Reference Book(s):	1."TCP IP Protocol Suite", Behrouz Forouzan, 4th Edition, McGraw–Hill, 2010.
Course Outcome	● Analyze and apply fundamental networking concepts, including protocol layers, network architectures, and addressing schemes (IPv4, IPv6). ● Evaluate and implement key application layer protocols (HTTP,DNS,SMTP) and understand their role in network communication. Apply socket programming concepts with TCP and UDP to develop network applications. ● Design and analyze transport layer protocols(UDP,TCP), including reliable data transfer, flow control, and congestion control mechanisms, and evaluate their impact on network performance. Additionally, gain an introductory understanding of Software Defined Networking(SDN) concepts. ● Explain and apply link layer protocols(Ethernet, Wi-Fi), error detection/correction techniques, and routing algorithms (Dijkstra, Bellman-Ford) for efficient network operation.

Course Code	UE24CS241B	Course Title	Design and Analysis of Algorithm				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	4	Type of Course	Core				
AI Tools /Tools/Languages	C-Programming language Visu Algo (Interactive Visualizations), Algorithm Visualizer (AI Explanations)..	Desirable Knowledge	UE25CS151B-problem solving with C, UE24CS252B-Data Structures and its Applications				
Prelude	Algorithms play a key role in science and practice of computing. Learning algorithm design technique isa valuable endeavor from practical standpoint and algorithm design techniques have considerable utility as general problem-solving strategies, applicable to problems beyond computing. This course includes classic problems of computer science, application of design techniques and analysis in terms of time and space.						
Course Objectives:	• Learn various algorithm design techniques and apply appropriate algorithmic design techniques for specific problems. • Learn to design and analyze algorithms with an emphasis on resource utilization interm softime and space. • Learn to trade space for time in algorithmic design using input enhancement and per- structuring. • Learn the limitations of algorithmic power and techniques to handle these limitations.						
Course Contents	**Unit 1: Introduction and Brute Force** Algorithms: Fundamentals of Algorithmic Problem Solving, Important Problem Types. Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms. Brute Force: Selection Sort, Bubble Sort, Sequential Search, Brute Force String Matching, Exhaustive Search**.** **14 Hours** **Unit 2: Decrease–and–Conquer & Divide-and-Conquer** Decrease-and-Conquer: Decrease by constant number algorithms - Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms – Fake Coin Problem, Russian Peasant Multiplication, Josephus problem, Decrease-by-Variable-Size Algorithms – Computing a median and the selection problem. Divide-and-Conquer: Master Theorem, Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Complexity analysis for finding the height of BST, Multiplication of Large Integers, Strassen's Matrix Multiplication. **14 Hours** **Unit 3: Transform-and-Conquer Space and Time Tradeoffs & Greedy Technique** Transform and Conquer: Pre-sorting, Heap Sort, Red-Black Tree Construction and Time complexity Analysis for insert and search operation, 2-3 Trees and B Tree: insertion, deletion, searching, and time complexity analysis. Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching-Horspool and Boyer-Moore Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's						

	Algorithm and union-find algorithm, Dijkstra's Algorithm, Huffman Trees **14 Hours** **Unit 4: Limitations, Coping with the Limitations of Algorithm Power & Dynamic Programming Dynamic Programming:** Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, and NP Complete, NP-Hard Problems. Coping with the Limitations of Algorithm Power: Backtracking. **14 Hours**
TextBook(s):	1. "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi (Indian Version), 3rd Edition, 2012.
Reference Book(s):	1. "Introduction to Algorithms", Thomas H Cormen, Charles ELeiserson, Ronald L Rivest and Clifford Stein, Prentice Hall India, 3rd Edition, 2009. 2. "Fundamentals of Computer Algorithms", Horowitz, Sahni, Rajasekaran, Universities Press, 2nd Edition, 2007. 3. "Algorithm Design", Jon Kleinberg, Eva Tardos, Pearson Education, 1st Edition, 2006.
Course Outcome	● Identify the design technique used in an algorithm. ● Design and implement efficient algorithms for practical and unseen problems and analyze these algorithms using quantitative evaluation. ● Understand Time and Space Tradeoff's. ● Understand, and cope up with the limitations of algorithm power.

Course Code	UE24CS242B	Course Title	Operating System				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	4	Type of Course	Core				
AI Tools /Tools/Languages	C, Linux/Unix OS for system call implementation.	Desirable Knowledge	UE24CS252A-Data Structures and its Applications				
Prelude	This course focuses on fundamental operating systems concepts including various algorithms and trade-offs for efficient management of resources such as CPU, Memory, Storage and I/O. This course requires the student to have a desirable knowledge of Data Structures and its Applications.						
Course Objectives:	• Focus on fundamental Operating System concepts. • Provide an understanding of various components of the Operating System (OS). • Delve deeper into various algorithms and associated trade-offs for efficient resource management such as process, disk, and memory management. • Introduce design principles and trade-offs in the design of Operating Systems.						
Course Contents	**Unit 1: Introduction and Process Management** What Operating Systems Do, Operating-System Structure & Operations, Kernel Data Structures, Operating-System Services, Operating System Design and Implementation. **Shell programming:** Overview of bash shell programming – variables, control flow **Processes:** process concept, Process Scheduling, Operations on Processes, System calls for process management-fork (), vfork (), wait () and exec (), Life cycle of I/O request. **CPU Scheduling**: Basic Concepts, Scheduling Criteria, Scheduling Algorithms. Case Study: Linux Scheduling Policies. **Shell programming –**cron **14 Hours** **Unit 2: IPC, Threads and Concurrency** **IPC –** Introduction, Shared Memory systems, Message Passing, Communication in Client–Server Systems-Pipes, ordinary pipes and named pipes, system calls for shared memory, pipes and fifo's. **Threads:** Overview, Multicore Programming, Multithreading Models, Thread Libraries, Thread Scheduling. **Process Synchronization**: Background, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization- The bounded-Buffer Problem, The Readers–Writers Problem, The Dining-Philosophers Problem, Synchronization Examples-Synchronization in Linux. System calls for threads creation and synchronization-POSIX Threads. **Deadlocks:** System Model, Deadlock Characterization, Deadlock avoidance, Banker's Algorithm, Deadlock Detection. **14 Hours** **Unit 3: Memory Management** **Main Memory:** Background- Basic Hardware, Address Binding, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking and Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement Algorithms-FIFO, LRU, Optimal, Allocation of Frames, Thrashing. **14 Hours** **Unit 4: File and Storage Management** **File-System Interface:** File Concept, system calls for file operations-open(), read(),write(), lseek(), close() and system call to retrieve file attributes and file types-stat(), lstat(), Access Methods, Directory and Disk Structure, system calls for reading directories, system calls to create hard links (link()) and symbolic links-						

	symlink().**File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, File Sharing, Protection. **Storage management**: Overview of Mass-Storage Structure, Disk Scheduling, Swap-Space Management, RAID Structure. **System Protection:** Goals, Principles and Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control. **Shell programming -** awk, sed **14 Hours**
TextBook(s):	1. "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 10th Edition, John Wiley & Sons, Global Edition, 2023. 2. "Advanced Programming in the Unix Environment", Richard Stevens and Stephen A Rago, Pearson, 3rd edition, 2017.
Reference Book(s):	1:"Operating Systems, Internals and Design Principles", William Stallings, 9th Edition, Pearson, 2018. 2: "Modern Operating Systems", Andrew S Tanenbaum, 3rd edition, Pearson, 2007. 3: "Learning the bash shell", Cameron Newham, 3rd edition, O'Reilly, 2005.
Course Outcome	● Understand the principles and modules of Operating Systems. ● Understand the design of various algorithms for scheduling and their relative performance. ● Understand the concept of Deadlocks that typically occur in OS. Deadlocks - Avoidance and Detection. ● Implement Operating Systems Concepts related to process management, Concurrent processes, Threads and Memory Management.

<table>
<tr><td>Course Code</td><td>UE24MA241B</td><td>Course Title</td><td colspan="5">Linear Algebra and its Applications</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>4</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Python IDE, MathGPT, Deepseek</td><td>Desirable Knowledge</td><td colspan="5">-</td></tr>
<tr><td>Prelude</td><td colspan="7">This is a basic subject on matrix theory and linear algebra. Emphasis is given to topics that will be useful in Engineering discipline, including system of equations, vector spaces, eigenvalues, similarity, and positive definite matrices. The course provides hands-on experience in basic programming concepts using PYTHON/MATLAB for solving problems relevant to these areas.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">• To teach students how to use linear algebra as a powerful tool for computation.
• To show how these computations can be conceptualized in a geometric framework.
• To give a gentle introduction to the theory of abstract vector spaces.
• To visualize solution to linear system of equations with different approaches using Python/Matlab.</td></tr>
<tr><td>Course Contents</td><td colspan="7">Unit 1: Matrices, Gaussian Elimination

Introduction, The Geometry of Linear Equations, Gaussian Elimination, Singular Cases, Elimination Matrices, Triangular Factors -LU decomposition and Row Exchanges, Inverses and Transposes, Inverse by Gauss -Jordan method, Symmetric Matrices. Self-Learning Component: Algebra of Matrices.
Application Unit 1: Basic operations with matrices in Matlab/Python. Matrix operations and image manipulation Matrix multiplication, inversion, and photo filters Solving linear systems.
14 Hours
Unit 2: Vector Spaces & Linear Transformations
Vector Spaces and Subspaces (definitions only), Linear Independence, Basis and Dimensions, Row reduced Echelon form, The Four Fundamental Subspaces, Rank-Nullity theorem. Linear Transformations, Algebra of Linear transformations, Self-Learning Component: Examples of Vector Spaces and Subspaces.
Application Unit 2: Systems of linear equations and college football team ranking (with an example of the Big 12) Convolution, inner product, and image processing revisited Norms, angles, and your movie choices Interpolation, extrapolation, and climate change.

14 Hours
Unit 3: Orthogonalization, Eigenvalues and Eigen vectors
Transform Orthogonal Vectors and Subspaces, Orthogonal Bases, Cosines and Projections onto Lines, Projections and Least Squares. Orthogonalization, The Gram-Schmidt Orthogonalization process, Introduction to Eigenvalues and Eigenvectors, Properties of Eigenvalues and Eigenvectors, Cayley-Hamilton theorem (statement only), Diagonalization of a Matrix.
Application Unit 3: Orthogonal matrices and 3D graphics Discrete dynamical systems, linear transformations of the plane, and the Chaos Game Projections, eigenvectors Matrix eigenvalues and the Google's PageRank algorithm.
14 Hours</td></tr>
</table>

	Unit 4: Singular Value Decomposition Quadratic Forms, Definitions of Positive definite, negative definite, positive semi-definite, negative semi-definite, indefinite forms and matrices, Tests for Positive Definiteness, Cholesky's method, Singular Values and Singular Vectros, Image Processing by Linear Algebra, Principal Component Analysis (PCA by the SVD and Covariance matrix), Minimum Principles. Application Unit 4: Principal Component Analysis, and face recognition algorithms Social networks, clustering, and eigenvalue problems Singular Value Decomposition and image compression. **14 Hours** **Hands on learning using PYTHON and Applications to Computer Science Engineers.**
TextBook(s):	1. Linear Algebra and its Applications, Gilbert Strang, Thomson Brooks/ Cole, 5th Edition, Second Indian Reprint 2007. 2. Application : Applied projects for an introductory linear algebra class by Anna Zemlyanova.
Reference Book(s):	1. Linear Algebra and its Applications, David. C lay, Publication by Pearson Education, 5th Edition, 2015 2. Linear Algebra, Schaum's outlines, Seymour Lipschutz and Marc Lipson, Tata McGraw-Hill publications, 4th Edition, 2009. 3. Higher Engineering Mathematics, B S Grewal, Khanna Publishers, 44th Edition, 2020,. 4. Practical Linear Algebra, Gerald Farin and Dianne Hansford, CRC Press, Taylor & Francis Group, 3rd Edition.
Course Outcome	● Solve systems of Linear Equations using Matrix Transformations, Interpret the nature of Solutions, Visualize Consistency of Linear system of Equations and also compute inverse of a Matrix. ● Demonstrate the ability to work within Vector Spaces, distil Vector Space properties and understand the concepts of the four fundamental Subspaces, Linear Span, Linear Independence, Dimension and Basis. ● Learn the concepts of Orthogonal Vectors and Orthogonal Subspaces and apply the Gram-Schmidt process to find an Orthonormal Basis in a Subspace, Eigenvalues, Eigenvectors and Diagonalization of a Matrix. ● Apply the concept of Positive Definite Matrices, Singular Value Decomposition into application problems.

<table>
<tr><td>Course Code</td><td>UE23CS351A</td><td>Course Title</td><td colspan="5">Database Management System</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ CreditAssigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>2</td><td>5</td><td>5</td></tr>
<tr><td>Semester</td><td>5</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>MySQL Workbench, Python, ERwin, Any other tool for ER modeling, RelaX - relational algebra calculator.Bytebase SQL Editor, Chat2DB</td><td>Desirable Knowledge</td><td colspan="5">-</td></tr>
<tr><td>Prelude</td><td colspan="6">This course offers a solid theoretical foundation in Database Management System (DBMS) and explores the practical applications of DBMS in a real-world scenario. The course emphasizes on the creation and the design of relational database systems. It also introduces the databases that provides flexible schemas and scale easily with large amounts of data and high user loads.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">• Understand fundamental concepts, terminology, and application of relational databases and Construct ER diagrams for a desired application and transform the same into a relational schema.
• Understand database design concepts and design relational databases and construct basic and advanced SQL queries.
• Understand, and apply Normal Forms in database design, Transactions, Concurrency control, Locking, and Recovery, implement them in a real-time setup, and transform the given Normal Form into the desired form.
• Understand the concepts of Database Security, and NoSQL databases such as MongoDB, Neo4j, and DynamoDB.</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Introduction to Database Management
Introduction to databases, Database application architecture, Users and Administrators, E-R Model, reducing ER to a relational schema. Structure of relational databases,Database schema, constraints,Keys, Enhanced ER Model Subclasses, Super classes, Inheritance, Specialization and Generalization and its Constraints and Characteristics. Relational Model: Relational operations (Algebra), Unary Operations - Unity, Binary, Aggregate Functions, Grouping. 14 Hours
Unit 2: Database Design: Basic and Advanced SQL
SQL overview, Data definition, Structure of SQL queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Database Modification, Join expressions, Views, Triggers, Functions, and Procedures, Event scheduler - Automated tasks, Replication – Master-slave, Multi-source, CTEs, Window functions, Full-Text Search Functions, Introduction to strategies of Query processing and Query optimization. 14 Hours
Unit 3: Advanced Design Concepts and Implementation
Functional Dependencies, Inference Rules, Closure, Equivalence, Minimal Cover Normal Forms Based on Primary Keys (1NF,2NF,and3NF), General Definitions of Second and Third Normal Forms Boyce-Codd Normal Form, Properties of Relational Decompositions, Overview of Higher Normal Forms. 14 Hours
Unit 4: Advanced Databases
Database transactions, Concurrency control, Locking, Recovery, Database Security, Security Threats (SQL Injection), Data protection at rest and in transit, Introduction to NoSQL databases, CAP theorem, Document database (MongoDB),Key-Value database (DynamoDB), Graph databases (Neo4j), Introduction to Multi-model Databases. 14 Hours</td></tr>
</table>

Laboratory	1. Draw an ER diagram for a given problem statement, Conversion of an ER diagram into a Relational schema. 2. Execution of Relational algebraic expressions using Relaxtool. 3. DDL–create constraints, alter, rename, drop, truncate table, Views. 4. DML–Insert,Update,Delete,Transactions-commit,rollback,savepoint. 5. SQL-Setoperators:union,intersect,minus.SQL–Aggregate functions. 6. SQL–Joins:inner,outer;Subqueries:correlated and uncorrelated. 7. SQL–Full-Text Search Functions. 8. SQL-Event Scheduler-Automated tasks, Replication–master-slave, multi-source, CTEs, Window functions. 9. SQL–Creating Functions and Procedures. 10. SQL–Creating Triggers and Cursors. 11. XML-Database access. 12. NoSQL database queries. 13. High-level programming language accessing a database using an API.
TextBook(s):	1. "Fundamentals of Database Systems",Ramez Elamsri, Shamkant B Navathe,Pearson,7th Edition, 2017. 2. "Database System Concepts",Silberschatz, HKorth and S Sudarshan, McGrawHill, 7th Edition, 2019.
Reference Book(s):	1. Database Management Systems, R Ramakrishnan, JGehrke, 3rd Edition,McGrawHill,2002. 2. Data on the Web: From Relations to Semistructured Data and XML, S Abiteboul, P Buneman, D Suciu, Morgan Kauffman, 1999.
Course Outcome	At the end of this course, the student will be able to: ● Demonstrate an ability to explain the basic concepts of database management, construct and transform ER diagrams into Relational Schema. ● Design databases and construct simple and advanced SQL queries for given contexts. ● Explain Normal Forms, employ them in Database Design,and apply database security concepts in application contexts. ● Demonstrate the ability to use semi-structured and NoSQL databases.

<table>
<tr><td>Course Code</td><td>UE23CS352A</td><td>Course Title</td><td colspan="5">Machine Learning</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>2</td><td>5</td><td>5</td></tr>
<tr><td>Semester</td><td>5</td><td>Type of Course</td><td colspan="5">Core</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Pytorch, SkLearn, Keras, Tensor flow Google cloud VertexAI(end-to-end ML platform), Amazon Sagemaker, IBM Watson studio, Auto ML tools (H2O.ai (H2O AutoML), Google AutoML, Hugging Face, Haystack etc</td><td>Desirable Knowledge</td><td colspan="5">Mathematics for Computer Science Engineers Linear Algebra & its Applications, Design and Analysis of Algorithms.</td></tr>
<tr><td>Prelude</td><td colspan="6">Machine Learning surrounds us today: in phones that respond to voice commands, programs that beat humans at Chess and Go, robots that assist surgeries, vehicles that drive in urban traffic, and systems that recommend products to customers on e-commerce platforms. This course aims to familiarize students with the breadth of modern ML, to impart an understanding of the dramatic surge of ML in the last decade, and to foster an appreciation for the distinctive role that ML can play in shaping the future of our society. This course requires the student to have a desirable knowledge of Statistics for Data Science, Linear Algebra and its Applications and Design and Analysis of Algorithms.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● Formulate a well-defined Machine Learning problem with clear metrics.
● Understand the notions of Hypotheses Space, Hypotheses Structure and Search.
● Become conversant with types of Machine Learning Algorithms, their applicability, and Inductive Bias.
● Familiarize with techniques for Ensemble Learning and deep learning.</td></tr>
<tr><td>Course Contents:</td><td colspan="6">Unit 1: Foundations of Machine Learning and Supervised learning Models
Introduction to Machine Learning- Machine Learning Models-Supervised, Unsupervised, Semi-supervised, Reinforcement Learning. Concept Learning, Hypotheses and Hypothesis Space, Version space, Inductive bias. Decision Tree Learning- Decision Trees, ID3 and dealing with continuous values for classification in DTs, Decision boundary for decision trees(x-y axis), Bias Variance decomposition. Issues in Decision Tree Learning – Overfitting, Solutions to overfitting. Other Core Supervised Learning Approaches- Instance-based learning: k nearest neighbor learning (Classification & Regression), Decision boundary for KNN.
14 Hours

Unit 2: Neural Networks, Ensemble Learning,and Kernel-Based Methods for Classification
Artificial Neural Networks- Introduction, Perceptrons, Multi-layer Networks and Backpropagation, Activation Functions (Step, Sigmoid, Tanh, ReLU), Various Optimizers (Gradient Descent, Stochastic Gradient Descent, Momentum-based, Adagrad, Adam), Ensemble Models- Combining Weak Learners, Improving Performance with Gradient Boost. Support Vector Machines – Margin and Maximization, SVM – The Primal Problem, The Lagrangian Dual, SVM – Solution to the Lagrangian Dual (Hard Margin).

14 Hours</td></tr>
</table>

	Unit 3: Stochastic Models and Unsupervised learning Bayesian Learning- Bayes' Theorem, Maximum Likelihood Estimation (MLE), Bayes Optimal Classifier, Unsupervised Learning-Hierarchical vs. Non-Hierarchical Clustering, Agglomerative and Divisive Clustering, Bisecting K-Means, and Expectation Maximization (EM), K-Means as a Special Case of EM, Gaussian Mixture Models (GMMs), Hidden Markov Models (only Forward Algorithm derivation. Backward, Viterbi and Baum Welsch Algorithms applications only). **14 Hours** **Unit 4: Introduction to Deep Learning Architectures from CNNs to Transformers** Convolution Neural Networks-Introduction to Convolution Operation, Convolutional Neural Network (CNN) Architecture, Parameter Calculation in CNNs, Max Pooling, Average Pooling, Recurrent Neural Networks Introduction to Recurrent Neural Networks (RNN), Vanishing and Exploding Gradient Problems, Variants of RNN – LSTM and GRU (mention of gates), Advanced Deep Learning Architectures- Basic GAN and Its Architecture, Basic Architecture of Transformers (Encoder-Decoder Perspective with Basics of Attention Mechanism). **14 Hours**
Laboratory	1. Introduction to SciKit, Performance Metrics 2. Decision Trees. 3. ANN 4. Random Forest. - Concept & Implementation 5. SVM (Kernel Functions/soft margin SVM-Concept & implementation) 6. Naïve Bayes (concept and Implementation) 7. HMM 8. K means Clustering (Concept and Implementation) 9. CNN model.
TextBook(s):	1. "Introduction to Machine Learning", Ethem Alpaydin, Third edition, MIT Press Ltd,2015 2. "Machine Learning: The Art and Science of Algorithms that Make Sense of Data", Peter Flach, Cambridge University Press (2012). 3. Deep Learning by Ian Goodfellow,Yoshua Bengio,and Aaron Courville (MIT Press,1st Ed., 2016)
Reference Book(s):	1. "Machine Learning", Tom Mitchell, McGraw Hill Education(India),2013. 2. Pattern Recognition and Machine Learning by Christopher M.Bishop (Springer,1stEd.,2006). 3. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy (MIT Press, 1st Ed., 2012).
Course Outcome	• Distinguish categories of Data Attributes, Dimensions, and Sample Sizes. • Acquire a thorough understanding of Supervised, Unsupervised Learning, • Apply Ensemble Methods and deep learning methods for solving the problems. • Design and implement a solution to real world machine learning problem.

Course Code	UE23CS341A	Course Title	Software Engineering				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Core				
AI Tools /Tools/Languages	Git, GitHub, docker, Jenkins, gtest, pytest, Jira/OpenProject, SonarQube.	Desirable Knowledge	-				
Prelude	In this course, students learn and gain practical experience with software engineering principles, techniques, and tools. Topics in this course include requirements analysis, specification, architecture, design, programming style, testing, quality management, security validation, devOps, teamwork, and software project management. The practical experience centers on a semester-long team project, in which a software development project is carried through all the stages of the software life cycle. Particular emphasis is placed on performing all SDLC activities similar to industry setup. Assignments further provide a hands-on experience in using state-of-the-art techniques and tools.						
Course Objectives:	• Learn Software Engineering concepts. Ensure the relevance and need of an engineering approach to software development. Expose students to the software project management and Lifecycles. • Expose students to agile development methodology using standard tools and associated specifications. • Enable the students to understand the principles of Software Configuration Management and importance of software quality for Software Product Development. • Enable students to understand the continuous development, build, test and release of software products through a semester long project.						

Course Contents	**Unit 1: Introduction to Software Engineering and Requirements Engineering** Introduction, Case studies and motivation for SE, DevOps and CI/CD tools introduction and setup, Software development lifecycles overview (waterfall etc) and motivation for agile methodology. Agile approach of software development, Contrasting Agile and Plan driven approaches, Agile SCRUM model and exposure to other Agile approaches like Lean Agile, XP etc. Hands on Jira training. Security development lifecycle. Requirements Engineering: Requirement's introduction and properties, Feasibility, Requirements Elicitation, Analysis and modelling, Specification and Verification, Requirement Management and Requirements Traceability; Introduction to UML and development of use cases. Validation planning and security validation plan. **16 Hours** **Unit 2: Software Project Management and Software Architecture and Design and Quality** Fundamentals, Software Project Management Lifecycle, Planning activities, Choice of Lifecycles, Project Organization, Software Estimation, Scheduling, Risk Management, Quality Management, Gantt charts basics, Monitoring of Execution and Control, Project Closure. RACI matrix with example Software Architecture: Case study for motivation of SW Arch, Architectural View, Styles, Architectural and Design Patterns. Modularity, interface definition, microservices intro, Security architecture, Agile context. Software Design: Modularity, importance of interfaces (API and ABI), Error definition, handling and management, Forward and backward compatibility, usage of UML. Quality and Test case development: Case studies and motivation, types of testing, technical debt, Test cases documentation, Black and white box testing. **12 Hours**

	Unit 3: AI, Implementation, SCM, DevOps AI for SW Engineering and SE for AI/ML systems. CI/CD pipeline and tools for development, Introduction to software construction, Coding principles, standards and guidelines, Factors for effective coding - Defensive, Secure and Testable programming, Code Review / Peer Review, Security code reviews; logging, tracing, debugging techniques. Test case development and dynamic analysis, functional and nonfunctional testing, security testing, Automation; mutation testing to validate test suite Software Configuration Management (SCM): Elements of a Configuration Management System, Configuration Items, Baselines, Repositories, Branch Management, Build Management, Install, Change and Release Management, Patching and Patch Management. **16 Hours** **Unit 4: Usability, Quality and System Testing** ICase study discussion, Industry talk, Usability, UX design, Code coverage, branch coverage; Ethics and Software Engineering; Opensource Introduction; SW maintenance and support; Security and Privacy aspects; Measurements and Matrix for Quality; Risk management for Quality Security Validation – Penetration Testing, Fuzzing; System testing – Regression testing, Integration testing Demo and student presentations for the mini-project. **12 Hours**
TextBook(s):	1. "Software Engineering", International Computer Science Series, Ian Somerville, Pearson Education, 10th Edition, 2021. 2. "Software Engineering: A Practitioner's Approach", Roger S Pressman, McGraw Hill, 9th Edition, 2019.
Reference Book(s):	1. "Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling" by Jennifer Davis, Ryn Daniels, O' Reilly Publications, 2018. 2. "Software Engineering: Principles and Practice", Hans van Vliet, Wiley India, 3rd Edition, 2010. 3. "Foundations of Software Testing ", Aditya Mathur, Pearson, 2008. 4. "Software Testing, A Craftsman's Approach ", Paul C. Jorgensen, Auerbach, 2008. 6:IEEE SWEBOK, PMBOK, BABOK and Other Sources from Internet. 5. "Software Testing – Principles and Practices", Srinivasan Desikan and Gopalaswamy Ramesh, Pearson, 2006.
Course Outcome	● Relate to the challenges of Software Development and Software Engineering as a methodical approach for development. Use state-of-the-art tools and techniques for large-scale software systems development. ● Use Software project management ideas and apply suitable architecture and design approaches for software development. ● Implement the major software development methods in practical projects including CI/CD, various devOps tools etc. ● Follow industry accepted documentation and review processes to develop a high quality SW project.

<table>
<tr><td>Course Code</td><td>UE23AM342AA1</td><td>Course Title</td><td colspan="5">Advanced Foundation of Machine Learning</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>5</td><td>Type of Course</td><td colspan="5">Elective - I</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Pytorch, SkLearn, Keras, Tensorflow Google cloud Vertex AI (end to-end ML platform).</td><td>Desirable Knowledge</td><td colspan="5">Mathematics for Computer Science Engineers,
Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.</td></tr>
<tr><td>Prelude</td><td colspan="7">This course is a companion elective for the core Machine Learning course. This course equips the students, aspiring for a future career in Machine Learning, with additional mathematical foundation and topics that the core Machine Learning course may not cover. The course primarily focuses on topics related to Deep Learning, preparing the students for advanced machine learning elective courses in future semesters.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Learn vector and matrix calculus to understand the mechanics of deep learning using concepts such as Gradient, Jacobian, Hessian, and convex functions.
● Learn mathematical foundations of Optimization and Regularization techniques in machine learning, with a specific focus on Deep Learning
● Learn the mathematical treatment of Linear Regression and Logistic Regression, Generative Deep Learning, and additional dimensionality reduction methods.
● Learn Imbalanced Learning and Semi-supervised learning.</td></tr>
<tr><td>Course Contents:</td><td colspan="7">Unit 1: Vector and Matrix Differentiation for Deep Learning
Introduction - Vector, Matrix, Tensor in Machine Learning. Geometric intuition for machine learning- model error, feature similarity, independence of vectors, and multidimensional hyperplane. Gradient – vector of partial derivatives of a scalar-valued function of multiple variables. Jacobian - gradient of a vector-valued function. Chain rule on a multivariate function. Hessian Matrix and Discriminant. Application of Jacobian in Machine Learning – Gradient Descent and Back propagation algorithm. Multi-dimensional Taylor series and Hessian– why the gradient provides the best direction for descent. Convexity of functions –global and local minima. Application of Hessian in Machine Learning- convexity in multi-dimensional function. A few useful matrix differentiation identities in machine learning.

14 Hours

Unit 2: Optimization and Regularization
Categories of optimization algorithms – differentiable and non-differentiable target function, Convex and non-convex function, Local vs. Global minima. Optimization of model hyperparameters: Random and Grid Search, Stochastic Hill climbing, and Bayesian optimization.
Local Optimization over differentiable target functions - Stochastic, Batch, and Minibatch Gradient Descent. Convergence - Momentum, Nesterov Accelerated Gradient (NAG), AdaGrad, RMSProp, Adam. Normalization – Data, Batch, and Layer normalization. Loss functions and differentiability of loss functions– softmax, cross-entropy, and binary cross-entropy. Learning to learn (meta learning), i.e., Optimization-based approach – Model Agnostic Meta Learning. Metric-based approach – Prototypical Network. Global Optimization over non-differentiable target function – Genetic Algorithm. Regularization - Occam's Razor and Minimum Description Length, No Free Lunch Theorem. Regularization via objective function–Norm-based penalty, Lasso penalty, Ridge penalty, Elastic net. Regularization in neural networks– Weight decay, Learning curves, Parameter sharing, Residual connection, Model averaging or dropout.

14 Hours</td></tr>
</table>

	Unit 3: Regression, Generative Deep Learning, and Dimensionality Reduction Regression: Linear Regression - Linear Algebra, MLE, and Gradient Descent interpretation. Logistic Regression- Stochastic Gradient Descent& MLE interpretation. Generative Adversarial Network – Loss Function, GAN training, CycleGAN. AutoEncoder – deep autoencoder, sparse autoencoder, denoising autoencoder, variational autoencoder. Dimensionality Reduction: Linear methods – Linear Discriminant Analysis(LDA), Independent Component Analysis (ICA).Non-linear methods: Kernel PCA, Multi-dimensional Scaling (MDS), t-SNE, Isometric Mapping (ISOMAP). Neural Methods – Self-Organizing Map (SOM). **14 Hours** **Unit 4: Unsupervised Learning, Learning with Scarcely Labeled Data** Unsupervised Learning – Density based clustering (DBSCAN), Spectral Clustering Imbalanced Learning: Data Sampling – over-sampling and under-sampling, SMOTE. Cost-sensitive learning. One-class classifiers. Data Augmentation and synthetic data generation. Ensemble learning for Imbalanced Data. Semi-supervised Learning: SSL assumptions. Self-training and Co-training algorithms. Deep Semi-supervised Learning with Ladder Network. Graph-based classical semi-supervised learning algorithms– Label Propagation and Label Spreading Algorithm. **14 Hours**
Text Book(s):	1. Deep Learning Foundations and Concepts, Christopher M. Bishop with Hugh Bishop, Springer, 2023. 2. MATHEMATICS for MACHINE LEARNING, MarcPeter Deisenroth, A.Aldo Faisal,Cheng SoonOng, Cambridge University Press(2020).
Reference Book(s):	1. Dive into Deep Learning, Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Cambridge Univ Press, 2023. 2. Deep Learning, Ian Good fellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016. 3. Matrix Differentiation, Randal JBarness, Springs Journal(page1-6),2006,UniversityofMinnesota. 4. The matrix Calculus you need for need for Deep Learning , arXiv, Terence Parr, Jeremy Howard.
Course Outcome	● Deal with matrix calculus concepts necessary for advanced deep learning courses . ● Deal with optimization and regularization techniques in ML. ● Deal with Generative Deep Learning and additional dimensional reduction methods. ● Deal with scarcely labeled datasets using suitable techniques.

<table>
<tr><td>Course Code</td><td>UE23AM342AA2</td><td>Course Title</td><td colspan="5">Active Learning</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>5</td><td>Type of Course</td><td colspan="5">Elective - I</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Pytorch, SkLearn, Keras, Tensorflow Google cloud Vertex AI (end to-end ML platform).</td><td>Desirable Knowledge</td><td colspan="5">Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.</td></tr>
<tr><td>Prelude</td><td colspan="7">Creating good datasets requires humans in the loop and is often neglected, leading to suboptimal results. Dataset creation is often an important activity in research and industry, where an application-specific model is developed. Data scientists need better training and evaluation data for their machine learning models, as a good model with poor data yields poor results. Active Learning comes to the rescue when we have a very small amount of labeled data for a learning task, and this situation is often encountered. This course does a deep dive into active learning methods to deal with how to create better datasets with humans in the loop.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Should develop a deep understanding of the importance of well-annotated data in machine learning and active learning.
● Should be able to use different sampling algorithms in Active Learning.
● Should be able to use Active Learning methods for different machine learning tasks.
● Should be able to create an annotated dataset by applying active learning methods.</td></tr>
<tr><td>Course Contents:</td><td colspan="7">Unit 1: Introduction to Active Learning and Uncertainty Sampling
Introduction: Importance and challenges of quality human annotation. Introduction to Active Learning – when to use. The three main active learning strategies - uncertainty, diversity, and random sampling. Human-computer interface in creating training data. Uncertainty Sampling: Interpreting uncertainty in a machine learning model, Uncertainty sampling algorithms, identifying when models are confused, Measuring uncertainty across predictions, Selecting the right number of samples.
14 Hours

Unit 2: Diversity Sampling and Advanced Active Learning
Diversity sampling – model-based outlier sampling, cluster-based sampling, representative sampling, sampling for real-world diversity. Advanced Active Learning - combining uncertainty sampling and diversity sampling. Active Transfer Learning- applying active transfer learning to uncertainty sampling, representative sampling, and adaptive sampling.
14 Hours

Unit 3: Applying Active Learning to Machine Learning Tasks
Applying active learning to machine learning tasks-Applying Uncertainty sampling, Diversity sampling, Active Transfer Learning, and Representative sampling to Object detection, Semantic segmentation, Sequence labeling, and Language generation tasks.
14 Hours

Unit 4: Data Annotation
Working with people – in-house experts, crowd sourced, and outsourced. Estimating annotation volume required. Quality control for annotation – comparing annotation with ground truth, inter-annotator agreement, and aggregating multiple annotations. Machine Learning for annotation quality control. Annotation quality for different machine learning tasks.
14 Hours</td></tr>
</table>

Text Book(s):	1. Human in the loop machine learning", Robert Monarch, Manning Books, 2021. 2. Active Machine Learning with Python: refine and elevate data quality over quantity with active learning" , Margaux Masson-Forsythe (Author), Packt, 2024.
Course Outcome	● Should develop a deep understanding of the importance of well-annotated data in machine learning and active learning. ● Should be able to use different sampling algorithms in Active Learning. ● Should be able to use Active Learning methods for different machine learning tasks. ● Should be able to create an annotated dataset by applying active learning methods.

Course Code	UE23AM342AA3	Course Title	Social Computing				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Python tools and libraries for graph analytics, such as NetworkX, along with Python text processing and Machine Learning libraries, such as NLTK, Spacy, PyTorch, etc.	Desirable Knowledge	Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.				
Prelude	As society has largely moved online, people's footprints on social media have become the windows to people's minds. This interdisciplinary course explores how computational models and techniques help understand social interactions. The first part of the course focuses on text mining for social media data and social graphs that often help explain how a social event builds up. The second part of the course dives deep into contemporary social computing, i.e., social NLP-based investigations dealing with Language, Users, and Social media events of interest today.						
Course Objectives:	• Learn text mining techniques. • Learn fundamental principles of social graph analytics. • Learn social NLP use cases based on language and user information in social media datasets. • Learn how to analyze events of interest in today's online societies.						
Course Contents:	**Unit 1: Text Mining - Linguistic Pre-processing and Analysis of Text** Linguistic Pre-processing of Text – Text Tokenization, Text normalization, Understanding text syntax and structure -Part Of Speech Tagger, Chunker, and Parser. Text classification – Text Normalization Feature Extraction with Bag Of Words (TF IDF), Using dense embedding models -Word2Vec, BERT, etc. Evaluating the classification model. Text summarization and Information Extraction - key phrase extraction, topic modeling – Latent Semantic Indexing, Latent Dirichlet Allocation, Automatic document summarization - TextRank. Text Similarity and Clustering- Term and Document Similarity, K-means Clustering, and Affinity Propagation. Semantic and Sentiment Analysis of Text- Named entity recognition, opinion, and sentiment mining. Suggested Case Study – Mining Product Reviews collected from Social Media. **14 Hours** **Unit 2: Social Graph from Social Media Data** Social Graphs – Degree and degree distribution, Paths, strong and weak ties, signed networks. Centrality – degree centrality, closeness centrality, betweenness centrality, Eigenvector, Katz, and PageRank centrality. Components in the graph: clique, N-clique, N-clan, N-club, N-core, connected components. Clustering- Global and Local clustering coefficient. Community Detection - Clique Percolation method, modularity optimization methods. Properties of the real-world network. Generative models- Random Graph Models, Preferential Attachment model, Watts and Strogatz Small world model. Suggested case study- Building a social graph from a corpus of tweets and analysis of the same. **14 Hours** **Unit 3: Pro-social behavior, Anti-social behavior, and Social Media Events** Language Civility - Politeness, Hate Speech, Toxicity, and Argumentation. User - Credibility, User modeling, Social Roles. Anti-social computing-Trolling, Flaming, and Dark content. Polarization and Mental Health. Social Media Events -Predicting Voting Intentions, Crisis Informatics and Disaster Response, Rumor, Fake News and Propaganda. **14 Hours** **Unit 4: Large Language Models in Social NLP, Challenges - Bias, Ethics, and Privacy** Prompt Engineering Patterns for Text- Elaborator, Corrector, Expander, Paraphraser, Classifier, Extractor, Fact Checker, Summarizer, Generator, Completer, Balancer, Disambiguator, Reformatter, Context Binder. Large Language Model Social Computing – Generation of a Dataset and its effectiveness. Learning models using LLM using OPENAI –Classification, Information Extraction, and Clustering. Challenges of social computing – Ethics and Privacy. Bias and Fairness Issues in Social Computing. **14 Hours**						

Text Book(s):	1. Deep Learning Foundations and Concepts, Christopher M. Bishop with Hugh Bishop, Springer, 2023. 2. MATHEMATICS for MACHINE LEARNING, Marc Peter Deisenroth, A.Aldo Faisal, ChengSoon Ong, Cambridge University Press (2020).
Reference Book(s):	1. Dive into Deep Learning, Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Cambridge Univ Press, 2023. 2. Deep Learning, Ian Good fellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016. 3. Matrix Differentiation, Randal JBarness, Springs Journal (page1-6), 2006, University of Minnesota.
Course Outcome	● Deal with matrix calculus concepts necessary for advanced deep learning courses . ● Deal with optimization and regularization techniques in ML. ● Deal with Generative Deep Learning and additional dimensional reduction methods. ● Deal with scarcely labeled datasets using suitable techniques.

Course Code	UE23AM342AA4	Course Title	AI for Smart Manufacturing				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Oracle ERP, Salesforce CRM, SCM applications and tools, ML tools.	Desirable Knowledge	-				
Prelude	This course explores the integration of AI in smart manufacturing systems, focusing on both Enterprise Business Systems (EBS) and Supply Chain Management (SCM). Students will gain a comprehensive understanding of the role of AI in streamlining manufacturing processes, optimizing supply chains, and enhancing enterprise resource planning. The course includes theoretical insights, practical applications, and case studies and mini project to understand the functionalities, implementation methodologies, challenges, and strategic implications of AI technologies in modern manufacturing enterprises.						
Course Objectives:	• Understand the fundamentals of Enterprise Business Systems (EBS) and Supply Chain Management (SCM). • Learn the strategic importance of ERP and SCM systems in the industry. • Explore AI techniques applied to solve real-world SCM challenges. • Identify critical success factors and associated risks of ERP and SCM implementations.						
Course Contents:	**Unit 1: Introduction to Enterprise Resource Planning Systems** Introduction to ERP: Introduction, Value Chain Framework, Problems with Disintegrated Data in an Organization, Evolution of ERP Systems, Role of ERP Systems in an Organization, Three-tier Architecture for ERP Systems, Stages Theory and Its Application to Evolution of ERP, Scope of ERP systems, General Model of Business and Role of ERP, Major ERP Players, Implementations in India. Lifecycle of an ERP Implementation Project, Teams. Benefits and Cost of an ERP System: Cost-Benefit Analysis, ERP Trends like Analytics, AI, Cloud, and Technology Integration. **Case Studies:** AI in ERP Implementation at Siemens: Explore how Siemens used AI to streamline their ERP implementation, enhancing data integration and decision-making processes. Predictive Analytics in ERP at Schneider Electric: Analyze Schneider Electric's use of predictive analytics within their ERP system to forecast demand and optimize production schedules. **14 Hours** **Unit 2: Change Management, Re-Engineering, and BPM** Change Management: Introduction, Change management for ERP project team, team and roles, change management activities during lifecycle of a project, six key processes of ASAP. Re-Engineering: Processes and Characteristics, Lifecycle of BPR and IT-driven BPR Projects. Business Process Modeling (BPM): Introduction, Hierarchy, Standards, and Integrated Data Modelling. **Case Studies:** AI in Change Management at General Electric: Assess how General Electric utilized AI tools to manage change during their ERP re-engineering project, focusing on employee adaptation and process improvements. Business Process Re-Engineering with AI at Procter & Gamble: Evaluate Procter & Gamble's use of AI to re-engineer business processes, resulting in increased efficiency and reduced operational costs. **14 Hours** **Unit 3: Procurement and Supply Chain Planning** Procurement and Inventory Management through ERP, Understanding the Supply chain. Demand forecasting in a supply chain. Aggregate planning in a supply chains. Managing predictable variability. **Case Studies:** AI-Driven Supplier Selection at IBM: Study IBM's implementation of AI for optimizing supplier selection and performance evaluation, improving procurement efficiency. Demand Forecasting using ML at Amazon: Examine Amazon's use of machine learning models to predict customer demand, reducing inventory costs and improving supply chain responsiveness. **14 Hours** **Unit 4: Planning and Managing Inventory and Logistics Management in SCM** Production planning and execution – understanding MRP II concepts, how ER production planning module supports MPP II processes, critical master data elements for production planning, managing different production scenarios. Supply chain planning and its modules, collaborative planning solutions. Logistics						

	Execution: warehouse and transport management – logistics execution and transport management. **Case Studies:** AI-Enhanced Warehouse Management at Walmart: Investigate Walmart's application of AI for optimizing warehouse operations, leading to improved inventory accuracy and reduced picking times. AI-Driven Logistics Optimization at DHL: Explore DHL's use of AI for route planning and last-mile delivery, enhancing logistics efficiency and customer satisfaction. **14 Hours**
Text Book(s):	1. "Enterprise Resource Planning: A Managerial Perspective" by Veena Bansal, Pearson Education India, 2013. 2. "Enterprise Resource Planning- Text& Cases" by Rajesh Ray, Tata McGraw Hill, 2011. 3. "Supply Chain Management: Strategy, Planning, and Operation" by Sunil Chopra, Pearson, 7th Edition, 2024.
Reference Book(s):	1. "ERP Demystified" by Alexis Leon, McGraw Hill Education, 4thEd,2019. 2. "Modern ERP: Select, Implement, and Use Today's Advanced Business Systems" by Marianne Bradford, Pearson, 4th Edition, 2020. 3. "Enhancing Enterprise Intelligence: Leveraging ERP, CRM, SCM, PLM, BPM, and BI" by Vivek Kale, Auerbach Publications, 2016. 4. "Essentials of SCM" by Michael H. Hugos, John Wiley & Sons, 3rd Edition,2018. 5. "Principles of Supply Chain Management: A Balanced Approach" by JoelD. Wisner, Cengage, 5th Ed, 2019. 6. "Strategic Supply Chain Management: The Five Core Disciplines for Top Performances" by Shoshanah Cohen & Joseph Roussel, Mc-Graw Hill, 2nd Ed, 2013. 7. "Logistics and Supply Chain Management" by Martin Christopher, Pearson, 2022. 8. "Supply Chain Logistics Management" by Donald Bowersox, etal., Mc-Graw Hill Publishing, 5th Edition, 2024. 9. "Supply Chain Management: A Logistic Approach" by John J. Coyle, Cengage, 10th Ed, 2019.
Course Outcome	● Identify typical functionalities of ERP and SCM sub-systems. ● Apply AI and ML techniques to optimize inventory, improve logistics, and predict trends. ● Evaluate and select strategies for planning and managing inventories and logistics. ● Develop systematic plans for ERP and SCM implementation projects.

Course Code	UE23AM342AA5	Course Title	AI in Medical Imaging				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Python, Tensorflow/PyTorch, ImageJ, MATLAB.	Desirable Knowledge	UE23MA241A- Mathematics for Computer Science Engineers, UE23MA241B - Linear Algebra and its Applications.				

Prelude	This course explores the application of Artificial Intelligence (AI) in the analysis and interpretation of medical imaging data. It covers the fundamentals of medical imaging techniques, the principles of AI and machine learning (ML), and the development of AI models for enhancing diagnostic accuracy, automating image analysis, and supporting clinical decision-making. The curriculum aims to equip students with the skills to develop AI solutions that address challenges in medical imaging.
Course Objectives:	● Understand the basics of medical imaging modalities (e.g., X-ray, CT, ultrasound, MRI, PET). ● Explore techniques to pre-process the medical images. ● Learn about AI and ML techniques applicable to medical imaging analysis. ● Explore the process of developing, training, and validating AI models for medical imaging.
Course Contents:	**Unit 1: Fundamentals of Medical Imaging** Basics of Digital Image Processing, Introduction to Medical Imaging, Basics of Pixel/Voxel, Metadata, Visualization Tools, Data Types, Medical Image Modalities - X-ray, CT, Ultrasound, SPECT, MRI, PET. **14 Hours** **Unit 2: Processing of Medical Images** Bias Field Correction, Histogram Normalization, Denoising, Masking, Image Interpolation, Handling 3D Images, Augmentation, Object Detection and Localization, Medical Image Segmentation, Basics of Image Registration. **14 Hours** **Unit 3: Medical Image Classification and Regression** Data Harmonization, Classification and Regression Architectures, Deep Representation Learning, Transformer-based Deep Learning Segmentation, Medical Diagnosis and Prognosis, Deep Learning Models on Medical Images - CNN. **14 Hours** **Unit 4: Deployment of AI models** VAE, GANs, Diffusion Models, Case Studies of AI Applications in Radiology, Pathology, and other domains, Integration of AI Tools with clinical workflows, Handling class imbalance in medical imaging datasets, Deploying of AI models in research to clinical practice. **14 Hours**
Text Book(s):	1. "Medical Image Analysis", by Alejandro Frangi, Jerry Prince, Milan Sonka, ISBN: 9780128136577, eBook ISBN: 9780128136584. 2. "Deep Learning for Medical Image Analysis" by S. Kevin Zhou, Hayit Greenspan, and Dinggang Shen. 3. "Biomedical Image Analysis" by Rangaraj M Rangayyan, 1st Edition, CRC Press. 4. "Medical Imaging Signals and Systems" by Jerry L Prince, Jonathan M Links, 2nd Edition, Pearson Education, 2014
Course Outcome	● Describe different medical imaging techniques and the types of data they produce. ● Apply AI and ML algorithms to analyze and interpret medical imaging data. ● Evaluate the performance of AI models in medical imaging applications.

Course Code	UE23CS342AA3	Course Title	Internet of Things				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Python, Embedded-C, Cloud Platforms, Single Board Computers	Desirable Knowledge	-				
Prelude	The Internet of Things is already changing the way people live and interact with humans and machines. Businesses in every vertical have already started to leverage the power of IoT platforms to increase their efficiency and performance. This course introduces the students to the three layer architecture of the IoT exploring all the major connectivity options and application layer protocols. It also encourages the students to develop interesting applications using different development boards, sensors and actuators. The course introduces the students to cloud platforms and analytics including security aspects of IoT Application development.						
Course Objectives:	• Learn the fundamentals of the Internet of Things. • Learn the concepts of smart objects & IoT architecture. • Compare different application and network layer protocols for Internet of Things. • Appreciate the role and importance of Data Analytics and Security in real word problems and know the role of IoT in various verticals.						
Course Contents	**UNIT I: Introduction to IoT & Architecture** **Introduction** IoT Traffic Model, IoT Connectivity, IoT Verticals, Use Cases & Applications, IoT Value Chain. Examples of IoT Use Cases &Applications, IoT Project Implementation, IoT Standards. **IoT Architecture:** Introduction, Factors Affecting an IoT Architectural Model, IoT Architectural Model, IoT WF Model, Data Center &Cloud, Computing (Cloud, Fog & Edge). **14 Hours** **UNIT 2: IoT Sensors** Introduction, Sensors & Its Performance Metrics, Sensor Selection, Smart Sensors, MEMS, Sensor Fusion, Self-Calibration, Sensors of the Future. Introduction to Embedded Programming, Designing IoT applications using the STM32 development board. **14 Hours** **UNIT 3 : IoT-Protocols: IoT Wired Connectivity** Introduction, Ethernet, Ethernet TSN. Unlicensed-band Wireless Connectivity: Introduction, Zigbee, BLE, WiFi, LoRaWAN. Cellular IoT Technologies: Introduction, GSM-IoT, LTE, Practical Use Cases. Application Protocols for IoT: The Transport Layer-IoT Application Transport Methods, SCADA, Generic Web-Based Protocols, IoT Application Layer Protocols-CoAP and MQTT. **14 Hours** **UNIT 4: IoT–Analytics, Security & Privacy** Introduction, Data Pipeline, AI, Machine Learning, Supervised Learning Technique: Classification: Decision Tree Example. IoT Cloud-Based Services & Platforms: Introduction to thinger.io and Thing Speak for IoT. IoT Security - Introduction, IoT Threats, IoT Vulnerabilities, IoT Threat Modeling & Risk, IoT Security Regulations, IoT Privacy Concerns & Regulations, IoT Security & Privacy Examples, IoT & Blockchain. **14 Hours**						

TextBook(s):	1. Dian, F. J, "Fundamentals of Internet of Things", Wiley Professional Development (P&T), 2022.David Hanes, Gonzalo Salgueiro, Patrick Grossetete,Robert Barton,Jerome Henry,"IoT. 2. Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition, Pearson Education (ISBN: 978-9386873743).
Reference Book(s):	1. "Building Enterprise IoT Applications",Chandrasekar V uppalapati, CRCPress,Taylor & Francis Group, 2020. 2. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015. 3. Embedded Systems And Internet Of Things by Dr.Sonu Kumar,Dr.M.Venkatanarayana, ISBN:978-93-5747-591-4. 4. "Embedded Software Development for the Internet Of Things: The Basics, the Technologies and Best Practices" by Klaus Elk.
Course Outcome	● Recognize the principles of smart objects and the potential for IoT and comprehend IoT architecture from sensors to the cloud, including edge gateways. ● Determine the static and dynamic performance metrics for core sensors and to understand the architecture of smart sensors for IoT system design. ● Learn how to use various wired and wireless technologies to provide connectivity for IoT applications and select a suitable application-layer protocol among the ones used for IoT applications. ● Recognize how analytics and artificial intelligence algorithms are used in the IoT ecosystem along with IoT security and privacy concerns.

Course Code	UE23CS342AA4	Course Title	Applied Cryptography				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	SEED labs, Python Programming Language	Desirable Knowledge	-				
Prelude	Cryptography is the science of securing data by using mathematical concepts. This course will present the fundamentals of cryptography, as well as its applications and issues of how cryptography is used in practice. Students will have opportunities to dwell well into problem solving and hands-on sessions.						
Course Objectives:	• To enable understanding foundations of Cryptography. • To discuss about various symmetric encryption techniques. • To understand the concept of public key cryptography. • To introduce message authentication and hash function. • Explore emerging cryptography and future Directions.						
Course Contents	**Unit 1: Classical Ciphers.** Introduction to Cryptography, Cryptanalysis and brute-force attack, Basic cryptographic primitives. Mathematical background for cryptography: Modulo arithmetic, GCD, Euclidean algorithm, Introduction toprobability,Conditionalprobability,Lawoftotalprobability.Classicalciphers:Substitution-Caesar, Playfair and Hillcipher, Transposition cipher -Railfence, Columnar and Double columnar, Cryptanalysis of classical ciphers. Shannon's theorem and Perfect secrecy, One-time pad encryption and its limitations. **14hours** **Unit 2: Symmetric Key Cryptography** Introduction to symmetric key cryptography and Pseudo Random Number Generator (PRNG), The Feistel cipher, Data Encryption Standard (DES), Analysis, Multiple DES, Security and the avalanche effect, Mathematics- I: Galois fields, Polynomials- Advanced Encryption Standard (AES), Transformation functions, Key expansion, Analysis of AES, Block and Stream ciphers, Block cipher modes of operation, Drawback of symmetric key cryptosystem, Need for asymmetric key cryptography. **14 Hours** **Unit 3: Public Key Cryptography and Hashing techniques.** Introduction, Mathematics- II: Prime number, Primitive root, Prime factorization, The RSA algorithm, Diffie-Hellman key exchange, Elgamal cryptographic system, Elliptic curve cryptosystem, Digital Certificates, Cryptographic hash functions: Introduction, Applications, Collision resistance, Secure Hash Algorithm (SHA), Message Digest algorithm (MD5). **14 Hours** **Unit 4: Hashing Techniques and key** Key-Distribution Centre (KDC), Kerberos, Public Key Infrastructure (PKI) and standard (PKCS), Birthday attack, Entity authentication methods: Password, Challenge-Response, Zero knowledge protocols. Introduction to Post-Quantum cryptography, Types of Quantum-Resistant Algorithms, Key Algorithms in PQC, PQC in Emerging Technologies. **14 Hours**						

TextBook(s):	1. Understanding Cryptography: A Textbook for Students and Practitioners ,Christoff Paar and Jan Pelzl, Springer 2010. 2. Cryptography and Network Security: Principles and Practice, William Stallings,7th Edition, Pearson, 2017. 3. Cryptography and Network Security,Behrouz A.Forouzan,3rd Edition,TataMcGraw Hill,2017.
Reference Book(s):	1. Introduction to Modern Cryptography",JonathanKatz,Yehuda Lindell,2nd Edition, CRC Press, 2015.
Course Outcome	● Explain classical encryption techniques, cryptographic primitives, and cryptanalysis methods, along with the mathematical foundations that support secure communication. [Bloom's Level: Understanding] ● Implement and analyze symmetric encryption algorithms such as DES and AES, including their structure, operation modes, and cryptographic strength. [Bloom's Level: Applying, Analyzing] ● Apply number-theoretic concepts to design and evaluate public-key algorithms including RSA, Diffie-Hellman, ElGamal, and elliptic curve cryptography. [Bloom's Level: Applying] ● Analyze and apply cryptographic solutions in real-world contexts such as secure messaging, e-payments, blockchain, HTTPS, and metaverse environments. [Bloom's Level: Analyzing, Applying] ● Discuss and explore post-quantum cryptographic principles and their role in securing future-proof cryptosystems. [Bloom's Level: Understanding, Evaluating]

Course Code	UE23CS342AA5	Course Title	Virtual and Augmented Reality				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Virtual and Augmented Reality - C/ C++/ JAVA/ Pythonusing OpenGL. Unit3D, ThreeJS, Blender, Cloud Compare, OpenGL, Unity ML Agents	Desirable Knowledge	UE23CS252A- Data Structures.				
Prelude	This course presents an introduction to virtual and augmented reality technologies, with an emphasis on designing and developing interactive virtual and augmented reality experiences using blender and Unity3D.						

Course Objectives:

- Introduce the use of geometric transformations on graphics objects, their application in composite form and its implementation.
- Impart the basics of computer graphics and Introduce graphics programming using OpenGL and Graphics Pipeline.
- Introduce Virtual and Augmented Reality essentials and understanding human physiological aspects with respect to virtual reality applications and user interfaces.
- Understanding of use of Artificial Intelligence in the field of Virtual and Augmented Reality and applying the concepts to create creative virtual experiences.

Course Contents

Unit 1: Geometric Objects and Transformations

Scalars, Points and Vectors, Three-Dimensional Primitives, Coordinate Systems and Frames, Modelling a Coloured Cube, Overview of 2D Transformations: Rotation, Translation and Scaling, Affine transformations, Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Interfaces to Three Dimensional Applications, Quaternion's.

14 Hours

Unit 2: Graphical System and Programming and 3D Modelling

The Programmer's Interface, Graphics Architectures, Programmable Pipelines. Graphics Programming: Programming Two Dimensional Applications. The OpenGL: The OpenGL API, Primitives and Attributes, Colour, Viewing, Control Functions, Polygons, Viewing, Control Functions, the gasket Program, Polygon and Recursion, The Three-dimensional gasket, Creating Scene on ThreeJS, Working with Three.Js Material and Geometries, Animations, Adding Physics, Working with Blender, Three.JS with React and WebXR.

14 Hours

Unit 3: Augmented and Virtual Reality.

Introduction to Augmented Reality: Definition and Scope, A Brief History, Examples, Requirements and Characteristics: Methods of Augmentation, Spatial Display Models, Visual Display, Stationary Tracking Systems, Mobile Sensors. Introduction: What is Virtual Reality, Modern VR Experience. Bird's Eye View: hardware, Software. Eye movement and its implications or VR. Tracking: 2D and 3D orientation, Tracking Position and Orientation, Tracking Attached bodies.

14 Hours

Unit 4: IO modalities, AI, and Behaviour in VR

Computer Vision and Augmented Reality; marker tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Incremental Tracking, Simultaneous Localization and Mapping, Outdoor Tracking, Interaction: Output and input modalities, Haptic interaction and Multimodal interaction. 3D Scanning of environments. Reactive AI: Adaptability, Complexity and Universality, Feasibility, More Intelligence in the System: Deliberative AI, Reinforcement learning through interaction, Imitation Learning

	through human demonstration. **14 Hours**
TextBook(s):	1. "Interactive Computer Graphics - A top-down approach with shader-based OpenGL", Edward Angel and Dave Shreiner, Pearson Education, Sixth edition, 2012. 2. Steven M. LaValle. Virtual Reality. Cambridge University Press, 2017, http://vr.cs.uiuc.edu/ (Links to an external site.) (Available online for free). 3. Creating Augmented and Virtual Realities, by Erin Pangilinan, Steve Lukas, Vasanth Mohan, PUBLISHED BY:O'Reilly Media, Inc.PUBLICATION DATE:March 2019.
Reference Book(s):	1. "Interactive Computer Graphics: A Top -Down Approach with WebGL", Edward Angel, Pearson Education, 7th Edition, 2015. 2. Unity Game Development in 24 Hours , Geig, Mike. Sams Teach Yourself . Pearson Education, 2014. 3. "OpenGL Programming Guide": Mason Woo, Jackie Neider, Tom Davis, Dave Shrenier: 3rd Edition, openGl version 1.2, Addision Wesley, 1999. 4. Blender 3D Basic, Gordon Fisher, PACKT Publishing, 2nd Edition (Note: For working with recent version the course material for UNIT 4 can be substituted with appropriate web content. 5. D. Schmalstieg and T. Höllerer. Augmented Reality: Principles and Practice. Addison-Wesley, Boston, 2016, ISBN-13 978-0-32-188357-5
Course Outcome	• Apply techniques and methods of augmenting virtual objects in real space. • Demonstrate the fundamentals of computer graphics and display pipeline systems and use OpenGL for complex 3D graphical visualization and demonstrate its applications. • Apply techniques and tool to design a immersive virtual reality experience and graphics in greater depth to more complex aspects of Image Processing, Tracking and Human Computer Interfaces, etc. • Apply AI models and methods for creating dynamic interactive and adaptable virtual spaces.

Course Code	UE23CS342AA7	Course Title	Topic in Wireless and 5G Networks				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - I				
AI Tools /Tools/Languages	Cisco Packet Tracer, OMNET++	Desirable Knowledge	UE23CS252B – Computer Networks				

Prelude

This course offers an in-depth understanding of wireless communication systems, covering WLAN, WPAN, and cellular networks from 1G to 5G. Students will explore key topics like wireless access methods, modulation schemes, MIMO, LTE, and 5G network architecture. The course also delves into emerging areas such as Network slicing, OpenRAN, SDN/NFV, 5G security, and applications in IoT, AI/ML, and digital twins. With a forward-looking perspective on 6G, the course balances theoretical knowledge with practical skills, preparing students for careers in wireless communication and next-generation mobile networks.

Course Objectives:

- Understand fundamental wireless technologies, IEEE 802.11 standards, and access methods.
- Learn modulation techniques, cellular architecture, and evolution of mobile communication 1G to 5G.
- Explore 4G and 5G architecture, protocol stack, authentication, and signaling mechanisms.
- Introduce 5G applications, security, SDN/NFV, and emerging trends toward 6G.

Course Contents

Unit 1: Overview of Wireless communication

Introduction, Different Wireless Networks – WWAN, WBAN, WPAN, WLAN , Different Wireless Standards of 802.11, Wireless Local Area Networks: IEEE 802.11, 802.11 Frame Format, Basic Access Methods – CSMA/CA, ALOHA, Distribution Coordinating Function (DCF), DIFS, SIFS, Point Coordination Function (PCF), Multiple Access for Wireless Systems – FDMA, TDMA, CDMA, OFDMA, 802.11 Physical Media, WPAN Technologies: Bluetooth, Bluetooth Protocol Stack, NFC, 6LOWPAN, LORAWAN, WiMAX, Adhoc Routing Protocols – DSDV, AODV, Dynamic Source Routing (DSR), MANET & VANET, Wireless networking and security issues in 802.11 – Static filtering based on MAC address, Wired Equivalent Privacy (WEP), WPA & WPA2, Wireless Link Characteristics, RFID: Concept, frequency band, classification of RFID tags, applications.

14 Hours

Unit 2: Advanced Wireless Technologies and Cellular Standards

Wireless Modulation Schemes – Amplitude Modulation (AM), Phase Modulation (PM), Frequency Modulation (FM), Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), Frequency Spectrum, Cellular Concept –basic mobile communication, Mobile systems concept: mobile terminal, base station (BS), mobile switching centre (MSC), Frequency Reuse, Techniques: Multiple-Input and Multiple-Output (MIMO), mmWave, massive MIMO, Small Cells, Handoff, Mobile Radio standards – AMPS, GPRS, GSM, UMTS, CDMA 2000, Evolution of Wireless Network generations (1G, 2G, 3G, 4G, 5G), How does a mobile connectivity work? Global System for Mobile Communication (GSM), 3G UMTS – architecture, Need for 3G, 4G and 5G technologies – architectural difference.

14 Hours

Unit 3: Fundamentals of Cellular System

4G – LTE – EPC architecture, LTE protocol stack, UMTS vs LTE, Configuring LTE Control and Data Plane, Authentication Key Agreement in 4G LTE, Introduction to 5G, 5G Architecture, features, specifications and other procedures, 5G-SA/NSA, 5G vs 4G architectural differences, Simple call registration flow in 5G – AKA Authentication, Session establishment in 5G call flow, Signaling System No. 7 (SS7).

14 Hours

	Unit 4: 5G Applications and Evolution to 6G OpenRAN, Network Slicing, Long Range Communication- Satellite Communication, VoLTE, SDN, NFV, Private 5G networks, Application use cases: Digital Twin, IoT, AI/ML, Satellite Communications, 5G Security and Beyond, Mobile Forensics, 5G Security, Introduction to 6G. **14 Hours**
TextBook(s):	1. "Computer Networking: A Top Down Approach", James Kurose and Keith Ross, Pearson, 8th Edition.
Reference Book(s):	1. "Wireless and Mobile Network Architectures", Lin Yi-Bang and Clamtac Imrich, John Wiley & Sons 2001 2. "Wireless Communications and Networking", Vijay Garg, Morgan Kaufmann Publishers. 3. "Wireless Network Security: Second Edition, by Wolfgang Osterhage, CRC Press. 4. "Wireless and Mobile Networks: Concepts and Protocols", Dr. Sunil Kumar S. Manvi, Mahabaleswar S. Kakkasageri, Wiley India. 5. "Wireless Communication and Networks: 3G and Beyond", ITI Saha Mishra, McGraw-Hill Education. 6. "Wireless Communications: Principles and Practice", Theodore S. Rappaport 7. "Introduction to Mobile Network Engineering: SGM, 3G-WCDMA, LTE and the Road to 5G", Alexander Kukushkin, 2018 John Wiley & Sons Ltd 8. "5G Mobile and Wireless Communications Technology", AFIF OSSEIRAN, JOSE F. MONSRAT, PATRICK MARSCH, Cambridge University Press 9. "5G Mobile Networks: A Systems Approach", Larry Peterson and OgMuz Sunay, 2020, Morgan & Claypool Publishers.
Course Outcome	• Explain wireless standards, access protocols, and security in wireless networks. • Compare wireless modulation schemes and mobile communication generations. • Analyze 4G and 5G network architectures and key protocols. • Evaluate 5G applications, network slicing, and the transition to 6G technologies.

Course Code	UE23AM343AB1	Course Title	Advanced Data Analytics				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	Python and related libraries for Machine learning and Causal Inference.	Desirable Knowledge	Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.				
Prelude	In this digital transformation era, businesses across industries are embracing data science to gain insights, optimize processes, and drive strategic decisions. Two core pillars within this landscape are Data Mining and Data Analytics thinking. While traditional data science focuses on identifying correlation and building a machine learning model based on that, causal inference enhances data analytics thinking by diving into cause-and-effect relationships. Causal Inference is fast becoming an important area of data science in fields such as medicine, economics, sociology, psychology, and engineering, as it allows us to answer causal questions. This course aims to give students a broad perspective of Data Science by blending both. The course addresses them sequentially, i.e., starting with data mining, proceeding to do an end-to-end data science job by starting from data and then building a model, and finally elaborating on causal inference as an alternative approach in Data Analytics.						
Course Objectives:	● Learn to assess a dataset for the data type, inter-relationships of features, and interpretation. ● Understand regression models and forecasting techniques for time-varying data. ● Get introduced to the concept of causality and confounders. ● Learn various techniques for calculating causal impact.						
Course Contents:	**Unit 1: Introduction to Data Science, Understanding Data** Terminologies: Data Mining, Data Analytics, Business Analytics, Data Science, and Causal Inference. Data Science for Business– Data Mining vs. Data Analytics Thinking. Understanding Data: Describing Data - descriptive analysis, Understanding relationships and potential influences between different factors and their effect on outcome - ANOVA, Statistical measure of associative relationship between variables - Correlation analysis. Case Study. **14 Hours** **Unit 2: Predictive Analytics** Preparing Data for Machine Learning Tasks – Data Cleaning and Imputation, Data Transformation, Feature Selection. Regression-Simple and multiple linear regression, multivariate and nonlinear regression. Time Series - Time series data components, moving average and exponential smoothing methods, regression models for forecasting, AR model identification using ACF/ PACF, concept of stationarity – its importance, testing for stationarity and converting an on-stationary signal to stationary, ARMA and ARIMA modeling. Time Series modeling with Neural Network. Case Study. **14 Hours** **Unit 3: Introduction to Causal Inference** Introduction to Causal Inference– What is Causal Inference? Causal Inference workflow, Simpson's Paradox, Causal inference vs. Traditional Data Analytics, Potential Outcome, Individual Treatment effect, Fundamental problem of CI, Factual and Counterfactuals, Judea Pearl's Ladder of Causation, Average Treatment Effect, Associational Difference. Experimental studies A/B testing and RCT, limitations of A/B testing and RCT. Observational Studies - Causal effect under confounding, Intervention vs. RCT. Average Treatment Effect (ATE) and Adjustment formula. Estimating Causal effect With single binary Treatment variables. Conditional Average Treatment Effect (CATE). Assumptions and challenges in Causal Inference. Formulating the Causal Inference problem with a graph – Causal DAG. Dealing with Complex Graph – Fork, chain, and Colliders, d-separation. Backdoor criteria. Case Study. **14 Hours** **Unit 4: Estimating Causal Effect** With multiple binary treatment variables – using meta-learners such as S-learner, T-learner, X-learner algorithms. With continuous treatment variables - Propensity Score, using propensity score to check						

	positivity assumption, and calculation of ATE. Estimating Causal Effect with Linear Models. Double ML – introduction to Double Machine Learning, Heterogeneous Treatment Effect, Confidence Interval. Do Why and EconML. Case Study. **14 Hours**
Text Book(s):	1. "Business Analytics, The Science of Data-Driven Decision Making", U. Dinesh Kumar, Wiley 2022, second edition. 2. "Lecture Notes on Introduction to Causal Inference from a Machine Learning Perspective by Brady Neal, 2020", available online: https://www.bradyneal.com/causal-inference-course. 3. "Causal Inference and Discovery in Python: unlock the secrets of modern causal machine learning with DoWhy, EconML, Pytorch and more": by Aleksander Molak (Author), Ajit Jaokar (Foreword), May 2023.
Reference Book(s):	1. "The Book of Why: The New Science of Cause and Effect" by Judea Pearl and Dana Mackenzie. 2. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking", Foster Provost, 2013. 3. Data Mining: Concepts and Techniques by JiaweiHan, Micheline Kamber and Jianei, The Morgan Kaufmann Series in Data Management Systems, Elsevier publications, 3rd Edition, 2012.
Course Outcome	● Perform data analysis using techniques such as ANOVA and Correlation Analysis. ● Perform predictive analytics for regression setup and time series analysis for a dataset with a temporal dimension. ● Learn to use causal inference in the presence of confounding variables. ● Calculate Average Treatment Effect using various methods.

Course Code	UE23AM343AB2	Course Title	Privacy-Preserving Machine Learning				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	Python libraries and tools used for PPML, such as PySyft.	Desirable Knowledge	Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.				

Prelude	Fueled by recent advancements in data science, machine learning, and AI, our data is continuously being collected. While machine learning models trained on this data can substantially benefit individuals and society, they pose significant risks, including serious privacy breaches. Therefore, we face the challenge of balancing two opposing goals, i.e., maximizing the utility of machine learning models while safeguarding the privacy of individuals whose data is analyzed. How can we achieve high utility and robust privacy, or at least find an optimal trade-off? This interdisciplinary course addresses this issue by focusing on differential privacy—a mathematical approach that provides strong guarantees—and explores an algorithmic framework for developing practical privacy-preserving algorithms for data analytics and machine learning. The course also deals with modern methods of decentralized privacy, i.e., Federated learning and compressive privacy, and ends with applications of techniques learned in the healthcare domain.
Course Objectives:	• Learn the concept of privacy-preserving machine learning (PPML) and differential privacy. • Learn differential and local differential privacy. • Learn privacy preservation measures in data generation. • Learn decentralized privacy, compressive privacy.
Course Contents:	**Unit 1: Introduction to Privacy-Preserving Machine Learning and Differential Privacy Foundation** Need - Privacy consideration in machine learning, the threat of learning beyond the purpose, Threats, and attacks for ML systems - private data in clear, reconstruction attack, model inversion attack, membership inference attack, de-anonymization attack, the challenge of privacy protection in Big Data analysis. Privacy-preserving machine learning (PPML) overview–Differential Privacy, Local Differential Privacy, Privacy Preserving Synthetic Data Generation, Privacy-Preserving Data Mining, and Compressive Privacy. Differential Privacy- concept of differential privacy, mechanisms of differential privacy – Binary Mechanism, Laplace Mechanism, Exponential Mechanism. Properties of differential privacy –postprocessing property, group privacy property, composition property. **14 Hours** **Unit 2: Applying Differential Privacy in Machine Learning and Local Differential Privacy** Applying Differential Privacy in Machine Learning- input perturbation, algorithm perturbation, output perturbation, objective function perturbation. Differentially private supervised learning algorithm–naïve bayes, logistic regression, and linear regression. Differentially private unsupervised learning algorithm – k-means. Differentially private dimensionality reduction algorithm – PCA. Local differential privacy - Mechanisms of local differential privacy (LDP) - randomized response, direct encoding, histogram encoding, binary encoding, Laplace mechanism, Duchi's mechanism, Piecewise mechanism. Applying local differential privacy in machine learning — LDP Naïve Bayes with discrete and continuous features. Evaluating the Performance of various LDP protocols. **14 Hours** **Unit 3: Privacy Preserving synthetic data generation** Importance of synthetic data, application of synthetic data in privacy preservation. Assuring privacy via data anonymization- private information sharing vs. privacy concern, k-anonymity against re-identification attack, anonymization beyond k-anonymity. Differential privacy techniques for synthetic data generation. Case Study on Private Synthetic Data Generation. Privacy Preserving in data processing–protecting privacy by modifying input, protecting privacy when publishing data – data sanitization by k-anonymity, l-diversity, t-closeness. **14 Hours** **Unit 4: Decentralized Privacy, Compressive Privacy**

	Introduction to Decentralized Privacy-Preserving Machine Learning Algorithms - Federated Learning. Compressive privacy- Concept. Mechanisms of compressive privacy - Principal Component Analysis and Discriminant Component Analysis (DCA). **14 Hours**
Text Book(s):	1. Privacy-Preserving Machine Learning -J Morris Chang, D Zhuang, Manning. 2. Privacy-Preserving Machine Learning – Srinivasa Rao Aravilli and Sam Hamilton, Packt Book, 2024.
Reference Book(s):	1. Privacy Preserving Machine Learning (Springer Briefs on Cyber Security & Networks), 2022. 2. Privacy Preserving Data Mining: models and applications, Charu C Agarwal, 2008. 3. C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science, 2014.
Course Outcome	● Implement PPML for machine learning algorithms. ● Implement differential and local differential privacy preservation for machine learning algorithms. ● Implement privacy preservation techniques in data generation systems. ● Implement compressive privacy and decentralized differential privacy for Machine Learning systems.

Course Code	UE23AM343AB3	Course Title	Optimization and Metaheuristics				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	Python and various search and optimization libraries in Python, such as scikit-opt and scikitlearn, pyswarm, scipy, DEAP, etc	Desirable Knowledge	Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.				

Prelude

Optimization is a ubiquitous challenge in different scientific problems, specifically machine learning, and this course equips students with a broader understanding of optimization by complementing the learning of optimization in other machine learning courses. It deals with metaheuristics-based approaches for search/optimization problems for non-differentiable objective functions. The course starts by discussing the deterministic search algorithms of both types, i.e., blind and informed search, and then forays into stochastic approaches such as trajectory-based heuristics and population-based heuristics. Ultimately, the course connects the dots between derivative-free and derivative-dependent optimization by applying stochastic metaheuristics in hyperparameter optimization in machine learning models.

Course Objectives:

- Learn search vs. optimization and deterministic search algorithms.
- Learn stochastic trajectory-based metaheuristics.
- Learn stochastic population-based metaheuristics.
- Learn metaheuristic-based derivative-free optimization in machine learning.

Course Contents:

Unit 1: Introduction to Search and Optimization, Deterministic Search Algorithms - Blind and Informed Search

Search and optimization – Introduction to search and optimization. Basic ingredients of an optimization problem and Classification of optimization problems – variables, constraints, and objective functions. Well-structured and ill-structured problems. Exploration-Exploitation dilemma in search. Heuristics, Metaheuristics, and Nature-inspired algorithms. Classification of search & optimization algorithms – local vs. global search, Deterministic vs. Stochastic algorithms. Deterministic Blind search on graphs - Graph Traversal algorithms & Shortest Path Algorithms (uniform cost search, Bidirectional Dijkstra). Deterministic Informed Search algorithm – Minimum Spanning Tree, Shortest Path Algorithms (Hill climbing, Beam Search, and A* search). Applications- Blind Search and Informed Search in the Routing Problems.

14 Hours

Unit 2: Trajectory-based Stochastic metaheuristics

Simulated Annealing – Physical annealing, pseudocode, Transition probability, Cooling schedule, initial and final temperature, annealing schedule. Case Study in continuous and discrete optimization problems.

Tabu Search (TS) – Enhance the opportunity for local search. Tabu Search algorithm – memory structure, aspiration, and adaptation. Applications – solving Constraint satisfaction problems, Solving TSP and Routing Problems, and Solving continuous problems.

14 Hours

Unit 3: Population-based Stochastic metaheuristics

Evolutionary, Nature-Inspired, and Swarm intelligence algorithms. Genetic Algorithm (GA) - Binary Genetic Algorithm. Implementation of genetic algorithm in Python. Real valued GA. Permutation-based GA. Swarm intelligence- Particle Swarm Optimization (PSO)- Continuous PSO - motion equation, initialization, fitness update, neighborhood. Binary PSO. Permutation PSO. Adaptive PSO. Solving TSP Problem using PSO. Ant Colony Optimization (ACO) - ACO metaheuristics.

14 Hours

Unit 4: Evolutionary Machine Learning

Opportunities of evolutionary computing in machine learning – hyperparameter optimization and model selection. Hyperparameter optimization (HPO)– Applying Evolutionary Computing such as PSO and GA to Hyperparameter Optimization. GA optimization in MLP - Multi-layer perceptron (MLP) in Numpy and Applying GA optimization to MLP.

	14 Hours
Text Book(s):	1. "Natural Computing Algorithms", Anthony Brabazon, Michael O'Neill, Seán McGarraghy, Springer, Natural Computing Series, 2015. 2. "Evolutionary Deep Learning: Genetic algorithms and neural networks" , Micheal Lantham, Manning Books, 2023. 3. "Algorithms for Optimization" , Mykel J. Kochenderfer and Tim A. Wheeler, MIT Press, 2019.
Reference Book(s):	1. "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", Floreano D. and Mattiussi C., MIT Press, Cambridge, MA, 2008.
Course Outcome	● Should be able to implement informed search metaheuristics. ● Should be able to implement trajectory-based metaheuristics. ● Should be able to implement population-based metaheuristics. ● Should be able to implement metaheuristics-based derivative-free optimization in machine learning.

Course Code	UE23AM343AB4	Course Title	Supply Chain Optimization Using AI				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	Python and various search and optimization libraries.	Desirable Knowledge	-				
Prelude	This course explores the application of Artificial Intelligence (AI) techniques to optimize supply chain management (SCM) processes. It covers AI fundamentals relevant to SCM, including predictive analytics, machine learning models, and decision-making algorithms. The course aims to equip students with the skills to design, implement, and evaluate AI-driven solutions that enhance efficiency, reduce costs, and improve supply chain resilience.						
Course Objectives:	● Understand the key components and challenges of modern supply chains. ● Learn about the various AI technologies applicable to SCM optimization. ● Explore case studies of AI applications in forecasting, logistics, inventory management, and procurement. ● Develop and implement AI models to solve real-world SCM problems.						
Course Contents:	**Unit 1: Introduction to SCM and AI** Overview of supply chain management, Basics of AI and its relevance to SCM, Review of AI technologies: Machine Learning, Neural Networks, Natural Language Processing, History and evolution of AI in SCM, Role of big data in SCM, Introduction to AI tools and platforms for SCM optimization. **14 Hours** **Unit 2: AI in Supply Chain Forecasting and Planning** Demand forecasting models, Predictive analytics for supply chain planning, Case studies on AI-driven forecasting, Techniques for improving demand accuracy, AI in production planning and scheduling, Scenario analysis and simulation using AI, Case studies on AI applications in supply chain planning. **14 Hours** **Unit 3: AI in Inventory and Logistics Management** Inventory optimization techniques, AI for warehouse management and automation, Enhancing transportation and logistics with AI, Real-time tracking and monitoring with AI, AI-driven route optimization, Role of AI in last-mile delivery, Case studies on AI applications in logistics. **14 Hours** **Unit 4: AI in Procurement and Supplier Relationship Management** AI applications in procurement processes, Supplier selection and performance assessment with AI, Risk management and compliance using AI, AI in contract management, AI-driven negotiation strategies, Blockchain and AI for supply chain transparency, Case studies on AI applications in procurement. **14 Hours**						
Reference Book(s):	1. "Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies" by Hartmut Stadtler and Christoph Kilger. 2. "Artificial Intelligence for Supply Chain Management: Techniques and Applications for Improved Efficiency and Effectiveness" by N. Viswanadham. It focuses on how AI can be applied to various aspects of supply chain management to improve operations. 3. "Data Science for Supply Chain Forecasting" by Nicolas Vandeput.						
Additional Resources:	● Access to AI development tools and software for SCM applications. ● Collaboration with industry partners for practical insights and case studies.						
Course Outcome	● Identify areas within SCM that can be optimized using AI. ● Apply AI and machine learning techniques to predict trends, optimize inventory, and improve logistics. ● Evaluate the impact of AI-driven solutions on supply chain performance. ● Design a project that implements an AI solution for a specific aspect of SCM.						

Course Code	UE23AM343AB5	Course Title	Machine Learning with Business Data				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	Python along with standard machine learning libraries such as ScikitLearn, XGBoost,LightGBM, CatBoost will be used. Specific to tabular data, the libraries like TabNet, Pytorch Tabular, etc.,	Desirable Knowledge	Mathematics for Computer Science Engineers, Linear Algebra and its Applications. Also, students should be undertaking Machine Learning in parallel.				

Prelude

Today, immense public interest, most research initiatives, and most commercial incentives are on text and image data. In this course, we shift the priority slightly to the modality of the data on which the business mainly operates, i.e., tabular data. Tabular data is ubiquitous in the real world, i.e., electronic healthcare records, census data, cybersecurity, credit scoring, finance, and natural sciences. Quantitative research relies on these datasets, which in turn progress scientific knowledge and influence public policy. This hands-on course investigates data analytics and machine learning on tabular datasets and prepares students aspiring for an industry-embedded data scientist role in the future, while offering a much-needed hands-on experience for the machine learning concepts.

Course Objectives:

- Learn the importance of tabular business data and exploratory analysis on such data.
- Learn classical ML approaches on tabular business data.
- Learn deep learning libraries on tabular business data.
- Learn how to ensemble classical and deep learning methods while providing interpretability.

Course Contents:

Unit 1: Introduction to Tabular Data and Exploring Tabular Datasets

Why Tabular Data – the world runs on tabular data! What makes tabular data different? Characteristics of Tabular Data – row and column characteristics, possible issues in tabular business data, and remedies. Exploratory Data Analysis of Tabular Business Data. Classical Machine Learning vs. Deep Learning on Tabular Business Data. Case Study – Exploratory data analysis of a tabular business dataset, comparing Deep Learning and Classical Machine learning approaches – transparency and efficacy of the approaches.

14 Hours

Unit 2: Classical Machine Learning with Tabular Business Data

Exploring and processing a tabular dataset – Scikit Learn pipeline, Decision Tree - extremely randomized trees, Gradient Boosting – explaining its effectiveness and applying early stopping to avoid overfitting, XGBoost – key parameters, applying early stopping. Advanced Feature processing –handling missing data imputation, handling missing data with Gradient Boosting algorithms, Feature selection methods – stability selection, Boruta, Forward and Backward selection. Optimizing hyperparameters – random, grid search, Bayesian, and manual methods. Case Study - An end to end implementation on a tabular business dataset using XGBoost.

14 Hours

Unit 3: Deep Learning with Tabular Business Data

Deep Learning with Tabular Data Ecosystem (low level API, high level API, tabular data library): Tensorflow stack – Tensorflow, Keras, (Tabnet, DeepTable), Pytorch stack –Pytorch, Lightning, (Pytorch Tabular, Tabnet, Lightning Flash, fastai) , Pytorch with Tabnet and Lightning Flash, Case Study – Deep Learning on a tabular Business dataset : processing the columns, defining the model, Training the model –cross validation, regularization, normalization.

14 Hours

Unit 4: Blending XGBoost and Deep Learning, Interpretability

Blending Gradient Boosting and Deep Learning solution – select a tabular business dataset, selecting a deep learning solution on a platform (fastai, tabnetetc), Comparing XGBoost and selected deep learning solution, Ensemble the two methods, comparison.

Interpretability of XGBoost: Interpretability vs. Explainability. Types of interpretability techniques – local vs. global, model specific vs. model agnostic, intrinsic vs. posthoc.XGBoost feature Importance.

SHAP (Shapely Additive exPlanations), SHAP Bar plot, Summary plot, Dependence plot, Force plot. Comparing XGBoost Feature Importance vs. SHAP.

14 Hours

Text Book(s):	1. "Applied Machine Learning for Tabular Data", by Max Kuhn and Kjell Johnson, available online for reading 2. "Modern Deep Learning for Tabular Data: Novel Approaches to Common Modelling Problems", Andre Ye and Zian Wang, Apress, 2022. 3. "Machine Learning for Tabular Data: XGBoost, Deep Learning, and AI", Mark Ryan and Luca Massaron, Manning Book, 2025
Reference Book(s):	1. "Why Tabular Foundation Models Should Be a Research Priority", Boris van Breugel and Mihaela van der Schaar, arXiv. 2. "Deep Neural Network and Tabular Data - a Survey",Vadim Borisov et al., IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 2024 3. "TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED", Ravid Shwartz-Ziv and Amitai Armon, Information Fusion, 2021. 4. "Explainable Artificial Intelligence for Tabular Data : a Survey " , MARIA SAHAKYAN et al. IEEE Access, 2021 5. " A Short Chronology Of Deep Learning For Tabular Data", Sebastian Raschka, 2022, Available online 6. "Why do tree-based models still outperform deep learning on typical tabular data?", Léo Grinsztajn, Edouard Oyallon, Gaël Varoquaux, HAL Open Science, 2023.
Course Outcome	● Do exploratory analysis of a tabular business issues after addressing dataset issues. ● Do classical ML, specifically XGBoost, on tabular business data. ● Do deep learning models on tabular business data. ● Do an ensemble of deep and classical learning methods on tabular data and interpretability.

Course Code	UE23CS343AB5	Course Title	Advanced Computer Networks				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	GNS3, Cisco Packet Tracer, Mininet	Desirable Knowledge	Computer Networks.				

Prelude

The Advanced Computer Networks course is designed for students who have basic knowledge of networking (OSI model, TCP/IP, ARP etc.). The course builds its contents by considering a mixed approach involving theory and practical. From this perspective, the course introduces advanced topics that are basic blocks of networking and covers fundamentals that are used to develop few of the advanced technologies, including routing protocols and switching. The course aims to provide information so that at the end of the course, the students find themselves comfortable in taking either the direction- industrial job or further research in networking.

Course Objectives:

- Provide in-depth study of dynamic and distance vector routing protocols.
- To learn about Layer 2 Switching in detail.
- To get an insight of Link State routing protocols.
- To explore the new approach of Networking, SDN and NFV.

Course Contents

Unit 1: Dynamic and Distance Vector Routing, Access Lists

IP Routing Process, Static routing – IPv4/IPv6, Floating Static Routes, Load Sharing, Fast and Process Switching Distance Vector Routing – RIPv2 packet format, classless Routing, compatibility with RIPv1, Metric Calculation, Route Summarization, Standard IP and Extended IP access lists, Calling access lists, Named access lists. **14 Hours**

Unit 2: Exterior Gateway Routing Protocol (EIGRP), Layer 2 Switching, Spanning Tree Protocol (STP), Virtual LAN (VLAN) EIGRP – Neighbor discovery, Reliable Transport Protocol (RTP), Diffusing Update Algorithm (DUAL), VLSM and Summary Routes, Metrics – Maximum Paths & Hop Count, Load Balancing with EIGRP, Discontiguous Networks, Auto summarization, passive interface Layer 2 switching, Bridging vs LAN Switching, Forwarding Table, Spanning Tree Protocol and its operations, VLAN Basics, Static and Dynamic VLANs, VLAN Trunking Protocol (VTP), Routing between VLANs, configure VLANs and Inter-VLAN routing, MAC VLANs, VXLAN **14 Hours**

Unit 3: Link State Routing Protocols – OSPF & BGP

Open Shortest Path First (OSPF)v2 – OSPF packet format, Network Types, Designated Routers and Backup Designated Routers (DR & BDRs), OSPF Neighbors, Areas – Intra, Inter, External, Router Types, Virtual Links, Link State Database, LSA Types, Metric Calculation, Summary Routes, Route redistribution Border Gateway Protocol (BGP) – BGP Packet Format, External and Internal BGP. **14 Hours**

Unit 4: Introduction to Software Defined Networks (SDN) and Network Function Virtualization (NFV) Segment Routing, SDN Control and Data Plane, SDN Controller and Network-control Applications, OpenFlow Protocol, Data and Control Plane Interaction Network Virtualization architecture, Management, Control and Data Planes, Virtual Switches, Micro-segmentation. **14 Hours**

Text Book(s):	1. "Routing TCP/IP volume 1", 2nd Edition, Jeff Doyle, Cisco Press 2005. 2. "Computer Networking: A Top-Down Approach", James F. Kurose, Keith W. Ross, 7th Edition, Pearson Publication, 2017 [For SDN]
Reference Book(s):	1. "Software-Defined Networks: A Systems Approach", Peterson, Cascone, O'Connor, Vachuska, and Davie, https://sdn.systemsapproach.org/index.html [For NFV] "TCP IP Protocol Suite", Behrouz Forouzan, 4th Edition, McGraw-Hill, 2010 [For BGP]
Course Outcome	● Designing a network topology by demonstrating the ability to configure routers with different routing techniques. ● Understand the concepts of Layer 2 switching, VLANs and its usage. ● Designing networks with one or multiple routing protocols and investigating its behaviour/features. ● Comprehend features of Software Defined Networking (SDN) for next generation systems.

<table>
<tr><td>Course Code</td><td>UE23CS343AB6</td><td>Course Title</td><td colspan="5">Computer Network Security</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>5</td><td>Type of Course</td><td colspan="5">Elective - II</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.</td><td>Desirable Knowledge</td><td colspan="5">UE23CS252B – Computer Networks.</td></tr>
<tr><td>Prelude</td><td colspan="6">This course will give an overview and conceptual understanding of network related security aspects. Students will have opportunities to dwell well into technical “how to” with hands-on sessions and case study discussions.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● To provide an overall view of Network Security and introduce the concept of packet analysis.
● To understand the security problems in the design and implementation of the TCP, IP/ICMP, ARP protocols.
● To learn the vulnerabilities in DNS protocol and to implement and experiment with Firewall rules.
● To provide an overview of network management techniques and implementation of VPN. To understand concepts of risk management and security aspects of wireless networks.</td></tr>
<tr><td>Course Contents</td><td colspan="6">Unit 1: Introduction, Packet Sniffing & Spoofing, MAC Layer Attacks, Network Layer Attacks
CIA principles, Attack surface and types, Assets, Vulnerabilities and Threats, Countermeasures, Privacy, General Data Protection Regulation, Security vs Privacy, Data Breaches. Real Life Examples of Cyber Crime, Security framework, Job outlook. Packet Sniffing and Spoofing: Introduction, Sending packets: Network Interface Card (NIC), BSD packet filter (BPF). Packet sniffing: Receiving packets using sockets, Packet sniffing using Raw sockets, Packet sniffing using PCAP API, Processing captured packets. Packet spoofing: Sending normal packets using sockets, Constructing spoofed raw ICMP packets and UDP packets. Sniffing and then spoofing, Python vs Scapy, Endianness. MAC layer and attacks: The MAC layer, ARP protocol, ARP cache poisoning attacks, MITM using ARP cache poisoning, Demo, Countermeasure.
16 Hours
Unit 2: TCP Attacks and DNS Attacks
Network layer: IP, ICMP and attacks: IP protocol, IP fragmentation, Attacks using IP fragmentation: Problem and solution, Routing, and spoofing prevention, ICMP protocol, ICMP redirect attack, Smurf and other ICMP attacks. Attacks on the TCP protocols: TCP overview, Send and receive buffers, SYN flood attack: TCP 3-way handshake, the SYN flooding attack, Launching the attack using Netwox and C, Countermeasure. TCP reset attack: TCP reset attack on Telnet, SSH and video streaming connections. TCP session hijacking attack: TCP session and session hijacking, Launching the attack, Hijacked TCP connection. Reverse shell: working, redirecting IO to TCP connection, Creating reverse shell. Countermeasure. DNS Attacks: DNS hierarchy, zones and servers, DNS query process, Constructing DNS request and response using Scapy, DNS attacks: Overview, Local DNS cache poisoning attack.
12 Hours
Unit 3: Firewall, IDS, IPS, Honeypots
Remote DNS cache poisoning attack (Kaminsky attack), Reply forgery attacks from malicious DNS servers, Countermeasure against DNS spoofing attacks, DoS attacks on DNS servers. Case Study – 1.Firewall: Introduction, Requirements of a firewall, Firewall characteristics and Access policy, Types of firewalls, NG firewall, Shortcomings, Firewall location and configuration: DMZ networks, Firewall topologies. Introduction, Build a simple firewall, Netfilter, iptables firewall in Linux, Stateful firewall and connection tracking, Application/Proxy firewall and Web proxy, Evading firewalls.</td></tr>
</table>

	Intrusion Detection and Prevention: Intruders, Intrusion detection, Analysis approaches, Host-based intrusion detection, Network-based intrusion detection, Distributed or hybrid intrusion detection, Honeypots, Example system: Snort. **14 Hours** **Unit 4: Virtual Private Networks, Wireless security** Virtual Private Network: Introduction, Why VPN, analogy, and tunnelling. Overview of TLS/SSL VPN: Establishing a tunnel, Forwarding, and releasing IP packets, TLS/SSL VPN details. Building, Setup and Testing VPN. Bypassing Firewall using VPN. Case Study – 2. The Heartbleed Bug and Attack: Introduction and the Heartbeat protocol, Launching the attack, Fixing the Heartbleed bug. Wireless Security: Communications and 802.11 WLAN standards: Wired Equivalent Privacy (WEP), Wireless Protected Access (WPA), IEEE 802.1x, 802.11i/ WPA2, Wireless Network Threats. SOC and SIEM. **14 Hours**
Text Book(s):	1. Internet Security: A Hands-on Approach, Wenliang Du, 2nd Edition, 2019 2. Internet Security: A Hands-on Approach, Wenliang Du, 3rd Edition, 2022 3. "Computer Security: Principles and Practice", William Stallings and Lawrie Brown, 3rd edition, 2015
Course Outcome	● Sniff packets from clients and analyse them to extract important info such as headers, passwords etc. ● Launch DoS and MITM attacks using various protocol vulnerabilities and mitigate them. ● Configure firewalls on Linux machines and exploit vulnerabilities on DNS protocol. ● Design and implement VPN for a secure connection over the internet.Master in wireless network security systems in depth and perform effective network management.

Course Code	UE23CS343AB7	Course Title	ROS for Autonomous Systems				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	5	Type of Course	Elective - II				
AI Tools /Tools/Languages	C++, Python, Java, ROS, Rviz, Gazebo and Webots.	Desirable Knowledge	-				
Prelude	To introduce the foundational principles of autonomous systems, focusing on mobile robotics and programming using ROS (Robot Operating System). The course integrates theoretical understanding with simulation-based hands-on experience.						
Course Objectives:	• To introduce students to the fundamentals of autonomous robotics, including key components such as sensors, actuators, and controllers. • To familiarize students with the Robot Operating System(ROS) environment, including node creation, topic communication, and workspace setup. • To enable practical skills in integrating and programming common sensors and actuators in simulated robotic platforms. • To develop an understanding of robot perception, environment modeling, and mapping techniques such as SLAM. • To provide exposure to robot navigation using ROS frameworks and modern learning-based methods like reinforcement and imitation learning.						
Course Contents	**Unit 1: Introduction to Autonomous Robotics & Simulation Tools** What is a robot? Definitions, types, and criteria, Sensors, actuators, controllers, Autonomous vs. telerobot systems; Environment classification. Overview of ROS, its architecture, and installation, RSVP: Robot Scenario Visual Planning; Hands-on Activities: Installing ROS and setting up workspace, Creating and launching ROS nodes (publisher/subscriber) **14 Hours** **Unit 2: Sensor Integration, Motor Control and Simulation** Programming sensors: Color, ultrasonic, IMU, camera, Motor/servo characteristics and programming, Sensor packages and interfacing in ROS; Hands- on Activities: Simulating robot with LiDAR/camera in Gazebo, Sensor visualization with RViz and ROS topics, Motor movement control in a ROS simulation. **14 Hours** **Unit 3: Perception, Mapping, and Environment Modelling** Environment perception: Coordinate frames, TFs, Introduction to mapping: occupancy grids, SLAM, RSVP and SPACES methodology for decision making; Hands-on Activities: Building a 2D map using SLAM, Simulating perception using TurtleBot3 in Gazebo. **14 Hours** **Unit 4: ROS Navigation** Localizing the robot in a map, ROS Navigation Stack-hardware requirement- navigation packages, path planning, motion planning of robot – software requirement and configuration. Navigation using Reinforcement learning and Imitation learning. **14 Hours**						

Text Book(s):	1. Lentin Joseph, Robot Operating System (ROS) for Absolute Beginners: Robotics Programming MadeEasy, 1st Edition, APress, 2018. 2. Jonathan Cacace; Lentin Joseph, Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System, 2nd Edition, Packt Publishing, 2018.
Reference Book(s):	1. Hughes, C. and Hughes, T., Robot programming: a guide to controlling autonomous robots. Que Publishing, 2016. 2. Quigley, M., Gerkey, B. and Smart, W.D., Programming Robots with ROS: a practical introduction to the Robot Operating System. " O'Reilly Media, Inc.", 2015. **Recommended Materials: Robot Operating System (ROS): http://wiki.ros.org/ROS/Tutorials ROS2 tutorials: https://docs.ros.org/en/foxy/Tutorials.html**
Course Outcome	● Identify and describe the essential components and classifications of autonomous robotic systems. ● Configure and implement ROS-based simulations involving sensor-actuator integration using standard tools like Gazebo and RViz. ● Apply SLAM and environment modeling techniques for autonomous robot perception and mapping in simulated environments. ● Develop and test navigation algorithms using the ROS Navigation Stack with path and motion planning capabilities. ● Demonstrate foundational knowledge of learning-based navigation strategies such as reinforcement and imitation learning in robotics.

Course Code	UE23CS351B	Course Title	Cloud Computing				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	2	5	5
Semester	6	Type of Course	Core				
AI Tools /Tools/Languages	Amazon AWS (or equivalent), AWS SkillBuilder, AWS Educate, Qwiklabs, Docker, Kubernetes, Jenkins, Zookeeper, Github, NoSQL database, Flask, Python, Go Lang. AWS Sagemaker	Desirable Knowledge	UE23CS252B - Computer Networks UE23CS241B- Operating Systems.				
Prelude	The cloud computing course introduces not only the various technologies that go into building a cloud native application, but also how cloud systems are designed. The student is introduced to various tools and design techniques/trade-offs. It also gives a flavour for the business relevance/ethics of using cloud computing. This course requires the student to have a desirable knowledge of Computer Networks and Operating System.						
Course Objectives:	● Introduce the rationale behind the cloud computing revolution and cloud native application architecture. ● Explore the concepts of Virtualization and Containerization. ● Explore the concepts of Distributed storage and its various techniques. ● Design distributed systems for scalability and expose the student to various trade-offs in designing cloud architectures.						
Course Contents	**Unit 1: Cloud Programming Models** ParallelcomputingGridcomputingIntroductiontoCloudProgrammingModelsandserviceModels, Introduction to technology challenges with Distributed & Cloud computing, Business Drivers - deployment models, Multi-cloud, Cloud architecture and IaaS programming model, Web Services and REST, PaaS Programming Model, Communication using Message queues- Pub Sub model, SaaS Programming model – Microservices and differences with the traditional monolithic model; challenges of migrating monolithic applications. Microservice transactions with Saga pattern. **14 Hours** **Unit2: Virtualization** Hypervisor - Types, Para virtualization and Transparent virtualization, Software - Trap and Emulate virtualization,Software-BinarytranslationGoldbergPopekprinciplesforVirtualizationHardware - AMDv/Intel, Memory - Shadow page tables, Memory - Nested page tables, IO, VM Migration, Lightweight Virtualization - Containers and Namespaces, Deployment of cloud native applications through Docker – Unionfs, Orchestration and Kubernetes, DevOps, Jenkins pipeline. **14 Hours**						

	Unit 3: Distributed Storage Types of Cloud storage - Block, Object stores, Replication, lag, multileader replication, Leaderless replication, Partitioning-key-value data, Consistent hashing, Partitioning -rebalancing partitions, Request routing, Consistency Models, CAP Theorem, Transactions, Two-phase commit. **14 Hours** **Unit 4: Cloud Controller, Performance, Scalability and Security** Master-slave v/s p2p models, Resource allocation, Scheduling algorithms, Cluster coordination – consensus, Fault Tolerance - faults and partial failures, Failure detection - checkpointing and application recovery, Unreliable communication, Cluster coordination -leader election, distributed locking,CaseStudy: Zookeeper/Raft-distributed consensus infrastructure. Edge Computing and Fog Computing – principles and paradigms Scaling computation - reverse proxies, Scaling computation - hybrid cloud and cloud bursting, Multitenancy, Multitenant databases, Cloud security requirements - physical/virtual security, security design patterns, Container security features to work with Docker and Kubernetes, Authentication in the cloud:Keystone/IAM,Cloud Threats –Dos, Economic Denial of Sustainability. **14 Hours**
Laboratory	1. Migrating a monolithice-commerce application to a microservices architecture. 2. BuildingaTask Management Application with Raft Consensus Algorithm and MySQL. 3. Microservices communication using RabbitMQ. 4. Building a Distributed Key-Value Store with etcd. 5. BackUp service using docker and Kubernetes. 6. Building an E-commerce Microservices Application on Cloud using Docker, Kubernetes, Jenkins, and Git.
TextBook(s):	1. “Distributed and Cloud Computing”, KaiHwang, Jack Dongarra, Geoffrey Fox. ISBN:978-0-12-385880-1, Morgan Kaufmann, 2012. 2. “Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems”, Martin Kleppmann. O’s Reilly, 2017. 3. “Fog and edge computing: principles and paradigms”/edited by Rajkumar. Buyya, Satish Narayana Srirama. Description:Hoboken,NJ,USA: JohnWiley & Sons.
Reference Book(s):	1. “Docker in Action”,Jeff Nickoloff, Manning Publications,2016. 2. “Cloud Native DevOps with Kubernetes”,John Arundel and Justin Domingus, O Reilly,2019.: 3. “Moving to the clouds:Developing Apps in the new world of cloud computing”,Dinkar Sitaram and Geetha Manjunath. Syngress, 2011.
Course Outcome	● Comprehend the technical and business rationale behind cloud computing. Decide the model of cloud computing to use for solving a particular problem. ● Implement Microservice architecture through Containers and Orchestration tools. Analyse virtual machines and containers. ● Experiment with cloud storage Models such as object stores, key value stores. ● Apply the critical constraints such as Performance, scalability and security to the designed distributed system.

Course Code	UE23CS352B	Course Title	Object Oriented Analysis and Design				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	2	5	5
Semester	6	Type of Course	Core				
AI Tools /Tools/Languages	Star UML, Object Oriented Programming Language(Java/C++)	Desirable Knowledge	-				

Prelude

In this course students will learn to perform Analysis on a given domain and come up with an Object-Oriented Design (OOD). Various techniques will be discussed and practiced which are commonly used in analysis and design phases in the software industry. Unified Modelling Language (UML) will be used as a tool to demonstrate the analysis and design ideas and an object-oriented programming language such as Java would be used to implement the design. The theory is supplemented with implementations which are demonstrated/ practiced in class which provides the hands-on experiences of implementing the patterns.

Course Objectives:

- Familiarize students with static and dynamic models of object-oriented analysis and modelling using the unified modelling language (UML).
- Introduce students to object oriented programming concepts in Java.
- Make students appreciate the importance of system architecture design in software development.
- Introduce the students to understand the importance of GRASP and SOLID design principles.

Course Contents

Unit 1: Advanced OO, Object Oriented Analysis and Static Models and Diagrams

OOA: Requirements, Modelling and Analysis, Introduction to UML, Use Case Modelling: Use Cases Diagrams. Class Modelling: UML Class Diagrams, OO relationships, CRC Diagrams, Component model, Deployment model, Activity Modelling: UML Activity Diagrams and Modelling, Guidelines. Behaviour Modelling: Sequence Diagram, UML State Machine Diagrams and Models, Advanced State Models.

14 Hours

Unit 2: Object Orientated Programming and Architecture design

Object-oriented Programming: JVM, Abstraction, Encapsulation, Composition, Class Attributes, Behaviour, Objects, and Methods. Interface and Implementation: Role of Constructors and Destructors, Garbage Collector, Parameter Passing, Value Type and Reference Type, Overloading of Methods, Class Attributes and Behaviour:Difference between Class Methods and Instance Methods, Inheritance: Concepts of Single Rooted Hierarchy and Interface, Abstract Class in Programming Languages, Object Class in Java, Collection, Array, List and Stack; OO Development process, System Design and Frameworks, Architectural patterns, MVC architectural pattern.

14 Hours

Unit 3: Design principles

GRASP and its application to Object Design, Creator, Information Expert, Low Coupling, Controller, High Cohesion, Polymorphism, Pure Fabrication, Indirection and Protected Variations .SOLID: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion OO Design Principles and Sample Implementation of Patterns in Java. Introduction to Design Patterns, selection and usage of a design pattern, Creational Design Patterns Theory and Implementation in Java: Singleton, Factory,

	Builder and Prototype. **14 Hours** **Unit 4: OO Design Patterns & Anti-Patterns** Structural Patterns–Adapter, Façade, Proxy and Fly weight Behavioral Patterns–Chain of Responsibilities, Command, Interpreter, Iterator. Anti-patterns–Introduction and classification, Project Management, Architecture and Development anti-patterns (1 anti-patterns of each type) **14 Hours**
	Unit 4: OO Design Patterns & Anti-Patterns Structural Patterns–Adapter, Façade, Proxy and Fly weight Behavioral Patterns–Chain of Responsibilities, Command, Interpreter, Iterator. Anti-patterns–Introduction and classification, Project Management, Architecture and Development anti-patterns (1 anti-patterns of each type) **14 Hours**
Laboratory	● Lab assignment on Use case diagram. ● Lab assignment on Class diagram. ● Lab assignment on Activity and State diagrams. ● Lab assignment on Java fundamentals. ● Lab assignment on Java Advanced features (Inheritance, Composition, etc.). ● Self-Learning Assignment on Java Serialization and Multithreading ● Hands-on Assignment on MVC Framework. ● Assignment on Design Patterns. ● Mini Project using MVC Framework and incorporating all learning of the course.
TextBook(s):	1. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", by Craig Larman, 3rd Edition, Pearson 2015. 2. "Software Architecture Patterns"by Mark Richards,2nd Edition, OReilly 2022.
Reference Book(s):	1. "Object-Oriented Modelling and Design with UML",Michael RBlaha and James RRumbaugh, 2nd Edition,Pearson 2007. 2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, RalphJohnson, and John Vlissides, 1st Edition, Pearson 2015. 3. "Javathe Complete Reference", Herbert Schildt, McGraw-Hill,11th Edition,2018.
Course Outcome	● Construct static models, use cases, and class models, followed by analyzing the dynamics of the system using activity, sequence, state and process models. Depict the architecture of a software system by using component and deployment models. ● Use the concepts of classes and objects of object-oriented programming in Java to model a complex system. ● Use GRASP and SOLID principles in the design of software application and apply Creational software design patterns for variety of application scenarios. ● Apply Structural and Behavioural software design patterns for variety of application scenarios. Understand Anti- patterns.

Course Code	UE23CS341B	Course Title	Compiler Design				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Core				
AI Tools /Tools/Languages	Lex/flex and YACC/Bison. Compiler Explorer + AI Explanation	Desirable Knowledge	Data Structures and its Application, Automata Formal Languages and Logic				
Prelude	Language design and implementation is an active topic in programming and will likely always be. How we program and the tools we use, changes constantly. We try new ideas and come up with better or alternative approaches frequently. Any language that doesn't continue to adapt will fall into disuse, and any tool chain that remains stagnant will be forgotten. Hence knowledge of compilers in order to tweak these changes in the language design is a must for a Computer Science Engineer. This course requires the student to have a desirable knowledge of Data Structures and its Applications and Automata Formal Languages and Logic.						
Course Objectives:	• Introduce the major concept areas of language translation and compiler design • Develop a greater understanding of the issues involved in programming language design and implementation. • Provide practical programming skills necessary for constructing a compiler. Develop an awareness of the function and complexity of modern compilers. • Provide an understanding of the importance and techniques of optimizing code from the Compiler's perspective.						
Course Contents	**Unit 1: Compilers: Introduction, Lexical Analysis, Top-down Parsers** The Language Processing System, The Phases of a Compiler, The Grouping of Phases into Passes. Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Design of a Lexical Analyzer Generator. The role of the parser, Syntax Error Handling, and Error-Recovery Strategies. Top-down parsing: LL (1) Parser. **14 Hours** **Unit 2: Syntax Analysis: Bottom-up Parsers, Syntax-Directed Translation** Bottom-up parsing: Shift-Reduce Parsing, viable prefixes, CLR, LALR, Error recovery for Bottom-up parsers Syntax-directed definitions, Evaluation orders for SDD's: S-attributed SDD, L-attributed SDD. **14 Hours** **Unit 3: Implementation of Syntax-Directed Translation Schemes and Intermediate Code** Generation Applications of Syntax-Directed Translation-SDD for Syntax Trees, Expressions, Basic Types and flow control statements, Type Checking, Syntax-directed Translation Schemes – Parser. Stack Implementation of Postfix SDT's, SDT's with actions inside Productions, SDT's for L- Attributed Definitions. Implementing L-Attributed SDD's: Bottom-Up Parsing. Variants of Syntax Trees–Directed Acyclic Graphs for Expressions, Three-Address Code–Addresses and Instructions, Quadruples, Triples, Indirect Triples, SSA Form, Control Flow Graph. **14 Hours**						

	Unit 4: Machine Independent Code Optimization, Code Generation and RunTime Environment Machine Independent Optimization: Different Optimizations, Optimization of Basic Blocks. Data Flow Analysis: Live-variable analysis, Next-use algorithm. Storage Organization, Different Allocation Strategies, Stack Allocation of space, Access to Non-local Data on the stack. Code Generation: Issues in the design of a code generator, the target language, addresses in the target code, static allocation, stack allocation, run-time addresses for names. A Simple Code generator - The Code generation algorithm, Register allocation problem (Graph coloring). **14 Hours**
TextBook(s):	1. "Compilers–Principles, Techniques and Tools", Alfred V.Aho, Monica S. Lam, Ravi Sethi, Jeffery D. Ullman, 2nd Edition, Pearson Education, 2009.
Reference Book(s):	1. "Modern Compiler Design",Dick Grune, Keesvan Reeuwijk, Henri E.Bal,CerielJ.H. Jacobs, Koen Langendoen, 2nd Edition, 2012. 2. AW Appel, J Palsberg, Modern Compiler Implementation in Java, Cambridge University Press, 2002.
Course Outcome	● Use the knowledge of patterns, tokens, and regex for solving the problems in the field of data mining. ● Analyze and design the semantic behavior of a compiler. Choose the appropriate compiler internal representation for different kinds of compiler tasks. ● Translate a source-level language into a low-level compiler internal representation. ● Optimize the performance of a program in terms of speed and space using new code optimization techniques.

<table>
<tr><td>Course Code</td><td>UE23AM342BA1</td><td>Course Title</td><td colspan="5">Interdisciplinary Deep Learning on Graph</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>6</td><td>Type of Course</td><td colspan="5">Elective - III</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Pytorch Geometric for Graph Neural Networks.</td><td>Desirable Knowledge</td><td colspan="5">Machine Learning,
Advanced Foundations for Machine Learning</td></tr>
<tr><td>Prelude</td><td colspan="7">This advanced deep learning course deals with graph-based deep learning approaches for non-Euclidean data. Graphs are ubiquitous today –in social graphs, large-scale biological systems, road networks, polypharmacy, and user-item relations. Hence, this course aims to equip the students with modern graph-based techniques in interdisciplinary applications. This course starts with shallow representation learning techniques and then follows up with Deep Learning-based graph representation learning algorithms, including special cases such as Deep graph generative models, Heterogeneous GNN, Dynamic GNN, and Hypergraph GNN. The course then deals with GNN-based interdisciplinary applications in detail.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Learn shallow representation learning mechanisms.
● Learn deep neural methods of representation learning on graphs.
● Learn GNN-based interdisciplinary applications for different domains.
● Learn GNN-based interdisciplinary applications for more domains.</td></tr>
<tr><td>Course Contents:</td><td colspan="7">Unit 1: Neural Representation Learning on Graph
Machine learning on Graph – Euclidean vs. non-Euclidean data, Graphs of different types, Downstream learning tasks on graph. Random walk-based embedding–DeepWalk, Node2Vec, Encoder-Decoder perspective, Limitations of shallow embedding. Knowledge Graph-types of Knowledge Graphs, Industrial Applications. Different relation patterns. Knowledge Graph Embedding - Triple, Knowledge Graph Completion Task, Translational approaches such as TransE, TransR, KG Embedding model - Scoring function, Loss function, Optimization algorithm, Negative samples generation. Vanilla Graph Neural Network-Neural message passing framework. Generalized neighborhood aggregation. Generalized update methods.
14 Hours

Unit 2: Graph Neural Network
Graph Convolution Networks (GCN) and Multi-relational GCN, GraphSAGE, Graph Attention Networks (GAT), Graph Isomorphism Network(GIN), Graph Transformer. Generating graphs with GNN- Graph autoencoder and Graph variational autoencoder (VGAE).Dynamic graphs – spatial-temporal GNN. Heterogeneous GNN, Hypergraph GNN. Efficiency issues in GNN Modelling - Graph pooling. Applications and loss functions. GNN layer optimization, Stacking GNN layers, GNN modeling pipeline with different prediction heads, Graph augmentation, and Setting up graph datasets for a learning task.GNN Interpretability.
14 Hours

Unit 3: Interdisciplinary Applications with Graph Neural Network -1
Web data mining - Social Influence Prediction, Recommender System. Urban data mining - traffic prediction and anomaly detection. Cybersecurity data mining – Malicious account detection, Fake news detection. Natural language processing–Question Answering, Graph to Sequence learning. Software Engineering – Software Mining and program analysis.
14 Hours

Unit 4: Interdisciplinary Applications with Graph Neural Network -2
Computer vision – Visual QA, Image Classification, Skeleton-based action recognition. Biochemistry and Healthcare – Protein-protein interaction, Disease prediction, Drug discovery. Robust Graph Neural Network - Graph Adversarial Attack, Graph Adversarial Defenses. Graph Neural Network meets Large Language Models.
14 Hours</td></tr>
<tr><td>Text Book(s):</td><td colspan="7">1. “Deep Learning on graphs”, Yao Ma and Jiliang Tang, Cambridge University Press, 2021
2: “Graph Neural Networks: Foundations, Frontiers and Applications”, Lingfei Wu, Peng Cui, et al., Springer, 2023</td></tr>
</table>

Reference Book(s):	1. Lecture Videos of Stanford/Fall 2024 Course CS224W: Machine Learning With Graphs 2: "Graph Representation Learning", William L Hamilton, Morgan and Clay pool Publishers, 2020.
Course Outcome	● Downstream machine learning tasks on graphs using shallow embedding. ● Solve a downstream machine-learning task using a graph neural network. ● Appreciate the architectures of interdisciplinary applications using GNN. ● Do a meaningful course project using Graph Neural Network-based models.

<table>
<tr><td>Course Code</td><td>UE23AM342BA2</td><td>Course Title</td><td colspan="5">Large Language Models and Their Applications</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>6</td><td>Type of Course</td><td colspan="5">Elective - III</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Pytorch Geometric for Graph Neural Networks.</td><td>Desirable Knowledge</td><td colspan="5">Machine Learning.</td></tr>
<tr><td>Prelude</td><td colspan="7">Large language models (LLMs) are neural networks that can generate natural language text based on a given input, such as a word, a phrase, or a prompt. They are trained on massive amounts of text data, such as web pages, books, news articles, etc. Some examples of LLMs are GPT-3, BERT, and T5. LLMs have shown remarkable capabilities in various natural language processing tasks and beyond, such as translation, summarization, question answering, dialogue, image captioning, code generation, etc. This course covers the fundamental concepts, architectures, techniques, and applications of LLMs.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Understand the principles and challenges of LLMs.
● Learn the main architectures and components of LLMs, such as transformers, attention, self-attention, etc.
● Explore the methods and strategies for training, fine-tuning, and evaluating LLMs.
● Apply LLMs to various natural language understanding and generation tasks.
● Investigate the recent innovations and trends in LLM research.</td></tr>
<tr><td>Course Contents:</td><td colspan="7">Unit 1: LLM Architectures and Components
What are LLMs and why are they important? History and evolution of LLMs. Tokenization: Byte pair encoding, word piece encoding , sentence piece encoding. Embedding:TF-IDF,word2vec, SkipGram, CBOW and GLOVE
Transformers and self-attention, Encoder-decoder models and attention. Pre-training, Vision transformers. BART,BERT,T5 RoBERTa transformers.
14 Hours
Unit 2: NLP and Prompt Engineering
Natural language understanding tasks, such as classification, sentiment analysis, named entity recognition, translation, summarisation, paraphrasing, Prompt structure. Open and close prompts, Soft and Hard prompts. Prompt Engineering techniques, Zero/Few shot examples, Chain/Tree/Graph of Thought, Reasoning and Act. .
14 Hours
Unit 3: RAG
Naive Retrieval Augmented Generation: Chunking, BM25 sparse retrieval and ColBERT dense retrieval, Indexing: FLAT, IVF, HNSW and PQ. Vector stores: pinecone, milvus, chroma DB etc, Mixture-of-experts and retrieval-based LLMs. Advanced RAG: Multihop, Query expansion, Cross Encoders, hybrid search.
14 Hours
Unit 4: Fine Tuning, Agentic AI & Recent Innovations and Trends
Full fine tuning, PEFT: LORA, QLORA, Prompt tuning. Multi-task and multi-modal LLMs, MAMBA network: state space model, Agentic Workflows , Design Patterns in agents, Ethics and security of LLMs.
14 Hours</td></tr>
<tr><td>Text Book(s):</td><td colspan="7">1. Large Language Models: An Introduction By Oswald Campesato (Packt, Oct 2024).</td></tr>
<tr><td>Reference Book(s):</td><td colspan="7">1. Hands-On Large Language Models By Jay Alammar & Maarten Grootendorst (O'Reilly, Oct 2024).
2. How Large Language Models Work By Edward Raff, Drew Farris & Stella Biderman (Manning, Summer 2025).
3. Large Language Model-Based Solutions By Shreyas Subramanian (Wiley, May 2024).
4. Engineering Large Language Models: A Practical Guide from Design to Deployment By Landon Scott (Dec 2024)</td></tr>
</table>

	5. Large Language Models Projects: Apply and Implement Strategies for Large Language Models By Pere Martra (Apress, Sep 2024).
Course Outcome	● Explain the basic concepts and terminology of LLMs. ● Compare and contrast different LLM architectures and variants. ● Implement and adapt LLMs using popular frameworks, such as HuggingFace, TensorFlow, PyTorch, etc. ● Solve various natural language processing problems using LLMs. ● Critically analyze the strengths and limitations of LLMs.

Course Code	UE23AM342BA3	Course Title	Explainable AI				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – III				
AI Tools /Tools/Languages	Python Tools and Libraries.	Desirable Knowledge	Machine Learning.				
Prelude	Understanding the logic and reasoning behind machine learning predictions is essential in making business decisions, healthcare predictions, automotive and Aerospace AI based applications. Explainable AI(XAI) is a field of study in AI which includes set of methodologies that give the insight of making complex machine learning and deep learning model understandable and Interpretable. Students obtain a basic proficiency in interpreting and explaining the decisions of ML and AI systems, in a transparent and understandable manner to humans. The course will cover various XAI techniques and algorithms, including rule-based models, feature importance analysis, model-agnostic approaches, and post-hoc explanations.						
Course Objectives:	• To understand the concept of Explainable AI, its scope, and its impact on various application domains • To learn about XAI types and techniques • To learn the concept of interpretable models, Global and Local model agnostic Methods • To learn about various approaches for Neural Network Interpretation.						
Course Contents:	**Unit 1: Introduction to interpretability and Interpretable Models** Interoperability: Importance, Taxonomy, Scope, Evaluations of Interpretability, Explanations, Interpretable Models: Linear, Logistic, GLM and GAM, , Applying LIME Techniques, Decision Tree and Rule Based Methods: Anchors counter factual Explanations, Structured Data Explainers **14 Hours** **Unit 2: Global and Local model agnostic Methods** Global: Introduction, Partial Dependence Plot(PDP),Accumulated Local Effects(ALE) plot, Feature Interactions, Functional Decomposition, Permutation Feature Importance, Global Surrogate. Local: Individual Conditional Expectation, Local Surrogate-LIME, Counterfactual Explanations, Scoped Rules-Anchors, Shapley, SHAP (SHapley Additive exPlanations). **14 Hours** **Unit 3: Neural Network Interpretation** Learned Features, Pixel Attribution, Detecting Concepts, Adversarial Examples, Influential Instances, Image Explainers: LIME for Images, Grad-CAM. **14 Hours** **Unit 4: Interpretability for unsupervised Learning and NLP** Clustering: Pre-model Explainability, Unsupervised Learning for Clustering, Summary of Unstructured Data, Explainers K-means Clustering Explainer, NLP: Attention and Concept Based Explanations, Explaining Sentiment Analysis, Layer Integrated Gradients, Using Lime for Sentimental Analysis. **14 Hours**						
Text Book(s):	1. Interpretable ML Book Christoph Molnar Lethamand Rudin,2015. 2. Interpretable Classifiers Using Rules and Bayesian Analysis Lakkaraju et. al., 2016. 3. Interpretable Decision Sets Wachteret.al.,2018. 4. Counterfactual Explanations Without Opening the BlackBox Karimiet.al.,2020. 5. Algorithmic Recourse: From Counter factual Explanations to Interventions Mullenbachet.al.,2018. 6. Explainable Prediction of Medical Codes from Clinical Text Jain and Wallace, 2019. 7. Attention is not Explanation Covertet.al.,2021, Explainingby Removing: A Unified Framework for Model Explanation.						

Reference links(s):	https://github.com/jphall663/awesome-... DeepFindr: Explainable AI explained! | #1 Introd... Kaggle notebook: https://www.kaggle.com/code/parulpand...Interpretable MLbook: https://christophm.github.io/interpre... InterpretML: https://interpret.ml/ ShapLibrary:https://github.com/slundberg/shap Partial Dependence Plots: •Partial Dependence Plots (Opening the... Interpret ml. Includes explainability.
Course Outcome	● Ability to interpret models' results using XAI techniques. ● Ability to Understand Global vs Local Explanations and their applications. ● Ability to incorporate and build ML and DL models using widely used XAI techniques such as LIME, SHAP, Partial Dependence plot. ● Ability to select and correctly apply the interpretation method that is most suitable for machine learning models. ● Ability to analyse the explanations and find their limitations and biases.

Course Code	UE23AM342BA4	Course Title	Deep Learning for Life Sciences				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – III				
AI Tools /Tools/Languages	Pytorch, SkLearn, Keras, Tensorflow Google cloud Vertex AI (end to-end ML platform).	Desirable Knowledge	Machine Learning, Advanced Foundations for Machine Learning				
Prelude	This is an interdisciplinary introductory course for students aspiring for a future career in Life Science. Today, data science and life science have converged as automation and robotics in life science research generate enormous data, requiring data science to discover not-so-obvious relations. Machine Learning, specifically Deep Learning, has emerged as the technique of choice for finding relationships in a large volume of data in this domain. This course will introduce deep learning techniques in computer-aided diagnostics, microscopy, molecular chemistry, proteins, genomics, and drug discovery.						
Course Objectives:	• Learn deep learning applications in computer-aided medical diagnostics. • Learn deep learning applications in Molecular Chemistry. • Learn deep learning applications in proteins Genomics. • Learn applications of generative models in drug discovery.						
Course Contents:	**Unit 1: Deep Learning for Computer-Aided Diagnostics and Microscopy** Introduction to Computer-aided Diagnostics: Electronic Health Record (EHR), Deep Radiology – X-ray scan, CT Scan, MRI scan, and Histology, Learning models as Therapeutics. Case Study - Diabetic Retinopathy. Introduction to Microscopy: Modern optical Microscopy, Diffraction Limit, Super-resolution Microscopy, Fluorescence Microscopy, Preparing biological samples for Microscopy. Case Study – Cell counting from a sample, cell segmentation using U-net. **14 Hours** **Unit 2: Deep Learning in Cheminformatics** Introduction to Molecular chemistry: Molecular bonds, Molecular graphs, Molecular Conformations, Chirality of molecules. Featurizing a molecule for a machine learning task, RDKIT, SMILES, Extended connectivity fingerprints (ECFP), SMARTS strings, and Molecular descriptors. MoleculeNet dataset.Graph Convolution Network (GCN) - a graph representation learning framework. Case study – Training a GCN model to predict solubility. **14 Hours** **Unit 3: Deep Learning in Biophysical systems and Genomics** Introduction to Biophysical systems: Protein structure- protein sequence, predicting protein structure – homology modelling vs. physical modelling. protein binding. Biophysical featurization of protein – grid featurization. Case study - Featurization using PDBBind dataset, building a neural network for protein-lig and binding. Introduction to Genomics: Genetics vs. Genomics, DNA, RNA, and Protein. Transcription factor binding, RNA Interference. Case Study – A convolutional model for Transcription Factor binding. **14 Hours** **Unit 4: Deep Generative Models for Drug Discovery and the Interpretability of Deep Models** Introduction to Drug Discovery: A search problem in the intersection of chemical search space and biological search space. Virtual Screening to optimize Drug discovery –structure based vs. Ligand based Virtual screening. De Novo design by a Generative model. Case study – using VAE to create molecules. Interpretation of Deep Models: Saliency Mapping approach. Case study – An interpretable deep model using saliency mapping. **14 Hours**						
Reference Book(s):	1. Deep Learning for Genomics, Upendra Kumar Devisetty, Packt, 2022. 2. Bioinformatics with Python Cookbook, Tiago Antao, Third Edition, Packt, 2022. 3. DeepChem : https://deepchem.io/						
Course Outcome	• Should be able to apply machine learning in Medicine using health record datasets. • Should be able to use Deep Learning for applications in chem. informatics. • Should be able to use Deep Learning for applications in proteins and genomics. • Should be able to use generative models for de novo design.						

Course Code	UE23AM342BA5	Course Title	Intelligent Systems with Knowledge Graphs				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – III				
AI Tools /Tools/Languages	Python, Neo4J, Networkx, NLP Libraries.	Desirable Knowledge	Machine Learning, Advanced Foundations for Machine Learning				
Prelude	Intelligence is the ability to acquire and apply knowledge. Developing intelligent systems incorporating contextual knowledge is important in situations impacting human life. Knowledge Graphs represent a core abstraction for incorporating contextualized knowledge into intelligent systems. Life Science and Healthcare are two domains where knowledge graphs are pervasive. This course deals with knowledge acquisition, representation of knowledge in a graph format, and intelligent reasoning systems. This course equips students aspiring to pursue a career as a data scientist in healthcare and life science.						
Course Objectives:	• Introduce the students to concepts of Intelligent systems, Knowledge graphs, and building from a text corpus. • Introduce the students to techniques of knowledge graph-building from text data. • Introduce the students to the knowledge graph representation learning. • Introduce the students to Graph RAG using knowledge graphs.						
Course Contents:	**Unit 1: Introduction to Intelligent Systems and Knowledge Graphs, Building Knowledge Graph Using Natural Language Processing** Introduction to Intelligent Systems: What is an intelligent system? Autonomous vs. Advisor systems, characteristics of Intelligent Systems. Building an Intelligent System - knowledge acquisition, knowledge representation, Reasoning, Role of knowledge Graphs. Applications of knowledge graphs in Healthcare and Life Sciences – paper discussion. Introduction to knowledge graphs: Resource Description Framework (RDF) and Labeled Property Graph (LPG).Constructing a Knowledge Graph using NLP techniques: named entity recognition, coreference resolution, Relation extraction, External data enrichment, and Implementation of an information extraction pipeline. Life Science and Healthcare specific text embedding – BioBERT, PubMedBERT, ClinicalBERT. Constructing a Semantic Network using NLP techniques: Unsupervised keyword extraction, keyword co-occurrence graphs, clustering keywords, and topic identification. LLM trained on Biological and Healthcare data - LLaMA-Med, Vicuna-Med, GatorTron, BioGPT , Med-PaLM. KG Building by LLM - Traditional NLP vs. LLM. **14 Hours** **Unit 2: Knowledge Graphs using Neo4J, Creating Knowledge Graphs from Ontologies** Introduction to Neo4J – Cypher, APOC, and GDS plugins. Knowledge graph building by ontology ingestion with Neo4J, querying the data, and reasoning over the knowledge graph. Complex knowledge graph examples from the Biomedical Domain – multi-omic, pharmaceutical, and clinical applications. Case Study -Building a Knowledge Graph to capture microRNA (mRNA) and disease association. **14 Hours** **Unit 3: Deriving Implicit Information on Knowledge Graphs – Machine Learning on Knowledge Graphs** Machine Learning tasks on KG – node classification, link prediction, graph classification, and graph clustering. Translational embedding algorithms - TransE, TransR, DistMult. Knowledge graph completion task. Random walk-based algorithm - Node2Vec, Encoder-decoder perspective. Graph Convolution network–message passing framework, generalized aggregation and update.Suggested Case studies–friend recommendation. Knowledge Graph Mining - protein-protein interaction, protein-drug interaction, drug application prediction. **14 Hours** **Unit 4: Knowledge Graph and Large Language Model - Graph Retrieval Augmented Generation** Why Graph RAG – Limitations of LLM, Overcoming limitations by RAG, Knowledge Graph as the external data source. RAG overview – Retriever, Generator, RAG using vector similarity search. Introduction to Agentic RAG– Query rewriter, Retriever agent, Retriever router, Answer critics. Knowledge Graph vs. Vector Database - Strengths and Weaknesses. Hybrid approach with Knowledge Graph and Vector Database. RAG application evaluation. **14 Hours**						

Text Book(s):	1. "Graph-Powered Machine Learning: Use Graphs and Deep Learning to Solve Complex Problems", Alessandro Negro, Manning, 2021. 2. "Knowledge Graphs: Fundamentals, Techniques, and Applications", Dieter Fensel, Umutcan Şimşek, Kevin Angele, et al., Publisher: Springer (2020). 3. "Knowledge Graphs: Fundamentals, Techniques, and Applications", Mayank Kejriwal, Craig A. Knoblock, and Pedro Szekely, MIT Press, 2021.
Reference Book(s):	1. "Building Knowledge Graphs: A Practitioner's Guide", Jesus Barrasa & Jim Webber, OREILLY, 2025. 2. "Building AI Agents with LLMs, RAG, and Knowledge Graphs: A Practical Guide to Autonomous and Modern AI Agents", Salvatore Raieli , Gabriele Luculano, Packt, 2023. 3. "Healthcare Data Analytics", Chandan K Reddy and Charu C. Agarwal, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2020.
Suggested Papers:	1. Knowledge Graphs for the Life Sciences: Recent Developments, Challenges and Opportunities- Jiaoyan Chen et al. , 2023 , arXiv. 2. A Review on Knowledge Graphs for Healthcare: Resources, Applications, and Promises, Hejie Cui et al., arXiv, 2025. 3. Patient-Centric Knowledge Graphs: A Survey of Current Methods, Challenges, and Applications, Hasan S. et al. , arXiv, 2024. 4. Cost Effective Digital Prescriptions using Pharmaceutical Knowledge Graphs, Aryan Rathod et al., Asian Journal Of Research in Computer Science, 2023. 5. Knowledge Graphs in Pharma covigilance: A Scoping Review, Manfred Hauben et al. Clinical Therapeutics (Elsevier), 2024. 6. Knowledge-Guided Learning Methods for Integrative Analysis of Multi-Omics Data, Wenrui Li et al. , Computational and Structural Biotechnology Journal (Elsevier), 2024.
Course Outcome	● Ability to build a Knowledge Graph from a text corpus. ● Ability to build a Knowledge Graph using Neo4J from domain-specific ontologies. ● Ability to find implicit information from KG using the shallow and deep Knowledge Graph embeddings.

Course Code	UE23CS342BA5	Course Title	BlockChain				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - III				
AI Tools /Tools/Languages	Solidity, Ganache, Meta mask. RemixID, Ganache tool.	Desirable Knowledge	Data Structures for coding purpose				

Prelude

This course offers a comprehensive understanding of Blockchain technology, covering its foundational concepts, data integrity, cryptographic techniques, and network structures. Students will explore key blockchain models, consensus algorithms, smart contracts, and second-generation applications like DApps and DAOs. Through hands-on demonstrations, including wallet creation and transactions, learners gain practical skills in working with platforms like Etherscan and Hyperledger Fabric. The course also addresses security challenges and real-world use cases across industries, preparing students for research and professional applications in blockchain.

Course Objectives:

- To provide foundational knowledge of blockchain technology, including its architecture, data integrity principles, and key components such as nodes, tokens, and ledgers.
- To explore various consensus algorithms and their roles in achieving distributed agreement in blockchain networks.
- To introduce smart contracts and decentralized applications (DApps) using programming tools like Solidity and platforms such as Ethereum.
- To examine blockchain security, vulnerabilities, and real-world use cases in sectors like finance, supply chain, and public administration.

Course Contents

Unit 1: Introduction and Dataflow and Integrity

Key Blockchain Concepts, Nodes, Cryptocurrency, tokens, Public Ledger, Peer to peer Network, Types of Blockchain, Permissioned Blockchain model, Permission-less Blockchain model, Demonstration of Blockchain Construction steps, Demonstration of ether scan and Goerli/Sepolia etherscan. Types of cryptography, Digital signatures, Hash functions, SHA-256, Patterns of hashing, Hash Pointer, Merkle tree. Case study: Bitcoin Blockchain Network, Creation of metamask wallets and performing transaction.

14 Hours

Unit 2:The structure of the network: consensus algorithm

Case study: Bitcoin Blockchain Network, Creation of metamask wallets and performing transaction. Introduction to distributed consensus: What, why, Challenges, Proof of Work, Proof of Stake, Delegated Proof of Stake, Proof of Authority, Proof of Elapsed Time, Proof of Scope, Proof of Space, Proof of Burn, RAFT, PAXOS, Byzantine Fault Tolerance System, PBFT. Smart contracts: origins and how they function, Creating and deploying smart contracts.

14 Hours

Unit 3: Second generation applications of Blockchain technology

Solidity- Variable, Functions, Arrays/Strings, enum/structs, mapping, function visibility - public, private, internal, external, modifiers, view, pure, fallback, overloading, in-built mathematical and cryptography functions, Withdrawal pattern, Restricted Access, ether units.

Decentralized applications, Dapps construction, Decentralized Autonomous Organizations (DAOs)-Need, Principal agent dilemma, components, The DAO Story, Legality of DAPPs and DAOs.

14 Hours

Unit 4:Blockchain Security and use cases

Hyperledger Fabric: Blockchain-as-a-service (BaaS), Architecture and core components, Hyperledger fabric model, Creation of a simple DAPP. Blockchain vulnerabilities, Smart contract vulnerabilities,

		Blockchain based DNS security platform, deploying blockchain based DDOS protection. Use cases: Public Sector, Finance, Supply Chain. Research Aspects in Blockchain. **14 Hours**
	Text Book(s):	1. Introduction to Blockchain Technology, Tiana Laurence,1st edition, Van Haren Publishing,2019. 2. Blockchain Technology from Theory to Practice, SudeepTanwar,1st edition, Springer,2022.
	Reference Book(s):	1. Hands-On Cybersecurity with Blockchain: Implement DDoS protection, PKI-based identity, 2FA, and DNS security using Blockchain, Rajneesh Gupta, 1st edition, 2018. 2. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Narayanan, Bonneau, Felten, Miller and Goldfeder, Princeton University Press, 2016. 3.Drescher,Daniel.Blockchain Basics: A Non-Technical Introduction in 25 Steps. Germany:Apress,2017.
	Course Outcome	● Understand and explain the structure, types, and functioning of blockchain systems and cryptographic techniques used for data integrity. ● Analyze and compare different consensus mechanisms and their applicability in distributed networks. ● Design, code, and deploy smart contracts and DApps using Ethereum-based tools like Metamask and Solidity. ● Evaluate blockchain's impact and applicability in various industries, along with identifying and mitigating potential security vulnerabilities.

Course Code	UE23CS342BA6	Course Title	Digital Forensics and Incident Response				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – III				
AI Tools /Tools/Languages	Open-source tools on Forensics. Cellebrite, Deepfake Detection Tools	Desirable Knowledge	Machine Learning				

Prelude

Digital Forensics is the core set of principles and processes necessary to produce usable digital evidence and uncover critical intelligence. This course provides a deep understanding of techniques to gather, protect, analyze, and report digital confirmations through a combination of theoretical foundation and practical exercises. The content is organized into four units, each with a focused objective and corresponding outcomes

Course Objectives:

- Introduce the history, evolution, basic principles, challenges, and incident response techniques in digital forensics.
- Develop expertise in Windows file systems, including file structure analysis, registry examination, and related forensic recovery tools.
- Build foundational skills in Linux and macOS systems while integrating network forensics and browser-based analysis.
- Equip students with specialized knowledge in email forensics, mobile device investigations, anti-forensics techniques, and professional report writing.

Course Contents

Unit 1: Introduction to Digital Forensics and Incident Response

Foundations of Digital Forensics: Introduction to forensic science and digital forensics, Goals, categories, forensic readiness, life cycle, and challenges, Role of the examiner; types of investigations; importance of international standards

Incident Response: Attack lifecycle and introduction to incident response, Building an Incident Response Team and drafting an Incident Response Plan (IRP), Forensic tools and methodologies for initial response, Investigations Triad

Evidence Acquisition and Analysis: Types of digital evidence (HDD, SSD, RAM), Data recovery, allocated vs. unallocated space, and persistence (page/swap files),Major Concepts, Essential Technical Concepts Initial responder tasks: search and seizure, documentation, packaging, and transporting evidence

Labs & Activities:

- **Activity1:**Understanding and familiarization with Digital Forensic
- **Lab1:** Acquiring digital evidence (HDD/SSD and RAM) with open-source tools.
- **Lab2:**Analysing digital evidence(HDD/SSD and RAM)with open-source tools. **14 Hours**

Unit 2: Windows Forensics–File systems and Registry Analysis

Windows File Systems: Overview of FAT and NTFS file systems, Examination of file allocation structures and metadata, Techniques for timeline analysis and file recovery

Windows Registry Analysis: Key artifacts: Deleted data, hibernation file (Hiberfile.sys), print spooling, recycle bin, metadata, thumbnail cache, Most Recently Used (MRU) lists, restore points, and shadow copies, Methods for recovering deleted registry keys and performing registry forensics.

Labs & Activities:

- **Activity2:** Case study using tools like Autopsy for practical Windows forensics.
- **Lab3:** Data recovery and forensic analysis using EXIFTool.

	• **Lab4:** Examination of the Windows Registry with REGEDIT and open-source forensic tools. **14 Hours** **Unit 3: Linux, macOS, Network, and Browser Forensics** Linux Forensics: Linux file systems (Ext2/Ext3), structure layers (file, metadata, data unit), and artifacts, Concepts such as deleted data recovery, logical volume management, boot processes, and principles of file carving macOS Forensics: Exploration of OS X file system artifacts, including HFS+ structures and OS X system logs Network Forensics: Fundamentals of networking and types of networks, Techniques for securing network infrastructures, analyzing network evidence, and applying forensic science to networks, Tools and processes for the seizure and analysis of networking devices Browser Forensics: Overview of web browsers (IE, Firefox, Chrome), Forensic analysis of browser artifacts and investigation tools. **Labs & Activities:** • **Activity 3:** Practical session on understanding Linux artifacts using both CLI and GUI tools. • **Lab 5:** Analysis of PCAP files and network forensic investigations. • **Lab 6:** Browser forensic investigations using open-source tools. **14 Hours** **Unit 4: Email, Mobile Forensics, Anti-Forensics, and Report Writing** **Email Forensics:** Examination of email communications, protocols, and header analysis, Techniques for recovering and analyzing email evidence. **Mobile Device Forensics:** Architecture, boot processes, and the forensic significance of cellular network data, Procedures related to SIM filesystems, device locks, rooting, and jail breaking Methods for evidence acquisition and challenges associated with mobile devices. **Anti-Forensics:** Overview and classification of anti-forensics techniques, Practices including data wiping, shredding, obfuscation, encryption, and data hiding. **Report Writing:** Preparing and structuring forensic reports best practices for report design, documentation, legal acceptance, and presenting digital evidence. **Labs & Activities:** • **Lab 7:** Email forensics using open-source tools. • **Lab 8:** Demonstration of mobile forensics techniques. **Activity 4:** Collaborative report writing exercise and peer review. **14 Hours**
TextBook(s):	1. "The basics of digital forensics: the primer for getting started in digital forensics", Sammons, J.Elsevier, 2012. 2."Digital Forensics Basics: A Practical Guide Using Windows OS",Hassan,N.A.Apress2019. 3."Practical Cyber Forensics- An Incident-Based Approach to Forensic Investigations", by Niranjan Reddy, A Press, 2019. 4:"Introductory Computer Forensics – A Hands-on practical Approach",by XiaodongLin, Springer,2018.
Reference Book(s):	1: "Guide to Computer Forensics and Investigations", Bill Nelson, Amelia Phillips and Christopher Steuart, Course Technology, Cengage Learning, 2010. 2:"Digital forensics with open-source tools",Altheide,C.,&Carvey,H.Elsevier,2011. 3:"Digital Forensics Workbook-Hands-on Activities in Digital Forensics", Michael K Robinson, CreateSpace Independent Publishing Platform, 2015.

Course Outcome	• Students will understand the fundamental principles of digital forensics, be familiar with its historical evolution, and perform initial incident response and evidence acquisition. • Students will be able to analyze Windows file systems and registry artifacts effectively, perform timeline analyses, and recover deleted data. • Students will acquire and analyze forensic data from Linux and macOS environments and demonstrate proficiency in network and browser forensics. • Students will conduct mobile device and email investigations using proper tools, recognize anti-forensics practices, and compile legally sound forensic reports.

Course Code	UE23CS342BA7	Course Title	Digital Twins				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - III				
AI Tools /Tools/Languages	C/C++/JAVA/Python using OpenGL.	Desirable Knowledge	Data Structures, Design and Analysis of Algorithms				

Prelude

Digital Twins are revolutionizing how we design, monitor, and optimize complex systems by creating dynamic virtual replicas of physical entities. This course introduces the foundational concepts, modeling techniques, and simulation strategies needed to build and analyze Digital Twins across domains such as healthcare, manufacturing, and transportation. Students will gain hands-on experience with tools like Unity, Blender, and OpenGL, while also exploring critical topicslike3Dview synthesis and cybersecurity. By the end of the course, students will be equipped to conceptualize, build, and secure intelligent Digital Twin systems.

Course Objectives:

- Develop fundamental understanding of Digital Twins and its architecture. With help of case studies provide a structured way to conceptualize any Digital Twin.
- Develop understanding on various techniques for structural twinning and 3D view Synthesis.
- To build skill in modelling-based system engineering, focusing on discrete event systems.
- Collaborate with peers on practical projects to gain hands-on experience in developing intelligent systems. Develop understanding about secure designs and privacy concerns in the context of Digital twin.

Course Contents

Unit 1: Digital Twin Essentials

The Big Picture of Digital Twins, History of the Digital Twin, Origin of the Digital Twin concept, Digital Twin and Product Life cycle Relationship, Types of Digital Twin: Discrete Vs Composite, Product versus facility, Types of Digital Twin: Simulation versus operational, Analytics versus physics-based, Characteristics of a Digital Twin, Digital Twin Architecture, Industrial Digital Twin applications, Examples of mock, functional, and executable twins, Data Driven Modelling, Physics Driven Modelling, Hybrid (Data-Physics) Modelling, Examples of Physics Driven Modelling: DT of a Propeller of a Drone, DT of a ceiling fan, Bio-Mechanical DT of human body; Examples of Data Driven Modelling: DT for RUL of a Battery, DT for traffic Mobility in Bangalore, DT of RUL of GUT; Hybrid Modelling: Heart Digital Twin, DT for Solar Array, DT for CAR using OBD., Case Study: Manufacturing, Case Study: Healthcare, Case Study: Buildings, Case Study: Transportation & Logistics.

14 Hours

Unit 2: Foundations and Techniques in 3D View Synthesis

Manual 3D modelling and mesh editing, Point cloud to mesh conversion, Poisson surface reconstruction, Ball-pivoting algorithm, Alpha shapes and Delaunay triangulation, Camera models and calibration, Epipolar geometry and depth estimation, Structure from Motion (SfM), Multi-View Stereo (MVS), Depth Image-Based Rendering (DIBR), Forward and backward image warping, Neural Radiance Fields (NeRF) fundamentals, Volume rendering with MLPs, 3D Gaussian Splatting for real-time rendering.

14 Hours

Unit 3: System Modelling and Simulation

Introduction to Simulation, System and System Environment, Discrete Event System Simulation, General Principles, The Event Scheduling/Time Advance Algorithm, Input Modelling, Estimation of Absolute Performance, Verification, Calibration and Validation.

14 Hours

	Unit 4: Digital Twin & Cyber Security. Digital twins and cybersecurity, Security Framework, Digital twins threat modelling, Common attacks on digital twins, Common attacks on digital twins, Digital twin authentication and identification challenge, IDS, IPS, Authentication Methods, Communication Channel Protection, building cyber resilience in digital twins, Privacy Framework, Lack of Privacy, and trust, Privacy by Design, Enhancing trust with block chain integration. **14 Hours**
TextBook(s):	1. Building Industrial Digital Twins by Shyam Varan Nath & Pietervan Schalkwyk, by Packt Publishing Ltd. 2. JerryBanks,JohnS.CarsonII,BarryL.Nelson,DavidM.Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010. 3. Girod, Bernd, Günther Greiner, and Heinrich Niemann, eds. *Principles of 3D image analysis and synthesis*. Vol. 556. Springer Science & Business Media, 2013.
Reference Book(s):	1. Diab,W.W.,A.Ferraro,B.Klenz,S.W.Lin,E.Liongosari,W.E.Tannous,andB.Zarkout. "Industrial IoT Artificial Intelligence Framework." (2022): 1-59. 2. El Saddik, Abdulmotaleb, ed. *Digital Twin for Healthcare: Design, Challenges, and Solutions*. Elsevier, 2022.
Course Outcome	● Conceptualize a Digital Twin and build its characteristics and input output parameters. ● Use either manual or algorithmic techniques to model the structural twin of the DT. ● Translate the conceptualized Digital twin application into a Simulation model – a Time advanced discrete event system simulation. ● Collaborate effectively with peers on practical projects and demonstrate the ability to apply Digital Twin and Extended Reality solutions to real- world problems, such as improving operational efficiency, enhancing security, and reducing costs.

Course Code	UE23CS342BA8	Course Title	Cloud Security				
Category	Elective	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - III				
AI Tools /Tools/Languages	C/C++/JAVA/ Python using OpenGL.	Desirable Knowledge	-				
Prelude	This course introduces fundamental cloud security concepts, covering threats, access management, encryption, compliance, and incident response. Through hands-on labs in AWS, students will learn to design and implement secure cloud architectures while integrating DevSecOps practices.						
Course Objectives:	1. Introduce cloud computing models, security challenges, and regulatory frameworks relevant to cloud environments. 2. Develop understanding of secure cloud architecture, identity management, and data protection across hybrid and multi-cloud setups. 3. Apply secure software development practices, including DevSecOps, container security, and cloud configuration management. 4. Equip learners with skills for cloud threat detection, incident response, and disaster recovery planning.						
Course Contents	**Unit 1: Cloud Security Intro and Access Management** Introduction to Cloud Computing & Security Challenges: Cloud Computing Models(IaaS,PaaS,SaaS), Deployment Models(Public,Private,Hybrid,Multi-Cloud),SecurityChallengesinCloudEnvironments; Cloud Security Frameworks & Compliance: Shared Responsibility Model, Key Regulations: GDPR, SOC 2, PCI DSS, HIPAA, Cloud Security Standards: ISO 27017, NIST 800-144; Cloud Threats & Risk Assessment: OWASP Cloud Threats, Risk Management Frameworks(FAIR, NIST); Identity & Access Management (IAM) in Cloud: Role-Based AccessControl (RBAC) & Attribute-Based Access Control (ABAC), Identity Federation & Single Sign-On (SSO), Multi-Factor Authentication (MFA) & Zero Trust Security, IAM in multicloud. **14 Hours** **Unit 2: Cloud Security Architecture & Access Control** Secure Cloud Networking & Data Protection: Virtual Private Cloud (VPC) & Software-Defined Perimeters, Encryption Techniques: Data at Rest, Data in Transit, Data in Use, Cloud Key Management Systems (AWS KMS); Container & Serverless Security: Securing Docker & Kubernetes, Serverless Security Best Practices (AWS Lambda), Container Runtime Security & Image Scanning, Multi Cloud Security Architecture. **14 Hours** **Unit 3: Secure Software Development in Cloud** Secure Software Development in Cloud: DevSecOps Pipelines & CI/CD Security; Cloud Configuration Management: Configuration drift detection across clouds, Cloud Security Posture Management (CSPM), Infrastructure as Code (IaC) & Security Automation, Supply Chain Security & Code Vulnerability Scanning; CloudApplicationSecurity:SecureAPIDevelopment&AuthenticationMethods,WebApplicationFirewalls (WAF) & API Gateways, Security Best Practices for SaaS Applications, Challenges of Multi cloud applications. **14 Hours** **Unit 4: Cloud Threats, Incident Response & Security Operations** Cloud Security Monitoring & Threat Detection: SIEM in Cloud, Intrusion Detection & Prevention in Cloud Environments, Security Logging & Monitoring(CloudTrail, CloudWatch);Incident Response in Cloud:Root Cause Analysis of Cloud Breaches, Real-World Cloud/Multi Cloud Attack CaseStudies; Business Continuity & Disaster Recovery in Cloud. **14 Hours**						

Reference Book(s):	**1.** Mastering AWS Security: Create and maintain a secure cloud ecosystem by Albert Antony,Oct 2017,1st Edition, published by Packt. 2. Practical Cloud Security by ChrisDotson Copyright©2019 ChrisDotson. Published by O'Reilly Media. 3. Empirical Cloud Security: A Guide To Practical Intelligence to Evaluate Risks and Attacks by Aditya K. Sood, Apr 15, 2021, 1st Edition, published by Packt 4. Implementing DevSecOps Practices: Understand application security testing and secure coding by integrating SAST and DAST by Vandana Verma Sehgal, Dec22, 2023,1stEdition,published by Packt
Course Outcome	● Understand the core security challenges in cloud environments and apply identity and access management (IAM) principles to secure cloud resources. ● Implement network security measures, encryption techniques,and secure virtualization methods. ● Design secure cloud architectures, integrate DevSecOps practices to enhance security in cloud- native applications. ● Analyze, detect and mitigate cloud security threats using security monitoring tools.

<table>
<tr><td>Course Code</td><td>UE23AM343BB1</td><td>Course Title</td><td colspan="5">Natural Language Processing with Deep Learning</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>6</td><td>Type of Course</td><td colspan="5">Elective - IV</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>Python, Pytorch, and NLP libraries such as Spacy, NLTK, Open AI, and Huggingface transformer models.</td><td>Desirable Knowledge</td><td colspan="5">UE23AM432AA1 - Advanced Foundations for Machine Learning, UE22AM352A - Machine Learning. Prior exposure to text mining and classical NLP techniques in UE23AM342AA3 - Social Computing will be helpful.</td></tr>
<tr><td>Prelude</td><td colspan="7">This comprehensive course on modern NLP deals with the main learning challenges in NLP, i.e., semantic disambiguation for natural language understanding (NLU) and generation (NLG). Modern NLP techniques structure this learning problem as an end-to-end deep learning problem compared to the earlier generation approach of using a stack of general-purpose linguistic structures of syntax and semantics. This course focuses on the current generation of NLP, starting with deep learning-based NLP and ending with RAG-based NLG tasks. This is not a course on LLM but a course on modern NLP, i.e., it provides an end-to-end coverage by acquainting students with the various NLU and NLG tasks, training and tuning strategies of models for NLU and NLG tasks, and the corresponding evaluation metrics.</td></tr>
<tr><td>Course Objectives:</td><td colspan="7">● Learn basic neural NLP building blocks.
● Learn the internals and pre-training strategies of the Pretrained Large Language Models for NLU tasks.
● Learn the internals of the Causal Decoder-based Large Language Models for generative tasks, model optimization methods, and multimodal NLP.
● Learn Natural Language Generation(NLG) using Retrieval Augmented Generation(RAG)</td></tr>
<tr><td>Course Contents:</td><td colspan="7">Unit 1: Fundamentals of Neural NLP
Classical NLP - Language model: n-gram language model. Smoothing. Perplexity as an evaluation measure. Basic Text Mining – text segmentation, entity recognition, coreference resolution, dependency parsing, and relation extraction. Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks. Sparse models for vector semantics: TFIDF and Pointwise Mutual Information(PMI). Distributional semantics – Word2Vec (Skip-gram, CBOW), GloVe, fastText. Convolutional neural network for NLP -Finding a pattern in text. Neural sequence model - Encoder Decoder architecture, Autoregressive vs. Teacher Forcing in sequence models. Contextual Embedding – ELMo. Transfer Learning – Scenarios, Transductive vs. Inductive transfer learning, Catastrophic Forgetting, and Lifelong Learning. Multitask Learning – soft and hard parameter sharing, hierarchical multitask.

14 Hours

Unit 2: Modelling and Pre-training Strategies in Pre-trained Language Models(PLM) for NLU tasks
Transformer: attention, self-attention, masked self-attention, cross-attention, co-attention, causal attention, positional embedding, residual connection, layer normalization. Text Classification: BERT- Tokenization(BPE), Architecture, Pretraining – MLM and NSP objectives, Fine-tuning. RoBERTa– pretraining without NSP. ALBERT - factorized embedding parameterization, cross-layer parameter sharing. DistillBERT - knowledge distillation. ELECTRA - Replaced Token Detection (RTD) task. Benchmark– GLUE. Text Summarization: Extractive vs. Abstractive, BERT for summarization, BART– denoising pre-training. T5 – Denoising autoencoder in pretraining, unified text-to-text framework. Pegasus –Gap Sentence Generation (GSG) pre-training objective. LongFormer– combining local windowed attention, dilated sliding window attention, and global attention. Machine Translation: Vauquois Triangle and challenges, mBART and mBART-50 – multi-lingual pretraining, denoising auto-encoder. XLM- cross-lingual LM, maskedLM (MLM), Causal LM (CLM).XLM-RoBERTa – modified RoBERTa,MLM objective from XLM, Translational language model (TLM). Reference-based and Reference-free metrics for PLM-based NLU tasks – PERPLEXITY, BLEU, ROUGE, METEOR, BERTScore, Burstiness,QAEval, BLEURT, GRUEN.

14 Hours</td></tr>
</table>

	Unit 3: Language Models for Natural Language Generation(NLG) tasks and Multimodal NLP Types - Base, Instruction-tuned, and fine-tuned LLM, Small Language Model (SLM). Bias, Hallucination, and Emergent Abilities in LLM. GPT –Architecture, Building GPT like LLM. Training in ChatGPT vs. GPT. Instruction Tuning - Limitations of pre-training and fine-tuning in pre-trained Language model (PLM), Instruction Tuning in Google FLAN, T0. Model Architecture - Encoder only, Encoder Decoder, Causal Decoder, Prefix Decoder. Causal Decoder models–contextual embedding, decoding strategies. Hyperparameters – temperature, frequency penalty, presence penalty, context width, and max response. LLMs for NLG task – GELU and ADAMW in GPT, GPT1 to GPT3, Llama - Rotary Positional Embedding (RoPE), Others - InstructGPT, GPT-NeoX-20B, Redpajama, Dolly, Falcon, Alpaca. Model Optimization -Hyperparameter tuning, Pruning of model, Model distillation. Prompt-based tuning – In context Learning, hard vs. soft prompt, Prefix tuning. Parameter efficient fine-tuning (PEFT) – Quantization, LoRA, QLoRA, RLHF. Metrics for NLG tasks – Reference-based as in NLU, Reference-free (Supert, ROUGE-C, BLANC, Burstiness), and LLM-based metrics (GEMBA, G-eval). Multimodal NLP – Semantic challenges, complexities in figurative language constructs. Three generations of multimodal NLP- VQA, Visual Language Model, and multimodal LLM(MLLM). Architectures – multimodal converter and perceiver. Technologies- ViT, CLIP, Q-Former. Multimodal LLM–BLIP2, FLAMINGO, LLAVA. **14 Hours** **Unit 4: NLP Tasks (RAG) as a combination of IR and NLG tasks** RAG in NLP - a combination of Information Retrieval (IR) and NLG problems, Design of RAG-enabled systems.RAG indexing pipeline – Chunking, Embedding, and Storage. RAG generation pipeline – Retrieval, Augmentation, and Generation. Prompt engineering in Augmentation -Zero-shot, Few-shot, Chain of thought, Contrastive chain of thought, Tree of thought, Thread of thought, and ReACT prompting. RAG Progression - Naïve, Advanced, and Modular RAG. Advancement in Advanced RAG- Retrieval, Augmentation, and Generation. Other variants of RAG in NLP – Knowledge Graph(KG) RAG, Multimodal RAG, RAG with cognitive architectures -C-RAG and SELF-RAG. Agentic RAG – implementation APIs. RAG evaluation metrics– Retrieval metrics for IR Tasks (Precision@K, MAP, MRR, NDCG), RAG Metrics for NLG tasks (RAG Triad- context relevance, answer relevance, and Groundedness).RAG Evaluation Frameworks - RAGA. **14 Hours**
Text Book(s):	1. Speech and Language Processing, Jurafsky and Martin, 3rd edition draft dated Jan 7th, 2023 available 2. Natural Language Processing with Transformers – Building Language Applications with Hugging Face, by Lewis Tunstall, Leandro von Werra & Thomas Wolf, OREILLY. 3. Build a large language model from scratch – Sebastian Raschka, Manning, 2024.
Reference Book(s):	1. Natural Language Processing with Pytorch, Build Intelligent Language Applications Using Deep Learning, by Delip Rao and Brian McMahan, OREILLY. 2. Transformers for Natural Language Processing, Denis Rothman, Packt (2nd Edition, 2022). 3. CS22N (Natural Language Processing with Deep Learning) : Stanford/Winter, 2025. 4. Many web resources and published papers about the language models mentioned.
Course Outcome	● Should be able to use a basic Neural NLP workflow. ● Should be able to use transformer-based pre-trained LLM for standard NLU tasks ● Should be able to understand and appreciate the internals of LLM for NLG tasks, model optimization, and fine-tuning methods, and Multimodal NLP ● Should be able to use a combination of techniques (retrieval and generation) for advanced NLG tasks.

Course Code	UE23AM343BB2	Course Title	Deep Reinforcement Learning				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - IV				
AI Tools /Tools/Languages	Pytorch, Python Libraries and tools.	Desirable Knowledge	Machine Learning, Advanced Foundations for Machine Learning.				

Prelude

The goal of reinforcement learning is to enable AI agents to make optimal decisions in complex, dynamic environments by interacting with them and receiving rewards or penalties for their actions. Deep Neural Network pushes the envelope, allowing agents to handle complex, high-dimensional problems and make decisions based on unstructured data. This advanced Deep Learning course focuses on Deep Learning enabled methods of Reinforcement Learning. The course starts by revisiting the foundations of Reinforcement Learning and then follows up with Deep-Q network, Policy Gradient Methods, and Actor-Critic methods. The course ends with a discussion on advanced topics such as distributed DQN, multi-agent reinforcement learning etc.

Course Objectives:

- Get familiar with the fundamental principles of reinforcement learning and the scope of deep learning in it.
- Get familiar with the Deep-Q network.
- Get familiar with Policy-Gradient and Actor-Critic methods.
- Get Familiar with advanced topics such as Distributed DQN, Multi-agent DQN.

Course Contents:

Unit 1: Introduction to Reinforcement Learning

The Multi-arm Bandit – exploration and exploitation, epsilon greedy strategy, Softmax selection policy. Solving Ad Placement – state, action, and reward, contextual bandit, solving contextual bandit by building a model – Markov property in a Markov Decision Process (MDP), Predicting future rewards with Policy functions, Value Functions, and Optimal policy.

14 Hours

Unit 2: Deep-Q networks to predict best states and actions

State-space, action-space, state-value, action-value, policy function, Q-function, and Q-learning. Deep-Q network with Deep Learning. Off-policy learning and On-policy learning. Preventing Catastrophic forgetting by Experience Replay.

14 Hours

Unit 3: Policy-Gradient Method and Actor-Critic Methods

Policy Gradient Methods - Neural network as the policy function: stochastic policy gradient, exploration.
Policy Gradient Algorithm- objective function, action reinforcement, log probability, and credit assignment.
The REINFORCE algorithm as the implementation of the Policy Gradient method.
Actor-critic methods – combining Q-learner with Policy-learner, Distributed training, Advantage Actor-critic, and N-step Actor-critic.

14 Hours

Unit 4: Advanced Topics

Interdisciplinary applications – traffic network, autonomous driving, games, finance, healthcare, Natural Language Processing etc. Distributional DQN – issues with Q-Learning, Distributional Bellman Equation, Dist-DQN implementation. Multi-agent reinforcement learning – neighborhood Q-learning, mixed competitive-collaborative game. Inverse Reinforcement Learning. Interpretable reinforcement learning.

14 Hours

Text Book(s):

1. "Reinforcement Learning: An Introduction", By Richard S. Sutton and Andrew G. Barto (2nd Edition, 2020), http://incompleteideas.net/book/RLbook2020.pdf.
2. “Deep Reinforcement Learning in Python: A Hands-On Introduction”, By Laura Graesser and Wah Loon Keng (Addison-Wesley Data & Analytics Series, 2020).
3. Practical Deep Reinforcement Learning with Python: Concise Implementation of Algorithms, Simplified Maths, and Effective Use of TensorFlow and PyTorch, Ivan Gridin, 2022, BPB.

Reference Book(s):

1. Natural Open AI Spinning Up in Deep RL : https://spinningup.openai.com/en/latest/
2. David Silver’s RL Course (UCL/DeepMind): https://davidstarsilver.wordpress.com/teaching/.

Course Outcome	At the end of this course, the student will be able to: ● Become comfortable with the scope of Reinforcement learning and the opportunity that Deep Learning offers to it. ● Become comfortable in implementing Deep-Q Learning. ● Become comfortable in implementing Policy-gradient and Actor-Critic methods. ● Deliver a meaningful course project using Deep Reinforcement Learning.

Course Code	UE23AM343BB3	Course Title	Deep Learning for Images				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – IV				
AI Tools /Tools/Languages	TensorFlow, PyTorch, OpenCV, Jupyter Notebooks.	Desirable Knowledge	Machine Learning, Advanced Foundations for Machine Learning.				
Prelude	This advanced deep learning course deals with deep learning techniques with image data. This course deals with Deep Learning techniques for diverse tasks such as Image Classification, Image Recognition, and Image Synthesis. While it does not focus on Image Processing, it specifically focuses on computer vision tasks that perform well with Deep Learning and does not deal with advanced computer vision tasks. Additionally, this course provides an in-depth tutorial of generative AI for image data i.e. both traditional deep learning methods and diffusion models. The course ends with a discussion on various interdisciplinary applications and evaluation approaches.							
Course Objectives:	● Learn deep learning techniques for computer vision tasks. ● Learn deep learning based generative techniques such as VAE, GAN, and ViT. ● Learn advanced deep learning technique for Image Synthesis i.e. Diffusion Models. ● Understand evaluation methodologies and metrics for image related tasks.						
Course Contents:	**Unit 1: Deep Learning based Computer Vision Tasks** Computer Vision – Tasks in Computer Vision, an interdisciplinary field. First Generation Deep CNN Architectures – LeNet, VGGNet, AlexNet. Wide CNN architecture–InceptionV1, InceptionV2, InceptionV3, ResNetXt, Alternative Connectivity patterns - DenseNet, Xception, and SE-Net. Mobile Convolutional Network – MobileNet v1, MobileNet v2, SqueezeNet.Image Classification by Transfer Learning with Pre-trained CNN models - steps and strategies. Transfer Learning with VGG, GoogLeNet, and ResNet. Deep Learning based Computer Vision tasks – Object recognition with YOLOv3 and R-CNN, Face Detection, Face Identification, Face Classification – OpenCV, VGGFace2, and FaceNet. **14 Hours** **Unit 2: Deep Learning for Image Synthesis** Evolution of image synthesis – Deep Neural Network, GAN, Transformer, Diffusion models and Latent Diffusion models. Taxonomy for Generative AI for image synthesis–Autoencoder, Adversarial Network, Transformer, Diffusion Models. Choosing the right architecture. Variational Autoencoder: VAE, βVAE, cVAE. Latent space interpolation and disentanglement. Generative Adversarial Networks (GANs): Optimality in GAN, Adversarial Training, and training challenges. Variants of GANs- DCGAN, WGAN, CycleGAN. Evaluation of GANs. Transformer-based Vision Models: ViT, DeiT and SWIM. **14 Hours** **Unit 3: Advanced Deep Learning for Image Synthesis** Introduction to Diffusion Models. Forward Diffusion – mathematical foundation. Reverse Diffusion – mathematical foundation. U-net architecture – regular vs. transposed convolution. U-net architecture for Denoising – U-net vs. Autoencoder, Building Diffusion model with U-Net– DDPM (Denoising Diffusion Probabilistic model), Comparing DDPM with VAE and GAN. Optimization Techniques for Diffusion models – Group Normalization, GELU, Sinusoidal position embedding, etc. Classifier-free Diffusion Guidance (CFDG). Hybrid Architecture – Latent Diffusion Models. **14 Hours** **Unit 4: Interdisciplinary Applications, Evaluation, and Metrics** Interdisciplinary Applications-Object detection and recognition, Image classification and segmentation, Medical Imaging, Autonomous Vehicles, Surveillance systems, and Generative modelling. Metrics for Image Classification and Object Recognition Tasks – Intersection over union (IoU), Precision, Recall, Average Precision, Mean Average Precision, F1 Score. Metrics for Image Synthesis tasks - Challenges of the evaluation of generative models i.e. Qualitative approaches – visual inspection, case studies. Quantitative approaches - Inception Score, Frechet inception Distance, Kernel Inception Distance, fidelity, and Diversity of generated images. Model-specific Evaluation techniques – VAE, GAN, and Diffusion models.						

	14 Hours
Text Book(s):	1. "Deep Learning for Vision Systems", Mohamed Elgendy, Manning Book, 2020.
Reference Book(s):	1. Natural "Tutorial on Diffusion Models for Imaging and Vision", Stanley Chan, January 1925, https://arxiv.org/pdf/2403.18103. 2. "Computer Vision: Algorithms and Applications", Richard Szeliski, January 2023, Springer.
Course Outcome	● Become comfortable in deep learning-based computer vision tasks. ● Become comfortable in deep learning-based image synthesis tasks using VAE, GAN, and ViT. ● Become comfortable in deep learning-based image synthesis tasks using Diffusion Models. ● Become comfortable in using the metrics for evaluating image relates tasks.

Course Code	UE23AM343BB4	Course Title	Machine Learning for Finance				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective – IV				
AI Tools /Tools/Languages	TensorFlow, PyTorch, OpenCV, Jupyter Notebooks.	Desirable Knowledge	Machine Learning. Advanced Foundations for Machine Learning will be an advantage .				
Prelude	Today, the value of Machine Learning in finance has become crucial. Analysts, Portfolio Managers, Traders, and Chief Investment Officers rely on ML techniques to help them derive useful information and insights from the dataset. This Applied Machine Learning course is an introductory, interdisciplinary course designed for future data scientists in the Finance domain. This case-study driven course dedicates its units to a series of case studies that apply different machine learning techniques in finance domain-specific use cases.						
Course Objectives:	• Learn applications of regression and supervised learning in Finance. • Learn applications of dimensionality reduction and unsupervised learning in Finance. • Learn applications of reinforcement learning in finance. • Learn applications of graph neural network in finance.						
Course Contents:	**Unit 1: Regression and Supervised Learning in Finance** Machine Learning Recap: Time series breakdown, Autocorrelation and stationarity, Traditional Time Series models (ARIMA). Deep Learning approach to Time Series modelling. Modifying time series data for regression modelling. Natural Language Processing recap: preprocessing, feature representation, and Inference. Case study 1: Stock Price Prediction using supervised learning. Case Study 2: Yield curve prediction using supervised learning. Case study 3: Modelling investor risk tolerance and Robo-advising using machine learning. Case study 4: Machine Learning model for Derivative pricing. Case Study 5: Supervised classification model to predict Loan Default probability. Case study 6: Trading strategy based on social media sentiment analysis. **14 Hours** **Unit 2: Dimensionality Reduction and Clustering in Finance** Dimensionality Reduction Techniques recap: PCA, K-PCA, t-SNE. Unsupervised Learning recap: K-means and Hierarchical clustering, Affinity propagation clustering. Case study 7: Finding an Eigen portfolio for asset allocation. Case study 8: Clustering for Grouping investors. Case Study 9: Portfolio allocation using Hierarchical Clustering. **14 Hours** **Unit 3: Reinforcement Learning in Finance** Reinforcement Learning recap: RL components and modelling framework. Reinforcement Learning models– model based and model free. Model free methods –value-based (Q Learning, SARSA, Deep-Q network), Policy-based (Policy Gradient).Case Study-10: Reinforcement Learning based Hedging Strategy. Case Study-12: Reinforcement Learning based Portfolio Allocation. **14 Hours** **Unit 4: Anti-Money Laundering in Bitcoin network using Graph Neural Network** Money laundering – Introduction, Illicit transactions in Bitcoin network. Graph Neural Network – message passing framework, graph convolution network, graph attention network. Understanding Elliptic Bitcoin dataset, using multilayer perceptron model to detect illicit transactions, using Graph Neural Network (GCN, GAT) to detect illicit transactions. **14 Hours**						
Text Book(s):	1. Machine Learning & Data Science Blueprints for Finance, Hariom Tatsat, Sahil Puri & Brad Lookabaugh, OREILLY, 2020 2. Python for Finance: Mastering Data-Driven Finance, Yves Hilpisch, O'REILLY, 2019.						
Reference Book(s):	1. Advances in Financial Machine Learning, Marcos López de Prado, Wiley, 2018 2. Hands-On Machine Learning for Algorithmic Trading, Stefan Jansen, Packt, 2019 3. Machine Learning for Asset Managers (Elements in Quantitative Finance), Marcos López de Prado,						

	Cambridge University Press, 2020.
Course Outcomes	● Become comfortable in building classification and regression models for financial applications. ● Become comfortable in building unsupervised learning-based models for financial applications. ● Become comfortable in building reinforcement learning based models for financial applications. ● Become comfortable in building graph neural network-based models for financial applications.

<table>
<tr><td>Course Code</td><td>UE23AM343BB5</td><td>Course Title</td><td colspan="5">Large Language Models Agents</td></tr>
<tr><td rowspan="2">Program</td><td rowspan="2">B.Tech CSE(AI & ML)</td><td rowspan="2">Hours per week/ Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>S</td><td>C</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>4</td><td>4</td></tr>
<tr><td>Semester</td><td>6</td><td>Type of Course</td><td colspan="5">Elective – IV</td></tr>
<tr><td>AI Tools /Tools/Languages</td><td>HuggingFace, LangChain, CrewAI, OpenAI, FAISS, DevOps: Docker, GitHub Actions, Deployment: Streamlit, FastAPI.</td><td>Desirable Knowledge</td><td colspan="5">Python, Fundamentals of Machine Learning, NLP Basics.</td></tr>
<tr><td>Prelude</td><td colspan="6">This course goes beyond foundational understanding of Generative AI and introduces students to autonomous AI systems built on top of LLMs, integrating cutting-edge tools like LangChain, AutoGPT, and LLMOps frameworks. The emphasis is on system-level thinking, engineering workflows, deployment strategies, and real-world domain agentic applications.</td></tr>
<tr><td>Course Objectives:</td><td colspan="6">● Learn Understand the software engineering, customization, and deployment lifecycle of LLMs.
● Explore the mechanics of building intelligent agents using LLMs in real-world scenarios.
● Gain hands-on experience with tools like LangChain, AutoGPT, CrewAI, and vector stores.
● Explore emerging practices in LLMOps, edge deployment, and model observability.
● Implement domain-specific agents with responsible and explainable behavior.
●</td></tr>
<tr><td>Course Contents:</td><td colspan="6">Unit 1: Foundations of Agentic LLMs and Prompt Engineering
Differences between GenAI Applications vs. Agentic Systems, Introduction to Prompt Engineering: Structure, Templates, Prompt Flow, Task decomposition: Zero-shot, Few-shot, Chain of Thought, ReAct, Tree-of-Thought, Autonomous Agents: Planner-Executor pattern, Memory and Tools, Agent Frameworks: AutoGPT vs. BabyAGI vs. CrewAI overview, LangChain Ecosystem: Chains, Agents, Tools, Practical Lab: LangChain tool use, create an autonomous task handler. Tools: LangChain, OpenAI API, LM Studio, Jupyter Notebooks.
14 Hours
Unit 2: Customization Techniques: Fine-Tuning, RLHF, and Soft Prompts
Fine-Tuning LLMs: Quantization, LoRA, Adapter Layers, Prompt Tuning and Soft Prompts: PEFT libraries (e.g., HuggingFace PEFT), Reinforcement Learning with Human Feedback (RLHF): Aligning LLMs to human preferences, Datasets for fine-tuning: Open Assistant, Anthropic HH, Red-Teaming examples. Model Compression and Transfer: TinyGPT, Efficient transformers, Case Study: Fine-tuning a Q&A bot for educational content. Assignment: Fine-tune a distilled model for a legal chatbot use case. Tools: HuggingFace Transformers & PEFT, TRL (for RLHF), Weights & Biases.
14 Hours
Unit 3: Agent Deployment Engineering and MLOps for LLMs
LLMOps Introduction: CI/CD for LLMs, model versioning, Model Monitoring: Prompt drift, hallucination detection, input sanitization, Vector Databases: FAISS, Pinecone, ChromaDB, Weaviate, Deployment Frameworks: Docker, FastAPI, Streamlit, LangServe, Edge AI for Agents: Jetson Nano, Coral, TinyML applications for low-resource LLM agents, Pipelines: Apache Airflow, DVC for experiment tracking, Lab: Dockerize a LangChain agent with logging and monitoring. Tools: Docker, GitHub Actions, Airflow, LangServe, Streamlit. 14 Hours
Unit 4: Domain-Specific Autonomous Agents & Ethics
Healthcare Agents: Medical reasoning assistants, symptom-checkers, EdTech Agents: Personalized tutoring bots, syllabus planners, AgriTech Agents: Voice-based farmer advisory agents (tie-in with Kisan Vaani), Legal Agents: Document summarizers, legal search assistants, Multi-agent Collaboration: CrewAI hands-on for complex workflows, Ethical Concerns: Bias propagation, jailbreaking, LLM agent hallucination risks, Explainable Agents: LLM interpretability, traceability of decision steps. Group Project: Multi-agent system for a real-world use case (e.g., education or agri). Tools: CrewAI, LlamaIndex, OpenAI Function Calling, Explainable AI libraries.
14 Hours</td></tr>
</table>

Text Book(s):	1. Sinan Ozdemir - Quick Start Guide to Large Language Models, Addison-Wesley, 2023
Reference Book(s):	1. Denis Rothman - Transformers for NLP, Packt, 2023 2. Emily Webber - Pretrain Vision and LLMs in Python, Packt, 2023 3. Sebastian Raschka - Machine Learning Engineering with MLOps, Packt, 2023.
Course Outcomes	● Build and customize LLM pipelines using advanced prompting and feedback mechanisms. ● Implement autonomous AI agents with reasoning and memory. ● Deploy LLM-powered applications using LangChain, AutoGPT, and MLOps pipelines. ● Apply agent frameworks across domains such as health, agriculture, education, and law..

Course Code	UE23CS343BB6	Course Title	Information Security				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - IV				
AI Tools /Tools/Languages	SEED Labs VM, Scapy, BurpSuite,Metasploit, Nmap, etc.	Desirable Knowledge	-				

Prelude	This course will present security aspects from a secure software life cycle process – requirement, architecture, design, coding, and testing. Students will have opportunity to dwell well in to technical"how to" with hands-on sessions, assignments, and some case study discussions.
Course Objectives:	• To understand various cyber threats and attacks and secure software development process. • To learn attack and defence mechanisms for buffer overflow, shell shock attack,etc. • To understand the concept of threat modelling and its application. • To learn about the most common web application security vulnerabilities. • To understand and apply various penetration testing techniques and tools.
Course Contents	**Unit 1: Introduction and Privilege Escalation Attacks** Software Threats, Attacks and Vulnerabilities, CIA Triad, OWASP Top 10, CVE, Security and reliability, Security vs. privacy, Cyberattack Types, Anatomy of an Attack, Security Concepts and Relationships. Use cases and Misuse cases, Misuse case legend, Security use case vs Misuse case, Secure Software Development Life Cycle (SDL). Case Study: Target case study. Set-UID program: Need for privileged programs, Set-UID mechanism, Superman story, Attack surfaces, Invoking other surfaces, Principle of least privilege. Environment variables and attacks: Environment variables, Attack surface, Attacks via Dynamic linker, External program, and Library. Lab: Set-UID program & Environment variables and attacks. Shellcode attack on Set-UID and CGI programs. **14 Hours** **Unit 2: Software Vulnerabilities and Malicious Software** Buffer overflow attack: Program memory layout, Stack and function invocation, Stack buffer-overflow attack, Attacks with Unknown address and Buffer size, Shellcode, Countermeasures & Defeating it. Return-to-libc attack: Introduction, Launch the attack part I & part II. Format string vulnerability: Introduction to functions and format string, Vulnerable program, Exploiting the vulnerability, Code injection attack, Countermeasures. **Case study: Target case study**. Malware and its Types, Malware analysis: Conifer, Morris, Stuxnet worm, Ransomware. **14 Hours** **Unit 3: Threat modelling and Basic Web Security** Threat Modelling, Trust Boundaries, Attack Surfaces, Brainstorming, Modelling Methods, STRIDE model and variants, Defensive tactics, and Technologies, Privacy Threats, Taxonomy and Types. Web security basics, Attacks on HTTP GET and POST services, Cross Site Request Forgery (XSRF/CSRF): Cross-site requests and its problems, CSRF attacks, Counter measures. **Case study: Apple-Privacy vs Safety.** **14 Hours** **Unit 4: Web application security and Penetration Testing** (XSS/CSS)Attack: CSS attack, CSS attacks in action, Self-propagation, Preventing CSS attacks. SQL injection attack: Introduction to SQL, interacting with database in web, Launching SQL injection attacks, Countermeasures. Static analysis, Penetration testing: Introduction, Benefits, Drawbacks, Penetration testing tools and Fuzzing, Patching. **14 Hours**

TextBook(s):	1.“Computer & Internet Security: A Hands-on Approach”,Wenliang Du, 2nd Edition/3rdEdition.
Reference Book(s):	1:“Computer Security: Principles and Practice”, William Stallings and Lawrie Brown, Pearson Education, 3rd Edition, 2014. 2:“Secure Programming with Static Analysis”,Brian Chess and JacobWest, Pearson Education,2007.
Course Outcome	● Identify possible misuse cases in the context of software development. ● Defend against various attacks and how to write secure code. ● Apply threat modelling techniques to expose inherent vulnerabilities in applications. ● Design and develop secure web applications. ● Exploit software vulnerabilities and launch attacks.

Course Code	UE23CS343BB7	Course Title	Mobile and Autonomous Robot				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - IV				
AI Tools /Tools/Languages	C,C++,Python, ROS, Webots, AirSim, Gazebo	Desirable Knowledge	Introduction to Autonomous Systems with ROS , -Linear Algebra and its Applications				
Prelude	This course provides an introduction to autonomous robots, focusing on the principles and techniques behind their design, programming, and control. Students will explore key topics such as perception, planning, and navigation, while gaining hands-on experience through projects and simulation using ROS/ROS2. The course combines theoretical foundations with practical applications across both terrestrial and aerial robotic platforms.						
Course Objectives:	• Design Understand the evolution, current trends, and research directions in autonomous mobile robotics. • Learn the theoretical and practical fundamentals of designing and operating autonomous robots. • Explore core topics including perception, localization, planning, and control. Gain hands-on experience with robot simulation and programming using ROS/ROS2.						
Course Contents	**Unit 1: Introduction to Robotics and Locomotion:** Robotics overview: Past, Present and Future; Robot Hardware and Software. AI in Robotics: Machine Learning Basics for Robotics. Locomotion: The basics of SOTA locomotion systems, Legged Robots, Wheeled Robots, Aerial Robots. Robot Kinematics, ROS overview: ROS/ROS2 for robotics, ROS architecture and communication protocols, ROS packages for robot hardware control**.** **14 Hours** **Unit 2: Perception Introduction to Perception, Sensors for Robots, Visual and inertial measurements:** Gyroscope, accelerometers, IMU, GPS, Range sensors, camera vision and LiDAR. Fundamentals of Computer Vision and Image Processing and filtering, Feature Extraction: Object detection and Place Recognition, Stereo vision and 3D perception, Feature Extraction Based on Range Data (Laser, Ultrasonic). **14 Hours** **Unit 3: Localization** Introduction and Challenges of Localization, To Localize or Not to Localize, Belief Representation, Map representation, Probabilistic Map-Based Localization: Markov localization, Kalman Filter localization, Autonomous Map Building: SLAM, EKF SLAM, Particle filter SLAM, GraphSLAM, Open challenges in SLAM.. **14 Hours** **Unit 4: Planning and Navigation Introduction Path Planning and Navigation, Path planning:** Graph search and Potential field path planning. Obstacle avoidance: Bug algorithm, Vector field histogram, bubble band technique, Curvature velocity techniques, Dynamic window approaches. Navigation Architectures. Revisiting Navigation using Reinforcement Learning and Imitation learning: A Robotics Perspective, Applications and Social Implications. **14 Hours**						

TextBook(s):	1: Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series) second edition by Roland Siegwart, Illah Reza Nourbakhsh, Davide Scaramuzza 2: Artificial Intelligence for Robotics Build Intelligent Robots that Perform Human Tasks Using AI Techniques By Francis X. Govers · 2018.
Reference Book(s):	1: Introduction to Autonomous Robots: Nikolaus Correll, Magellan Scientific, 2016. 2: ROS Robot Programming, ROBOTIS Co., Ltd. From the basic concept to practical programming and robot application. YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, and TaeHoon Lim. 3: Introduction to Robotics: Mechanics and Control 4th Edition, John Craig, ISBN-13: 978-0133489798, Pearson; 4th edition 4: Probabilistic Robotics By Sebastian Thrun, Wolfram Burgard and Dieter Fox. ISBN-13: 978-0262201629, ISBN-10: 0262201623. Intelligent Robotics and Autonomous Agents series; 1st Edition
Course Outcome	● Explain the principles and applications of autonomous mobile robots, including their key components and system architecture. ● Identify and integrate appropriate sensors and actuators for robot perception and control. ● Develop and simulate robotic systems using ROS/ROS2 for navigation, planning, and hardware interfacing. ● Apply algorithms for localization, path planning, and obstacle avoidance in real or simulated environments. ● Design and implement autonomous robotic solutions for terrestrial and aerial platforms.

Course Code	UE23CS343BB8	Course Title	Security for Internet of Things			
Program	B.Tech CSE(AI & ML)	Hours per week/	L	T	P	S
		Credit Assigned	4	0	0	4
Semester	6	Type of Course	Elective - IV			
AI Tools /Tools/Languages	Wireshark, Yersinia, VoIP Hopper, Bettercap,aircrack-ng	Desirable Knowledge	Internet of Things			

Prelude	This course enables the learner to understand the security threats associated with Internet of Things and perform security testing on connected devices within our homes and enterprise to build a better model for protecting ourselves.
Course Objectives:	● Understand the threats of IoT devices. ● Understand vulnerabilities, possible attacks and learn the security testing methodology for networks, IoT hardware and radio protocols. ● Enhance knowledge with hands-on experience on IoT security tools and attack analysis. ● Understand the various Security measures to be adopted to secure IoT devices.
Course Contents	**Unit 1: Introduction to IoT Security** Introduction to IoT Security, Traditional security vs IoT security, Basic concepts of IoT Architecture from security perspective, Challenges of IoT Security, OWASP Top 10security risks and consumer IoT security guidance, Threat Modelling for IoT attack, Common IoT threats, Network Hacking: VLAN hopping in IoT networks. **14 Hours** **Unit 2: Network Security and Hardware Security for IoT devices** MQTT authentication, Analysing Network protocols: Wireshark dissector and Nmap Scripting Engine module for the DICOM protocol, exploiting zero-configuration Networking: UPnP, mDNS, DNS-SD, and WS-Discovery, Hardware Hacking: UART, JTAG, and SWD Exploitation- Hacking an STM32F103 microcontroller using UART and SWD.SPI and I2C, Firmware Hacking. **14 Hours** **Unit 3: Radio Hacking and Smart IoT devices hacking Radio Hacking** Short Range Radio: Abusing RFID, Bluetooth low energy, Medium Range Radio: Hacking Wi-Fi, Long Range Radio: LPWAN, Smart home, Hacking the smart home: Gaining physical entry to a building, cloning a keylock system's RFID Tag, Jamming the Wireless alarm. **14 Hours** **Unit 4: Targeting the IoT Ecosystem and Secure Design of IoT Devices** Playing back an IP Camera stream, Analysing IP Camera Network traffic, Extracting the video stream, attacking a Smart treadmill, Secure design goals: Mitigate automated attack risks, Secure points of integration, Hardware protection measures, IoT IAM infrastructure: PKI for IoT, Revocation support: OCSP, SSL pinning, Authorization and access control with OAuth 2.0 Cryptographic Controls for IoT Protocols. **14 Hours**
TextBook(s):	1:"Practical IoT Hacking", Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, Beau Woods, March 2021, No Starch Press Publishers, ISBN: 9781718500907.

Reference Book(s):	1:"Practical Internet of Things Security", Brian Russell, Drew VanDuren, Packt Publishers, 2nd Edition.
Course Outcome	• Identify and describe the variety of IoT systems architectures, essential components and challenges specific to IoT systems. • Analyze various network and hardware security mechanisms for IoT devices. • Gain hands-on experience on different tools to target IoT ecosystem. • Analyze and apply appropriate security and privacy solutions for real-world applications.

Course Code	UE23CS343BB9	Course Title	Applied ML in IoT with Tiny ML				
Program	B.Tech CSE(AI & ML)	Hours per week/ Credit Assigned	L	T	P	S	C
			4	0	0	4	4
Semester	6	Type of Course	Elective - IV				
AI Tools /Tools/Languages	Arduino Nano33 BLE sense , Arduino IDE , Google Colab	Desirable Knowledge	-				

Course Objectives:

- Design and deploy simple Machine Learning models on microcontrollers for various IoT applications.
- Preprocess sensor data, optimize code, and evaluate the trade-off between model accuracy and resource consumption.
- Utilize tools like TensorFlow Lite Micro and TinyML Kit for TinyML development.
- Integrate sensor data acquisition, communication protocols, and datapipelines for real-time TinyML applications.

Course Contents

Unit 1: Foundations of Machine Learning

Introduction to IoT & TinyML, The ML paradigm, Building blocks of DL, NN, Regression with Dense NN, Classification with Dense NN, Image Classification using CNN, Datasets and Model Performance Metrics, Preventing Overfitting. **14 Hours**

Unit 2: TensorFlow Lite for Microcontrollers & TinyML Kit

What Is TensorFlow Lite for Microcontrollers?, Build Systems, Supporting a New Hardware Platform, Supporting a New IDE or Build System, Integrating Code Changes Between Projects and Repositories, Contributing Back to Open Source, Supporting New Hardware Accelerators, Understanding the File Format, Porting TensorFlow Lite Mobile Ops to Micro, TinyML Kit Overview, TinyML Kit Setup, TinyML Kit Sensor Testing, Sensor Testing, Sensor Fusion. **14 Hours**

Unit 3: Applications and Deployment to Microcontrollers (MCUs)-1

TF-Lite, TFL-Micro, TFL-Micro Hello- World example, KeyWord Spotting (KWD) using Edge Impulse, KDW dataset creation, Micro Speech Example, Workflow, Model development & testing, Person-detection example, Image classification. **14 Hours**

Unit 4: Applications and Deployment to Microcontrollers (MCUs)-2

Magic Wand: Building and Application, Training a Model, Building a Weather Station with TensorFlow Lite for Microcontrollers, Voice Controlling LEDs with Edge Impulse. **14 Hours**

TextBook(s):

[1] Pete Warden and Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low- Power Microcontrollers, O'Reilly Media, 2020.

[2] Vijay Janapa Reddi at Harvard and open-source collaborators, Machine Learning Systems with TinyML, Open- source collaborative-effort book, 2023-present.

[3] Gian Marco Iodice and Ronan Naughton, TinyML Cookbook: Combine artificial intelligence and ultra- low-power embedded devices to make the world smarter, 2022.

[4] Francois Chollet, Deep Learning with Python, Manning, Second Edition,2021.

Course Outcome	● Understand the fundamental concepts of ML,Deep Learning(DL),and their application in the IoT domain. ● UtilizeTensorFlow LiteMicro and TinyML Kit for developing and deploying ML models on microcontrollers. ● Design and implement TinyML applications for various functionalities like keyword spotting, Image classification and sensor fusion. ● Evaluate and interpret the performance of TinyML models on real hardware.