॥ यः क्रियावान् स पण्डितः ॥

SAVITRIBAI PHULE PUNE UNIVERSITY

(Formerly University of Pune)

SYLLABUS

T. E. COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE)

2024 PATTERN

(With effect from Academic Year 2026-27)

National Education Policy (NEP) - 2020 Compliant Curriculum

www.unipune.ac.in

॥ यः क्रियावान् स पण्डितः ॥

Contents

Nomenclature

AEC Ability Enhancement Course

AICTE All India Council for Technical Education

CEP Community Engagement Project

CSE (AI) Computer Science and Engineering (Artificial Intelligence)

EEM Entrepreneurship/Economics/Management Courses

MDM Multidisciplinary Minor

MOOC Massive Open Online Course

NEP National Education Policy

NPTEL National Programme on Technology Enhanced Learning

OE Open Elective

PCC Program Core Course

PEO Programme Educational Objectives

PSO Program Specific Outcomes

SWAYAM Study Webs of Active-Learning for Young Aspiring Minds

UGC University Grants Commission

VEC Value Education Course

VEC Value Education Course

VSE Vocational and Skill Enhancement

WK Knowledge and Attitude Profile

Preface by Board of Studies

Dear Students and Teachers,

We, the members of Board of Studies Computer Engineering, are very happy to present Third Year CSE (AI) syllabus effective from the AY Year 2025-26. Subsequently this will be carried forward for TE and BE in the AY 2026-27, 2027-28, respectively.

Computer Science Engineering (Artificial Intelligence) have emerged as transformative forces reshaping industries, driving innovation, and impacting our daily lives. Recognizing the growing importance and pervasive nature of these fields, we have designed this comprehensive syllabus to equip students with the foundational knowledge and practical skills. This curriculum is meticulously crafted to provide a holistic learning experience, blending theoretical concepts with hands-on applications. The revised syllabus falls in line with the objectives of NEP-2020, Savitribai Phule Pune University, AICTE, UGC, and various accreditation agencies by keeping an eye on technological developments, innovations, and industry requirements.

Learners are now getting sufficient time for self- learning either through online courses or additional projects for enhancing their knowledge and skill sets. We believe that this well-structured and comprehensive syllabus will serve as a robust foundation for aspiring Computer Engineering and AI professionals, enabling them to contribute significantly to the technological progress and address the challenges of the 21st century.

We would like to place on record our gratefulness to the faculty, students, industry experts and stakeholders for having helped us in the formulation of this syllabus.

Dr. Nilesh Uke
Chairman - Board of Studies **Computer Science Engineering**
(Artificial Intelligence)
Savitribai Phule Pune University

Members of Board of Studies – Computer Science Engineering (Artificial Intelligence)	
Dr. Pramod Patil	Dr. Dipti Patil
Dr. Dhananjay Kshirsagar	Dr. Amol Potgantwar
Dr. Sachin Babar	Dr. Balwant Sonkamble
Dr. Suhasini Itkar	Dr. Sachin Sakhare
Dr. Dipak Patil	Dr. Vandana Dhingra
Dr. Deepali Ujalambkar	Dr. Vaishali Vikhe
Dr. Pradip Jawandhiya	Dr. Sandeep Deshmukh

Program Specific Outcomes (PSO)

- **PSO1:** Demonstrate proficiency in essential concepts of computer science and data science and programming solutions.
- **PSO2:** Formulate robust software design, execution, and testing strategies employing a software paradigms and Artificial Intelligence knowledge to solve real word problems.
- **PSO3:** Apply the techniques of AI and Data Science for forecasting future events in the domain of Healthcare, Education, and Agriculture, Automation , Transport etc

Programme Educational Objectives (PEO)

Program Educational Objectives are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.

PEO	PEO Focus	PEO Statements
PEO1	Core competence	To produce graduates equipped with cutting-edge skills in Computer Engineering, Artificial Intelligence (AI) and Data Science (DS), with expertise in domains such as Machine Learning (ML), Natural Language Processing (NLP), Generative AI, enabling them to collaborate effectively in interdisciplinary teams to solve real-world industrial and societal challenges.
PEO2	Problem solving skills and Ethics	To empower graduates to think critically, apply mathematical, computational, and ethical frameworks, and design scalable, secure, and fair AI-driven systems
PEO3	Professionalism and Lifelong Learning	To inculcate the ability to adapt to changing technology through continuous learning and contribute to research, innovation, and entrepreneurship in AI and Data Science.

Knowledge and Attitude Profile (WK)

A Knowledge and Attitude Profile (KAP), often represented as WK (Knowledge and Attitude Profile) in some contexts, is a framework or assessment tool used to evaluate an individual's knowledge and attitudes related to a specific area, topic, or domain.

WK1	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK2	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
WK4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK5	Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
WK6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK7	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK9	Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 55.

Programme Outcomes (PO)

Program Outcomes are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, attitude and behaviour that students acquire through the program. On successful completion of B.E. in Artificial Intelligence and Data Science, graduating students/graduates will be able to:

PO1	Engineering knowledge	Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	Problem analysis	Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	Design / Development of Solutions	Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4	Conduct Investigations of Complex Problems	Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	Engineering Tool Usage	Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	The Engineer and The World	Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7	Ethics	Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8	Individual and Collaborative Team work:	Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	Communication	Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
PO10	Project Management and Finance	Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11	Life-Long Learning	Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 56.

General Rules and Guidelines

- **Course Outcomes (CO):** Course Outcomes are narrower statements that describe what students are expected to know, and are able to do at the end of each course. These relate to the skills, knowledge and behavior that students acquire in their progress through the course.
- **Assessment:** Assessment is one or more processes, carried out by the institution, that identify, collect, and prepare data to evaluate the achievement of Program Educational Objectives and Program Outcomes.
- **Evaluation:** Evaluation is one or more processes, done by the Evaluation Team, for interpreting the data and evidence accumulated through assessment practices. Evaluation determines the extent to which Program Educational Objectives or Program Outcomes are being achieved, and results in decisions and actions to improve the program

Guidelines for Examination Scheme

Theory Examination: The theory examination shall be conducted in two different parts Comprehensive Continuous Evaluation (CCE) and End-Semester Examination (ESE).

Comprehensive Continuous Evaluation (CCE):

1. CCE of 30 marks based on all the Units of course syllabus to be scheduled and conducted at institute level.
2. Case studies included under each unit are intended to support applied learning and are part of Comprehensive Continuous Evaluation
3. These case studies will be assessed through internal assessment components such as presentations, assignments, or group discussions. They shall not be included in the End-Semester Theory Examination.
4. To design a Comprehensive Continuous Evaluation scheme for a theory subject of 30 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	12 Marks	Units 1 & Unit 2 (6 Marks/Unit)
2	Assignments / Case Study	12 Marks	Units 3 & Unit 4 (6 Marks/Unit)
3	Seminar Presentation / Open Book Test/ Quiz	06 Marks	Unit 5

5. CCE of 15 marks based on all the Units of course syllabus to be scheduled and conducted at institute level. To design a CCE scheme for a theory subject of 15 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	10 Marks	Units 1 & Unit 2 (5 Marks/Unit)
2	Seminar Presentation / Open Book Test/ Assignments/Case Studies	05 Marks	Units 3 & Unit 4

Format and Implementation of Comprehensive Continuous Evaluation (CCE)

- **Unit Test**
  - **Format:** Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, Create).
  - **Implementation**: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.
- **Sample Question Distribution**
  - Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
  - Understanding (2 Marks): Explain the principle of [Concept] in [Context].
  - Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario].
  - Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2.
  - Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].
- **Assignments / Case Study:** Students should submit one assignment or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.
  - **Format:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.
  - **Implementation:** Distribute the assignments or case study after covering Units 3 and 4. Provide clear guidelines and a rubric for evaluation.
- **Seminar Presentation:**
  - **Format:** Oral presentation on a topic from Unit 5, followed by a Q&A session.
  - **Deliverables:** Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation.
  - **Implementation:** Schedule the seminar presentations towards the end of the course. Provided students with ample time to prepare and offer guidance on presentation skills.
- **Open Book Test:**
  - **Format:** Analytical and application-based questions to assess depth of understanding.
  - **Implementation:** Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5.
- **Quiz:**

- **Format:** Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.
- **Implementation:** Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc

- **Example Timeline for conducting CCE:**
  - Weeks 1-4 : Cover Units 1 and 2
  - Week 5 : Conduct Unit Test (12 marks)
  - Weeks 6-8 : Cover Units 3 and 4
  - Week 9 : Distribute and collect Assignments / Case Study (12 marks)
  - Weeks 10-12 : Cover Unit 5
  - Week 13 : Conduct Seminar Presentations or Open Book Test or Quiz (6 marks)

- **Evaluation and Feedback:**
  - **Unit Test:** Evaluate promptly and provide constructive feedback on strengths and areas for improvement.
  - **Assignments / Case Study:** Assess the quality of submissions based on the provided rubric. Offer feedback to help students understand their performance.
  - **Seminar Presentation:** Evaluate based on content, delivery, and engagement during the Q&A session. Provide feedback on presentation skills and comprehension of the topic.
  - **Open Book Test**: Evaluate based on the depth of analysis and application of concepts. Provide feedback on critical thinking and problem-solving skills.

End-Semester Examination (ESE)

End-Semester Examination (ESE) of 70 marks written theory examination based on all the unit of course syllabus scheduled by university. Question papers will be sent by the University through QPD (Question Paper Delivery). University will schedule and conduct ESE at the end of the semester.

- **Format and Implementation:**
  - **Question Paper Design:** Below structure is to be followed to design an End-Semester Examination (ESE) for a theory subject of 70 marks on all 5 units of the syllabus with questions set as per Bloom's Taxonomy guidelines and 14 marks allocated per unit.
  - **Balanced Coverage**: Ensure balanced coverage of all units with questions that assess different cognitive levels of Bloom's Taxonomy: Remember, Understand, Apply, Analyze, Evaluate, and Create. The questions should be structured to cover:
    * Remembering: Basic recall of facts and concepts.
    * Understanding: Explanation of ideas or concepts.

- Applying: Use of information in new situations.
- Analyzing: Drawing connections among ideas.
- Evaluating: Justifying a decision or course of action.
- Creating: Producing new or original work (if applicable).

– **Detailed Scheme for 70 Marks:** Unit-Wise Allocation (14 Marks per Unit): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.

– **Detailed Scheme for 35 Marks:** Unit-Wise Allocation (08 Marks for Unit 1 , 09 Marks for Unit 2, Unit 3 and Unit 4): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.

***

Curriculum Structure - Semester V

Third Year Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern - Level 5.0

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Oral	Practical	Total	Theory	Tutorial	Practical	Total
PCC-301-CAI	Program Core Course	Theory of Computation	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC302-CAI	Program Core Course	Machine Learning	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-303- CAI	Program Core Course	Design and Analysis of Algorithms	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-304-CAI	Programme Core Course Lab 1	Design and Analysis of Algorithms Laboratory	-	-	2	-	-	25	25	-	50	-	-	1	1
PCC-305- CAI	Program Core Course Lab 2	Machine Learning Laboratory	-	-	4	-	-	25	-	25	50	-	-	2	2
PEC321-CAI	Programme Elective-I	Programme Elective Course-I	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC322-CAI	Programme Elective Course Lab 3	Programme Elective Laboratory-I	-	-	2	-	-	25	-	-	25	0	0	1	1
MDM331-CAI	Multi disciplinary Minor	Internet of Things	-	1	4	-	-	25	25	-	50	0	1	2	3
	Open Elective*	Open Elective*	2	-	-	30	70	-	-	-	100	2	0	-	2
ELC-342- CAI	Experiential Learning Course (Research Methodology)	Technical Seminar	-	-	2	-	-	-	25	-	25	-	-	1	1
Total			**14**	**1**	**14**	**150**	**350**	**100**	**75**	**25**	**700**	**14**	**1**	**7**	**22**

***Note:** Students can opt for Open Electives offered by different discipline/faculty like Arts, Science, Commerce, Management, Humanities or Inter-Disciplinary studies. Example – Open Elective I- IPR and Cyber Laws, Sustainability Development, Digital Personal Data Protection, The Constitution of India

Programme Elective I Courses	
PEC321ACAI	Cyber security
PEC321BCAI	Cloud Computing and DevOps
PEC321CCAI	Soft Computing
PEC321DCAI	Gaming and Animation

Third Year Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern - Level 5.0

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Oral	Practical	Total	Theory	Tutorial	Practical	Total
PCC351-CAI	Programme Core Course 1	Software Engineering and Project Management	3	-	-	30	70	-	-	-	100	3	0	-	3
PCC352-CAI	Programme Core Course 2	Deep Learning	2	-	-	30	70	-	-	-	100	2	0	-	2
PCC353-CAI	Programme Core Course Lab 1	Deep Learning Laboratory	-		4	-	-	25	-	25	50	0	0	2	2
PCC354-CAI	Programme Core Course Lab 2	Software Engineering and Project Management Laboratory	-	-	2	-	-	25	25	-	50	0	0	1	1
PEC361-CAI	Programme Elective Course	Programme Elective-II	3	-	-	30	70	-	-	-	100	3	0	-	3
PEC362-CAI	Programme Elective Course	Programme Elective-III	3	-	-	30	70	-	-	-	100	3	0	-	3
PEC363-CAI	Programme Elective Course Lab 3	Programme Elective-II Laboratory	-	-	2	-	-	25	-	25	50	0	0	1	1
MDM371-CAI	Multidisciplinary Minor	Robotics and Automation	-	1	2	-	-	25	25	-	50	0	1	1	2
VSE372-CAI	Vocational and Skill Enhancement Course	Solar Technology and Development	-	-	2	-	-	50	-	-	50	0	0	1	1
ELC381-CAI	Experiential Learning Course (Internship/OJT)	Internship	-	-	8	-	-	-	50	-	50	0	0	4	4
Total			**11**	**1**	**20**	**120**	**280**	**150**	**100**	**50**	**700**	**11**	**1**	**10**	**22**

Programme Elective II Courses	
PEC361ACAI	Blockchain
PEC361BCAI	Big Data engineering
PEC361CCAI	High Performance Computing
PEC361DCAI	AR/VR

Programme Elective II Course	
PEC362ACAI	Ethical Hacking
PEC362BCAI	Quantum AI
PEC362CCAI	Prompt Engineering
PEC362DCAI	UI/UX

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC-301-CAI: Theory of Computations		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

,

Prerequisite Courses, if any:

1. Data Structure
2. Computer Architecture & Organization.

Course Objectives: The course aims to:

1. To develop the ability to design and analyze Finite Automata and Regular Expressions for recognizing regular languages.
2. To construct and simplify Context-Free Grammars for representing context-free languages.
3. To design and analyze Pushdown Automata (PDA) for solving context-free language problems.
4. To apply Linear Bounded Automata (LBA) for modeling and solving context-sensitive language problems.
5. To design Turing Machines and understand computability, decidability, and complexity concepts for analyzing the solvability and limitations of computational problems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Design and analyse DFA, NFA, ε-NFA, and Regular Expressions for modelling regular languages.
- **CO2:** Construct, simplify, and convert Context-Free Grammars and remove ambiguity using normal forms.
- **CO3:** Design and implement Pushdown Automata (PDA) for recognizing context-free languages and establish equivalence with CFG.
- **CO4:** Apply Linear Bounded Automata (LBA) to model and solve context-sensitive language problems.
- **CO5:** Design and analyse Turing Machines to simulate computational processes and solve general computational problems.

Course Contents

Unit I - Finite Automata and Regular Expressions (09 Hours)

Finite Automata (FA): Symbols, Strings, Language, Formal Languages, FSM definition and notations, DFA, NFA, ε-NFA, Conversions: ε-NFA → NFA, NFA → DFA, Minimization and Equivalence of FA

Finite State Machines with Output: Moore and Mealy machines, Inter-conversion between Moore & Mealy models

Definitions and identities of Regular Expressions (RE): RE operators and equivalence of Regular Expressions & Regular Languages, Conversion: RE → FA (direct method), FA → RE (Arden's theorem)

Properties of Regular Languages**:** Pumping lemma for RLs, Closure properties.

Case Study: Design of Finite Automata for Binary Strings Ending with "01"

Unit II Context Free Grammars and Context Free Languages (09 Hours)

Grammar and representation Chomsky hierarchy and Regular Grammar transformations, Context-Free Grammars (CFG): Ambiguous vs unambiguous grammar, Removal of ambiguity. Simplification of CFG. Normal forms: Chomsky Normal Form (CNF), Greibach Normal Form (GNF).

Context Free Languages: Definition, Regular grammar definition, Left linear, Right linear grammar, Interconversion between left linear and right linear regular grammar.

Case Study: Design of Context-Free Grammar for Arithmetic Expression Parsing

Unit III Push Down Automata 09 Hours)

Formal definition, Pushdown automata (PDA), , Transition diagrams and tables, Instantaneous descriptions, Deterministic Pushdown automata (DPDA) – definition, Nondeterministic Pushdown automata (NPDA) - definition, Deterministic vs non-deterministic PDA ,relative powers of DPDA and NPDA. Acceptance by final state vs empty stack.

Case Study: Design and Implementation of Pushdown Automata for Balanced Parentheses Language

Unit IV Linear Bounded Automata (09 Hours)

Definition and formal model of Linear Bounded Automata (LBA), Structure and working of LBA, Configuration and transition representation, Acceptance criteria of LBA, LBA as a restricted form of Turing Machine. Relationship between LBA and Context-Sensitive Languages (CSL), Equivalence of LBA and Type-1 Grammars, Computational power and limitations of LBA, Deterministic vs Non-deterministic LBA

Case Study: Applications and examples of problems solved using LBA

Unit V Turing Machines and Undecidability (09 Hours)

Definition, Computing with Turing machine, Extensions of Turing machines, Random access Turing machines, Non-deterministic Turing machines, The Church's Turing hypothesis, Universal Turing machines, The Halting problem, Unsolvable problems about Turing machines. Recursive and Recursively Enumerable languages. Decidable and undecidable problems

Case Study: Applications of decidability theory

Learning Resources

Text Books:

1. Vivek Kulkarni, *Theory of Computation*, Oxford University Press, First Edition, 2013.
2. K. L. P. Mishra and N. Chandrasekaran, *Theory of Computer Science: Automata, Languages and Computation*, PHI Learning Pvt. Ltd., Third Edition.
3. O. G. Kakde, *Theory of Computation*, Laxmi Publications, First Edition, 2008.

Reference Books:

1. Daniel I. A. Cohen, *Introduction to Computer Theory*, John Wiley & Sons, First Edition, 1986.
2. John C. Martin, *Introduction to Languages and the Theory of Computation*, McGraw-Hill Education, Fourth Edition, 2010.
3. K. Krithivasan and R. Rama, *Introduction to Formal Languages, Automata Theory and Computation*, Pearson Education, First Edition, 2009.
4. C. H. Papadimitriou and H. R. Lewis, *Elements of the Theory of Computation*, Prentice Hall of India, Second Edition, 1997.
5. E. V. Krishnamurthy, *Introductory Theory of Computer Science*, Springer-Verlag, First Edition, 1985.

MOOC / NPTEL/YouTube Links:-

1. https://onlinecourses.nptel.ac.in/noc21_cs83
2. https://onlinecourses.nptel.ac.in/noc21_cs19
3. https://www.coursera.org/learn/cs-algorithms-theory-machines

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC-302-CAI: Machine Learning		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	**03**	**CCE : 30 Marks** **End-Semester: 70 Marks**

Prerequisite Courses:

1. Knowledge of Python programming fundamentals.
2. Understanding of Probability, Statistics, and Linear Algebra.

Course Objectives: The course aims to:

1. Understand core concepts of supervised and unsupervised machine learning
2. Design, train, and evaluate ML models for real-world problems
3. Develop ability to select appropriate algorithms and performance metrics
4. Gain hands-on experience with modern ML tools and workflows

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Understand basic concepts of machine learning and its types.
- CO2: Apply regression algorithms for prediction problems.
- CO3: Implement classification algorithms and evaluate their performance.
- CO4: Apply clustering techniques to analyze unlabeled data.
- CO5: Use dimensionality reduction and optimization techniques to improve models.

Course Contents

Unit I - Introduction to Machine Learning (09 Hours)

Machine learning overview, machine learning vs artificial intelligence vs data science, types of machine learning, supervised learning concepts, unsupervised learning concepts, machine learning workflow, training and testing process, bias and variance, overfitting and underfitting, model generalization, introduction to scikit-learn library.

Case Study: To analyze a machine learning system for predicting student academic performance using historical data.

Unit II - Supervised Learning – Regression (09 Hours)

Supervised learning framework, regression problems, simple linear regression, multiple linear regression, polynomial regression, regularization techniques, ridge regression, lasso regression, elastic net, loss

functions for regression, model evaluation metrics, mean absolute error, mean squared error, root mean squared error, R-squared score.

Case Study: To design a regression-based machine learning model for predicting continuous values such as house. Prices or crop yield.

Unit III - Supervised Learning – Classification (09 Hours)

Classification problems, logistic regression, k-nearest neighbors algorithm, decision tree classifier, ensemble learning concepts, random forest classifier, gradient boosting overview, hyperparameters in classification models, performance evaluation metrics, confusion matrix, accuracy, precision, recall, F1-score, ROC curve, AUC score.

Case Study: To develop a machine learning classification system for categorizing data into predefined classes.

Unit IV- Unsupervised Learning – Clustering (09 Hours)

Unsupervised learning framework, clustering techniques, k-means clustering algorithm, hierarchical clustering, agglomerative clustering, DBSCAN algorithm, distance measures, cluster validation techniques, elbow method, silhouette score, clustering applications, customer segmentation, pattern discovery, anomaly detection.

Case Study: To apply clustering techniques for grouping unlabeled data based on similarity.

Unit V Unsupervised Learning – Dimensionality Reduction and Model Optimization (09 Hours)

Curse of dimensionality, feature space reduction, principal component analysis, eigenvalues and eigenvectors overview, variance preservation, t-SNE overview, feature selection vs feature extraction, model tuning concepts, hyper parameter optimization, grid search, random search, cross-validation techniques, improving model performance. Overview and Scope of reinforcement learning.

Case Study: To apply dimensionality reduction and optimization techniques to improve machine learning model performance.

Learning Resources

Text Books:

1. rélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”, Publisher- O’Reilly Media, 3rd Edition, 2022, ISBN-13: 978-1098125974, ISBN-10: 1098125975.
2. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, Publisher- MIT Press, Updated Edition, ISBN-13: 978-0262018029, ISBN-10: 0262018020.

Reference Books:

1. Taeho Jo, “Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning”, Publisher- Springer, 1st Edition, 2021, ISBN-13: 978-3030659004, ISBN-10: 303065900X.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Publisher- Springer, 2nd Edition, 2009, ISBN-13: 978-0387848570, ISBN-10: 0387848576.
3. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Publisher- Springer, 1st Edition, 2006, ISBN-13: 978-0387310732, ISBN-10: 0387310738.
4. Andriy Burkov, “The Hundred-Page Machine Learning Book”, Publisher- Andriy Burkov, Latest Edition, ISBN-13: 978-1999579500, ISBN-10: 199957950X.

MOOC / NPTEL/YouTube Links:-

1. NPTEL Online Course on Machine Learning by Prof. Balaraman Ravindran / Prof. P. J. Narayanan.
2. scikit-learn Official Documentation – https://scikit-learn.org
3. MIT OpenCourseWare – Machine Learning Courses.
4. Coursera – Machine Learning and Supervised Learning Modules.
5. Kaggle Learn – Machine Learning and Model Evaluation Tutorials.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC- 303- CAI: Design and Analysis of Algorithm		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

Prerequisite Courses, if any

1. Programming Fundamentals.
2. Data Structures
3. Discrete Mathematics (Basics).
4. Mathematical Foundations

Companion Course if any: NA

Course Objectives: The course aims to:

1. Understand fundamental concepts and principles of algorithm design.
2. Analyze the efficiency of algorithms using time and space complexity measures.
3. Apply suitable algorithmic paradigms to solve computational problems.
4. Design efficient algorithms for optimization and graph-based problems.
5. Understand computational complexity and limitations of algorithmic solutions.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain fundamental concepts of algorithm design and analysis.
- CO2: Analyze time and space complexity of algorithms using asymptotic notations.
- CO3: Apply Divide and Conquer and Greedy techniques to solve problems.
- CO4: Design and analyze Dynamic Programming based solutions.
- CO5: Implement graph algorithms and explain NP-Complete problems.

Course Content

Unit I - Introduction to Algorithms & Complexity Analysis (9 Hours)

Introduction, Definition and characteristics of algorithms, Algorithm designs vs algorithm analysis, Growth of functions, Asymptotic notations: Big-O, Big-Ω, Big-Θ, and Best, average and worst case analysis, Time complexity analysis, Space complexity analysis, Mathematical analysis with examples.
Case Study: Product Lookup in an Online Store.

Unit II - Divide and Conquer Strategy (9 Hours)

Introduction, Divide and Conquer paradigm, Recurrence relations Methods to solve recurrences: Substitution method, Recursion tree method, Master's theorem, Algorithms: Binary Search, Merge Sort, Quick Sort, Strassen's Matrix Multiplication.

Case Study: Image Processing in Medical Diagnostics.

Unit III - Greedy Algorithms (9 Hours)

Introduction: Greedy strategy and characteristics, Elements of greedy algorithms Applications: Activity Selection Problem, Fractional Knapsack, Job Sequencing with Deadlines, Minimum Spanning Tree: Prim's Algorithm, Kruskal's Algorithm, Huffman Coding, Huffman Coding, Comparison with other paradigms.

Case Study: Job Scheduling in a Manufacturing Plant.

Unit IV - Dynamic Programming (09 Hours)

Introduction to Dynamic Programming, Optimal substructure and overlapping sub problems, Top-down and bottom-up approaches, Applications: Fibonacci sequence, Matrix Chain Multiplication, Longest Common Subsequence (LCS), 0/1 Knapsack Problem, Optimal Binary Search Tree, Performance comparison with Greedy and Divide & Conquer.

Case Study: Cloud Storage Optimization.

Unit V - Graph Algorithms & NP-Completeness(09 Hours)

Graph Algorithms: Graph representations, Breadth First Search (BFS), Depth First Search (DFS). Shortest path algorithms: Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm NP-Completeness: Introduction to computational complexity, Classes P, NP, NP-Hard and NP-Complete problems, Polynomial-time reduction, Examples: Travelling Salesman Problem, Clique Problem, Vertex Cover Problem.

Case Study: Cyber security Attack Simulation Using Vertex Cover.

Learning Resources:

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein "Introduction to Algorithms", The MIT Press, Cambridge, Massachusetts, 3rd Edition (2009, MIT Press), ISBN - 978-0-262-03384-8.
2. Robert Sedgewick, Kevin Wayne , "Algorithms", 4th Edition (2011), Addison-Wesley Professional, ISBN - 978-0-321-57351-3.

Reference Books:

1. Yashavant Kanetkar, “Let Us C”, 8th Edition, BPB Publications, ISBN: 9788183331777.
2. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition, Pearson Education, ISBN: 978-8177583588.
3. Aaron M. Tenenbaum, “Data Structures Using C”, 2nd Edition, Pearson Education, ISBN: 97881317114.

MOOC / NPTEL/YouTube Links:

1. https://onlinecourses.swayam2.ac.in/nou26_cs15/preview
2. https://onlinecourses.swayam2.ac.in/nou26_cs14/preview
3. https://www.youtube.com/watch?v=0IAPZzGSbME&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O&index=1

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC- 304- CAI: Design and Analysis of Algorithms Laboratory ,		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Term Work:** 25 Marks **Oral:** 25 Marks

Companion Course if any: Data Structures

Course Objectives: The course aims to:

1. Understand algorithmic paradigms (recursion, iteration, greedy, divide-and-conquer, dynamic programming, and graph algorithms) and their role in solving real-world problems.
2. Apply data structures and algorithms to practical scenarios such as banking, e-commerce, healthcare, telecom, cloud computing, and cyber security.
3. Analyze time and space complexity to evaluate efficiency and scalability of solutions.
4. Compare multiple algorithmic approaches (e.g., recursive vs. iterative, naive vs. optimized, exact vs. approximate) for the same problem.
5. Develop problem-solving skills by modeling case studies into computational problems and implementing them in programming languages.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Implement recursive and iterative algorithms for mathematical and application-based problems (e.g., Fibonacci in banking, product lookup in e-commerce).
- CO2: Design and analyze sorting and searching techniques for large datasets, demonstrating efficiency differences in best, average, and worst cases.
- CO3: Apply greedy algorithms (Job Sequencing, Activity Selection, Fractional Knapsack) to optimize scheduling, resource allocation, and storage utilization.
- CO4: Apply dynamic programming techniques (Fibonacci DP, LCS) to predictive analytics and DNA sequence analysis, demonstrating efficiency improvements.
- CO5: Explore NP-Complete problems (Vertex Cover) and approximation methods in cybersecurity contexts, understanding limitations of exact solutions.

Course Contents

Guidelines for Instructor's Manual

The instructor's manual/Lab Manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of course, conduction and Assessment guidelines, topics un- der consideration-concept, objectives, outcomes, set of typical

applications/assignments/guidelines, references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Students programs maintained on cloud or college server by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purpose.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be done based on overall performance and Laborotory assignments performance of student. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall as- assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute them among batches of students.

It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged. In addition to these, instructors may assign one real life application in the form of a mini-project based on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to respective branch beyond the scope of the syllabus.

Set of suggested assignment lists is provided in groups- A, B, C, D, and E. Each student must perform at least 9 assignments (All assignments for group A are compulsory, 2 from group B, 2 from group C, 1 from group D and 1 from group E).

- All assignments should be implemented in C/C++ language.
- Operating System Recommended: 64-bit Open Source Linux or its derivatives
- Programming Tools Recommended: Open Source C compiler such as GCC/G++.
- Development environments or text editors like Visual Studio Code, GeanV, Code:Blocks, or terminal-based editors like Vim or Emacs.

Guidelines for Practical Examinations

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the students for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Learning Resources

Virtual Laboratory:

1. https://ds2-iiith.vlabs.ac.in/List%20of%20experiments.html

Suggested List of Laboratory Experiments/Assignments

<table>
<tr><th>Sr. No.</th><th>Group A – Assignments (Any Two)</th></tr>
<tr><td>1</td><td>Title - Fibonacci Numbers in Application Scenario
A banking system wants to design a reward program where the reward points for loyal customers follow the Fibonacci sequence (1, 1, 2, 3, 5, 8, …). For example, the nth year of loyalty gives the customer Fibonacci(n) reward points. Since the system must handle millions of customers, efficiency of the algorithm is critical.
Design and implement a program to compute the nth Fibonacci number using:
1. Recursive approach (straightforward but inefficient).
2. Non-recursive approach (iterative, efficient).
The program should:
• Accept n as input (year of loyalty).
• Compute Fibonacci(n) using both recursive and iterative methods.
• Compare execution time and step count for both approaches.
• Demonstrate how algorithm design impacts efficiency in real-world applications like reward systems.
• Display Comparison Table
<table>
<tr><td>Method</td><td>Step Count (approx.)</td><td>Time Complexity</td><td>Space Complexity</td></tr>
<tr><td>Recursive</td><td></td><td></td><td></td></tr>
<tr><td>Iterative</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Title - Product Lookup in an Online Store
An online shopping platform maintains a catalog of thousands of products identified by unique product IDs. Customers frequently search for products, and the efficiency of the search algorithm directly impacts user experience, especially during peak sale events.
You are required to design and implement a program that simulates product lookup using four different search techniques:
1. Recursive Linear Search – Sequentially checking each product until the desired item is found.</td></tr>
</table>

2. Iterative Linear Search – Using a loop to traverse the product list.
3. Recursive Binary Search – Efficiently locating the product in a sorted catalog by dividing the search interval recursively.
4. Iterative Binary Search – Using a loop to perform binary search on a sorted catalog.

The program should:

- Accept a dataset of product IDs (simulated as integers or strings).
- Implement all four search techniques.
- Compare their performance by measuring execution time and step count for different dataset sizes (e.g., 100, 1,000, 10,000, 100,000 items).
- Demonstrate best-case, average-case, and worst-case scenarios for each algorithm.
- Present results in tabular or graphical form to highlight differences in asymptotic growth.

Algorithm	Best Case Scenario	Average Case Scenario	Worst Case Scenario	Time Complexity	Space Complexity
Recursive Linear Search					
Iterative Linear Search					
Recursive Binary Search					
Iterative Binary Search					

3 Title - University Student Database Management

A university maintains records of thousands of students. Each student has a USN, Name, Branch, and Phone number. The administration system must store these records and allow efficient retrieval and display. Since the system may run on limited hardware (like departmental servers), analysing time and space complexity of the solution is essential to ensure scalability.

Problem Statement

Design a Java program that:

1. Defines a student class with attributes: USN, Name, Branch, Phone.
2. Creates n Student objects and stores them in an array.
3. Prints all student details with suitable headings.
4. Analyses time complexity and space complexity of the program.

Group B – Assignments (Any two)

4 Title - Online Retail Inventory Management

An online retail company manages a large inventory of products. During festive sales, millions of product records (IDs, names, prices) need to be sorted quickly for efficient display on the website. The company initially uses simple sorting algorithms like Bubble Sort and Selection Sort, but wants to analyze their performance to understand scalability issues.

Problem Statement

Design and implement a program to compare the performance of Bubble Sort and Selection Sort algorithms on product inventory data.

	The program should: • Accept datasets of varying sizes (e.g., 100, 1,000, 5,000, 10,000 items). • Implement Bubble Sort and Selection Sort to arrange product IDs or prices. • Measure and record execution time for each algorithm across different dataset sizes. • Plot execution time vs. input size to demonstrate asymptotic growth. • Conclude which algorithm is more efficient and why both are unsuitable for very large datasets compared to advanced algorithms (like Merge Sort or Quick Sort).
5	Title - Image Processing in Medical Diagnostics A hospital uses advanced imaging systems (like CT scans and MRI) that rely heavily on matrix operations for image reconstruction. Since these systems process very large matrices, the efficiency of matrix multiplication directly impacts the speed of diagnosis. The hospital wants to evaluate whether Strassen's Matrix Multiplication can provide performance improvements compared to the standard matrix multiplication algorithm. Problem Statement Design and implement a program to multiply two square matrices using: 1. Standard Matrix Multiplication Algorithm (triple nested loop). 2. Strassen's Matrix Multiplication Algorithm (divide-and-conquer approach). The program should: • Accept two matrices of size n×n. • Implement both algorithms to compute the product. • Measure execution time for different matrix sizes (e.g., 64×64, 128×128, 256×256, 512×512). • Compare efficiency in terms of time complexity and step count. • Demonstrate how Strassen's algorithm reduces the number of multiplications compared to the standard method.
6	Title - Sorting Customer Records in an E-Commerce Platform An e-commerce company needs to sort millions of customer records daily for tasks such as: • Order processing (sorting by order ID). • Recommendation systems (sorting by purchase frequency). • Analytics dashboards (sorting by transaction amount). The choice of sorting algorithm directly impacts system performance and scalability. Problem Statement Design and implement a program to sort customer records using Merge Sort and Quick Sort, then compare their efficiency. The program should: 1. Accept an array of customer records (IDs, names, or transaction values). 2. Implement Merge Sort and Quick Sort. 3. Measure execution time for different dataset sizes (e.g., 1,000, 10,000, 100,000 records). 4. Compare performance in terms of time complexity and space complexity.

	5. Conclude which algorithm is more suitable for large-scale e-commerce systems.
Group C- Assignments (Any Four)	
7	Title - Job Scheduling in a Manufacturing Plant A manufacturing plant receives multiple production jobs daily. Each job has: • A deadline (by when it must be completed). • A profit (earned if the job is completed before or on its deadline). The plant has limited machines and can only process one job at a time. Management wants to schedule jobs in such a way that the total profit is maximized while ensuring deadlines are respected. Problem Statement Design and implement a program to schedule jobs in a manufacturing plant using the Job Sequencing with Deadlines algorithm. The program should: 1. Accept a list of jobs, each with: o Job ID o Deadline o Profit 2. Schedule jobs such that: o No two jobs overlap in the same time slot. o Jobs are completed before their deadlines. o The total profit is maximized. 3. Display the scheduled jobs and the total profit earned. 4. Compare the efficiency of this greedy approach with a naive scheduling method (e.g., sequential scheduling without optimization).
8	Title - Telecom Network Design A telecom company wants to establish a communication network connecting multiple cities. Each possible connection (edge) between two cities has an associated cost (e.g., laying fiber optic cables, leasing lines). The company's goal is to design a network that: • Connects all cities. • Uses the minimum total cost. • Avoids cycles (no redundant connections). This problem can be modeled as a graph, where: • Nodes (vertices) represent cities. • Edges represent possible connections with associated costs. To achieve the minimum-cost network, the company can apply Minimum Spanning Tree (MST) algorithms: • Prim's Algorithm (grows the MST by adding the cheapest edge from the tree to a new vertex). • Kruskal's Algorithm (sorts edges by weight and adds them one by one, avoiding cycles). Problem Statement Design and implement a program to construct a minimum-cost communication network using both Prim's and Kruskal's algorithms. The program should: • Accept input as a weighted graph (cities and connection costs).

	• Apply Prim's Algorithm to generate the MST. • Apply Kruskal's Algorithm to generate the MST. • Compare the total cost and structure of the MST obtained by both algorithms. • Demonstrate efficiency differences for sparse vs. dense graphs.
9	Title - Airline Ticket Booking System An airline company operates multiple flights daily. Each flight has a start time (departure) and an end time (arrival). The booking system must schedule flights in such a way that: • No two flights overlap on the same runway or gate. • The maximum number of flights can be accommodated. This is analogous to the Activity Selection Problem, where each activity (flight) has a start and finish time, and the goal is to select the maximum number of non-overlapping activities. Problem Statement Design and implement a program to schedule flights using the Activity Selection Problem approach. The program should: 1. Accept a list of flights with their departure (start time) and arrival (end time). 2. Sort flights by their end times. 3. Apply the Greedy Algorithm to select flights such that: o No two flights overlap. o The maximum number of flights are scheduled. 4. Display the scheduled flights and the total number of flights accommodated. 5. Compare with a naive scheduling approach (e.g., selecting flights in input order) to highlight efficiency.
10	Title - Data Compression in Messaging Apps Modern messaging apps (WhatsApp, Telegram, Signal) transmit millions of chat messages daily. To reduce bandwidth usage and storage requirements, these apps rely on data compression techniques. Huffman Coding is one such method that compresses text by assigning shorter codes to frequently used characters and longer codes to less frequent ones. The company wants to evaluate Huffman Coding against other compression methods (e.g., Run-Length Encoding, Fixed-Length Encoding) to determine efficiency in compressing chat messages. Problem Statement Design and implement a program to compress chat messages using Huffman Coding and compare its efficiency with other compression techniques. The program should: 1. Accept a chat message (string input). 2. Implement Huffman Coding to generate prefix codes and compress the message. 3. Implement at least one other compression method (e.g., Run-Length Encoding or Fixed-Length Encoding). 4. Calculate and compare: o Original message size (in bits/characters). o Compressed message size. o Compression ratio. 5. Demonstrate efficiency differences for short vs. long messages, and

	messages with varying character frequencies.
11	Title - Cloud Storage Optimization A cloud service provider offers limited storage capacity to users. Different files (documents, images, videos, backups) have varying sizes and importance values (e.g., revenue potential, priority level, or frequency of access). The provider wants to maximize storage utilization by selecting files that yield the highest value per unit of storage. Since files can be split into smaller chunks (e.g., video segments, compressed blocks), the Fractional Knapsack Algorithm can be applied to achieve optimal utilization. Problem Statement Design and implement a program to optimize cloud storage using the Fractional Knapsack Algorithm. The program should: 1. Accept a list of files, each with: o File ID o Size (weight) o Value (priority/importance) 2. Accept the total storage capacity of the cloud system. 3. Sort files by value-to-size ratio in descending order. 4. Select files (or fractions of files) until the storage capacity is filled. 5. Display the selected files, fractions (if any), and the maximum total value achieved. 6. Compare with the 0/1 Knapsack approach (where files cannot be split) to highlight efficiency differences.
Group D- Assignments (Any one)	
12	Title - Predictive Analytics in Finance Financial analysts often use mathematical models to identify patterns in stock market trends. One simplified approach is to model growth and decline cycles using the Fibonacci sequence, which is widely referenced in technical analysis (e.g., Fibonacci retracement levels in trading). However, computing Fibonacci numbers using naive recursion is inefficient for large values of n. In predictive analytics, where large datasets and real-time calculations are required, efficiency is critical. Dynamic Programming (DP) provides an optimized way to compute Fibonacci numbers by storing intermediate results and avoiding redundant calculations. Problem Statement Design and implement a program to model stock trend prediction using the Fibonacci sequence with Dynamic Programming. The program should: 1. Accept an integer n representing the number of trading days or cycles. 2. Compute Fibonacci numbers using: o Naive Recursion o Dynamic Programming (Memoization/Tabulation) 3. Compare execution time and step count for both approaches. 4. Demonstrate how DP improves efficiency for large n.

	5. Relate Fibonacci growth patterns to simplified stock trend prediction (e.g., retracement levels, support/resistance modeling).
13	Title - DNA Sequence Analysis Genetic researchers often need to measure the similarity between DNA sequences to identify evolutionary relationships, detect mutations, or compare genetic markers. DNA sequences are represented as strings composed of nucleotides: A (adenine), C (cytosine), G (guanine), T (thymine). One way to measure similarity is to compute the Longest Common Subsequence (LCS) between two DNA strings. The LCS represents the longest sequence of nucleotides that appear in both DNA strings in the same order (not necessarily contiguous). A longer LCS indicates higher genetic similarity. Problem Statement Design and implement a program to detect genetic similarity between two DNA sequences using the Longest Common Subsequence (LCS) algorithm. The program should: 1. Accept two DNA sequences as input strings. 2. Apply Dynamic Programming to compute the LCS length and sequence. 3. Display: o The LCS length. o The LCS sequence. o A similarity score (e.g., LCS length ÷ length of longer sequence). 4. Compare efficiency with a naive recursive approach for small sequences. 5. Demonstrate how LCS can be used in genetic similarity detection.
Group E- Assignments (Any one)	
14	Title - Ride-Sharing App Shortest Path Computation A ride-sharing company (like Uber or Ola) needs to compute the shortest travel routes between pickup and drop-off locations across a city. The road network can be modelled as a weighted graph, where: • Vertices (nodes): Locations (junctions, pickup points, drop-off points). • Edges: Roads connecting locations. • Weights: Travel time or distance on each road. Efficient shortest path computation is critical for: • Minimizing travel time for passengers. • Reducing fuel costs for drivers. • Improving overall system efficiency. Problem Statement Design and implement a program to compute shortest travel routes in a ride-sharing app using: 1. Dijkstra's Algorithm – for single-source shortest paths with non-negative weights. 2. Bellman-Ford Algorithm – for single-source shortest paths with possible negative weights (e.g., toll discounts or incentives). 3. Floyd-Warshall Algorithm – for all-pairs shortest paths (useful for precomputing routes between all locations). The program should: • Accept a weighted graph representing the city's road network. • Compute shortest paths using each algorithm. • Compare execution time and memory usage for different graph sizes

	(small, medium, large). • Display the optimal route and total travel cost/time.
15	Title – Cyber security Attack Simulation Using Vertex Cover In modern computer networks, attackers often exploit vulnerabilities in nodes (servers, routers, endpoints) to gain unauthorized access. To defend against such attacks, cybersecurity teams must identify a minimal set of critical nodes to monitor or secure. This problem can be modelled as a graph: • Vertices (nodes): Devices or servers in the network. • Edges: Communication links or potential attack paths. • Goal: Find the smallest set of vertices such that every edge in the graph is incident to at least one chosen vertex. This is the Vertex Cover Problem, which is NP-Complete. While finding the exact minimum cover is computationally expensive for large networks, approximation algorithms and heuristics can provide practical solutions for intrusion detection. Problem Statement Design and implement a program to simulate intrusion detection in a network using the Vertex Cover Problem. The program should: 1. Accept input as a graph representing the network (nodes and communication links). 2. Apply algorithms to compute: o Exact Vertex Cover (for small networks, using brute force or backtracking). o Approximation Algorithm (greedy approach for larger networks). 3. Identify the minimal set of nodes to monitor/secure. 4. Compare efficiency of exact vs. approximate solutions. 5. Demonstrate how NP-Complete problems arise in cybersecurity defnce.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC-304- CAI: Machine Learning Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	**Term Work** : 25 Marks **Practical :** 25 Marks

Companion Course if any: Machine Learning Theory

Course Objectives: The course aims to:

1. Provide hands-on experience in implementing machine learning algorithms using Python.
2. Understand and apply supervised and unsupervised learning techniques for real-world datasets.
3. Develop skills for data preprocessing, feature selection, and model building.
4. Evaluate machine learning models using appropriate performance metrics.
5. Enhance problem-solving ability through application-oriented machine learning experiments.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Implement basic machine learning programs using Python and scikit-learn libraries.
- CO2: Apply regression and classification algorithms to solve prediction and classification problems.
- CO3: Analyze model performance using suitable evaluation metrics.
- CO4: Apply clustering and dimensionality reduction techniques for data analysis.
- CO5: Design and optimize machine learning models for real-world applications.

Guidelines for Instructor's Manual

The instructor's manual/Lab Manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about university/program/ institute/ department/foreword/ preface), curriculum of course, conduction and Assessment guidelines, topics unreconsideration-concept, objectives, outcomes, set of typical applications/assignments/guidelines, references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Students programs maintained on cloud or college server by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purpose.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute them among batches of students.

It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged. The instructor may assign a mini-project or case study based on real-life Machine Learning applications to strengthen practical understanding beyond routine experiments. Multiple datasets or problem variants may be used to ensure originality and reduce repetition among students.

Recommended Laboratory Environment:

Operating System: 64-bit Open-Source Linux or equivalent

Tools: Python 3.x, Jupiter Notebook / Google Collab, NumPy, Pandas, Matplotlib, scikit-learn

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Learning Resources

Virtual Laboratory:

1. NPTEL Online Course on Machine Learning by Prof. Balaraman Ravindran / Prof. P. J. Narayanan.

2. scikit-learn Official Documentation – https://scikit-learn.org
3. MIT Open Courseware – Machine Learning Courses.
4. Coursera – Machine Learning and Supervised Learning Modules.
5. Kaggle Learn – Machine Learning and Model Evaluation Tutorials.

Suggested List of Laboratory Experiments/Assignments

Sr. No.	Practical List
1	Study and implementation of Python libraries used in Machine Learning such as NumPy, Pandas, and Matplotlib for data analysis and visualization.
2	Implementation of the complete Machine Learning workflow including data loading, preprocessing, model training, testing, and evaluation using scikit-learn.
3	Analysis of bias, variance, overfitting, and underfitting using polynomial regression on a given dataset.
4	Implementation of Simple Linear Regression for predicting a continuous variable and evaluation using appropriate error metrics.
5	Implementation of Multiple Linear Regression using multiple features and evaluation using MAE, MSE, RMSE, and R-squared score.
6	Implementation and comparison of regularization techniques such as Ridge, Lasso, and Elastic Net regression for improving model generalization.
7	Implementation of Logistic Regression for binary classification and performance evaluation using confusion matrix and accuracy score.
8	Implementation of the K-Nearest Neighbours (KNN) classification algorithm and analysis of the impact of different values of K on model accuracy.
9	Implementation of a Decision Tree Classifier and study of the effect of tree depth on classification performance.
10	Implementation of an ensemble learning model using Random Forest and comparison with Decision Tree classifier.
11	Implementation of K-Means clustering on an unlabelled dataset and determination of the optimal number of clusters using the Elbow method.
12	Implementation of Hierarchical Clustering using the agglomerative approach and visualization of clusters using dendrograms.
13	Implementation of DBSCAN clustering for identification of clusters and detection of outliers in a dataset.
14	Application of Principal Component Analysis (PCA) for dimensionality reduction and analysis of variance preservation.
15	Performance of hyperparameter tuning using Grid Search and Cross-Validation for improving machine learning model performance.
16	Mini Project 1: Design and implementation of a Machine Learning system for predicting student academic performance using supervised learning techniques. OR Mini Project 2: Design and development of a customer segmentation system using clustering techniques for analyzing customer behavior and business insights.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC321ACAI: Cyber security		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Computer Networks, Operating Systems, Cyber Security Fundamentals

Course Objectives: The course aims to:

1. Equip students with core knowledge of ethical hacking, its application scope, relevant legal frameworks, and the evolving cyber threat landscape
2. Develop understanding of reconnaissance techniques and attack surface analysis using OSINT and exposure assessment methods.
3. Build knowledge of common vulnerabilities, exploitation concepts, and attack techniques used in real-world scenarios.
4. Analyze identity-based attacks, social engineering methods, and post-compromise activities in modern systems.
5. Explore security challenges in modern environments including web, API, cloud, and AI-driven systems along with ethical responsibilities

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply knowledge of ethical hacking frameworks, lifecycle stages, and professional guide-lines to assess modern threat landscapes.
- **CO2 :** Utilize reconnaissance methodologies and tools to identify exposed assets and map attack surfaces effectively
- **CO3:** Identify common vulnerabilities and demonstrate basic exploitation concepts in controlled environments.
- **CO4:** Analyze identity-based attacks, credential misuse, and post-compromise activities to understand attacker behavior.
- **CO5:** Evaluate security risks in web, API, cloud, and AI systems and understand ethical responsibilities in cyber security practices.

Unit I - Ethical Hacking Approach & Modern Threat Landscape (09 Hours)

Ethical Hacking – definition, purpose, and scope; Difference between penetration testing, vulnerability assessment, and red teaming; Ethical hacking lifecycle – reconnaissance, scanning, exploitation, post-compromise, reporting; Rules of Engagement – authorization, scope, ethical boundaries; At-tacker mindset – targeting weakest link, identity-focused attacks, attack chaining; Overview of MITRE ATT&CK – understanding attacker tactics. Modern Threat Landscape: Modern attack trends – identity-based attacks,

cloud misconfiguration, API abuse, supply chain attacks, Phishing evolution – AiTM phishing, session/token theft, Zero Trust Security Model and identity-centric security principles ,AI-augmented attacks – LLM prompt injection, adversarial ML, deepfake-enabled social engineering.

Case Study: Modern identity-based breach involving credential misuse, AiTM phishing, and unauthorized cloud access (e.g., analysis of a real-world Microsoft 365 compromise scenario).

Unit II Reconnaissance & Vulnerability Discovery (09 Hours)

Reconnaissance – definition, importance, passive vs active; OSINT – public sources; Digital foot-print and attack surface; DNS and domain intelligence; Service exposure; Vulnerability scanning con-cept; WHOIS, ASN lookups, BGP peering analysis, Google Dorking and advanced search operators, GitHub/GitLab OSINT – exposed secrets, API keys, credentials in public repos

Scanning & Enumeration: Network scanning – Nmap techniques: SYN, UDP, version detection, OS fingerprinting ,Service enumeration :banner grabbing, version identification, Vulnerability scanning concept – Nessus, OpenVAS, Cloud attack surface :exposed S3 buckets, Azure blobs, GCP storage misconfigurations ,Internet-wide exposure awareness : Shodan queries, Censys facets, exposed IoT devices ,Attack surface management (ASM) platforms .

Case Study: Data exposure due to misconfigured cloud storage: full reconnaissance chain from OSINT to bucket discovery, impact analysis, and responsible disclosure process

Unit III Vulnerabilities & Exploitation Concepts (09 Hours)

Vulnerabilities : types; Difference between vulnerability and exploit; CVE, CVSS scoring, NVD, Exploit-DB, VulnHub, Metasploit Framework : architecture, modules Web vulnerabilities – OWASP Top 10 (2021 & 2025 updates) – overview of each category ,broken access control, authentication weaknesses; Common attack concepts – SQL Injection, XSS; Password attacks; Credential attacks; Exploitation concept.

Network & System Attacks: Password attacks – brute force, dictionary, credential stuffing, rainbow tables

Network sniffing: ARP poisoning, passive capture (Wireshark), Wireless attacks: WPA2 cracking, Evil Twin, Buffer overflow concept – stack vs. heap

Emerging: IoT & OT Attack Surface: IoT attack vectors, OT/ICS/SCADA security fundamentals and attack scenarios, Embedded device exploitation awareness (UART, JTAG)

Case Study: Web application compromise due to chained vulnerabilities – SQL injection to authentication bypass, privilege escalation, and data exfiltration in a simulated e-commerce portal.

Unit IV : Identity Attacks & Post-Compromise Overview (09 Hours)

Identity as Attack Surface: Identity-centric security ,Active Directory attacks ,Azure AD / Entra ID attack paths Credential-based attacks ,MFA bypass techniques.

Social Engineering: Social engineering , Phishing campaigns , AiTM (Adversary-in-the-Middle) phishing

Business Email Compromise (BEC), AI-enhanced social engineering – voice cloning, deepfake video calls, Physical security – badge cloning, social engineering for physical access.

Post-Compromise Activities: Post-exploitation objectives, Session/token abuse Data exfiltration methods and C2 (Command & Control) frameworks overview, Covering tracks.

Detection Awareness: SIEM alerts and EDR detection for attacker techniques, Attacker OPSEC – avoiding detection in red team engagements, Threat hunting basics from attacker perspective.

Case Study: Identity-based attack chain: spear phishing ***Ü AiTM credential capture ***Ü MFA bypass ***Ü lateral movement via Pass-the-Hash ***Ü data exfiltration from SharePoint. Analysis includes detection opportunities at each stage.

Unit V - Modern Attack Domains: Web, API, Cloud & AI Threats (09 Hours)

Web Penetration Testing: Web pretesting methodology , Advanced web attacks – HTTP request smuggling, web cache poisoning, clickjacking , Tools: Burp Suite Pro (intruder, repeater, active scan), OWASP ZAP, API Security Testing: REST, GraphQL, gRPC API attack techniques , API fuzzing, rate limit bypass, parameter pollution , JWT attacks Cloud Penetration Testing: Cloud pretesting scope – AWS, Azure, GCP specific attack paths,, Cloud metadata exploitation (SSRF to IMDS), Container security – Docker escape, Kubernetes RBAC misconfigurations Tools: Pacu (AWS), ScoutSuite, Prowler, CloudSploit AI/LLM Security (Emerging Area): AI attack surface – LLM prompt injection, jailbreaking, adversarial inputs Poisoning attacks on ML training data, Securing AI/ML pipelines – model hardening, input validation, output filtering. Ethical & Legal Framework: IT Act 2000 & Amendments – Sections 43, 66, 66C, 66F relevant to hacking

Case Study: Cloud/API breach simulation: SSRF to cloud metadata service ***Ü IAM token extraction ***Ü privilege escalation ***Ü S3 data exfiltration. Full penetration test report writing exercise.

Learning Resources

Text Books:

1. Patrick Engebretson – The Basics of Hacking and Penetration Testing, 3rd Ed., Syngress/Elsevier, (ISBN number 978-0443438868). March 2026
2. Rafay Baloch – Ethical Hacking and Penetration Testing Guide, Auerbach/CRC Press, 2014 (ISBN: 978-1482208917).
3. Matt Walker – CEH Certified Ethical Hacker All-in-One Exam Guide, 5th Ed., McGraw-Hill, 2022 (ISBN: 978-1264274567).
4. Georgia Weidman – Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press, 2014 (ISBN: 978-1593275648

Reference Books:

1. Jon Erickson – Hacking: The Art of Exploitation, 2nd Ed., No Starch Press, 2008 (ISBN: 978-1593271442).
2. 2Peter Kim – The Hacker Playbook 3: Practical Guide to Penetration Testing, Secure Planet, 2018 (ISBN: 978-1980901754). [Red teaming, advanced TTPs]
3. Christopher Hadnagy – Social Engineering: The Science of Human Hacking, 2nd Ed., Wiley, 2018 (ISBN: 978-1119433385).
4. Chris Anley et al. – The Shellcoder's Handbook: Discovering and Exploiting Security Holes, 2nd Ed., Wiley, 2007 (ISBN: 978-0470080238).
5. Michael Sikorski & Andrew Honig – Practical Malware Analysis, No Starch Press, 2012 (ISBN: 978-1593272906).
6. Daniel Miessler – The Real Internet of Things, Leanpub

MOOC / NPTEL/YouTube Links:-

1. Ethical Hacking by Prof. Indranil Sen Gupta, IIT Kharagpur (NPTEL)
2. Cyber Security and Privacy by Prof. Saji K Mathew, IIT Madras (NPTEL)
3. Introduction to Cyber Security – Dr. Jeetendra Pande, Uttarakhand Open University
4. Practical Ethical Hacking – TCM Security (Heath Adams) – Udemy/TCM Academy
5. Web Application Hacking & Bug Bounty – TryHackMe and HackTheBox Academy
6. CEH v12 Preparation – EC-Council Official (paid) / free preview modules

YouTube Links:-

1. The Cyber Mentor (TCM Security) – Full Ethical Hacking Course
2. NetworkChuck – Ethical Hacking Playlist (Beginner-friendly)
3. IppSec – HackTheBox Walkthroughs (Advanced Pentesting)
4. John Hammond – CTF & Malware Analysis
5. David Bombal – Network Hacking & Ethical Hacking
6. LiveOverflow – Binary Exploitation & Web Hacking
7. STOK – Bug Bounty Hunting & Web App Security

E- Books:-

1. Alana Maurushat – Ethical Hacking, University of Ottawa Press (Open Access, free PDF)
2. PCI Security Standards Council – Penetration Testing Guidance (Free PDF from pcisecuritystandards.org)
3. OWASP Testing Guide v4.2 (Free PDF from owasp.org) – comprehensive web pen testing

method-ology

4. NIST SP 800-115 – Technical Guide to Information Security Testing (free from nist.gov)
5. Zaid Sabih – Learn Ethical Hacking from Scratch (Packt eBook / Udemy course)
6. CREST – Penetration Testing & Security Assessment Guides (free PDFs from crest-approved.org)
7. HackTricks Book (free, gitbook.io/hacktricks) – continuously updated red team techniques

Practice Platforms & Lab Resources :-

1. TryHackMe (tryhackme.com) – Guided learning paths for ethical hacking
2. HackTheBox Academy (academy.hackthebox.com) – Professional penetration testing labs
3. PentesterLab (pentesterlab.com) – Web application security exercises
4. PortSwigger Web Security Academy (portswigger.net/web-security) – Free, world-class web hacking labs
5. VulnHub (vulnhub.com) – Downloadable vulnerable VMs for offline practice
6. DVWA, WebGoat, Metasploitable – Local lab environments for classroom use

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC321BCAI: Cloud Computing and DevOps		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

Course Objectives:

1. understand the implementation of virtualization in cloud computing
2. To learn the application and security on cloud
3. To study risk management in cloud computing
4. To comprehend the modern cloud environment and emerging technologies

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Comprehend basic concepts of cloud computing environment
- CO2: Analyze Virtualization for cloud and install Virtualization software
- CO3: Configure, Test and Deploy applications on Cloud
- CO4: Understand and Apply security in cloud applications
- CO5: Analyze emerging technologies in modern cloud computing

Course Contents

Unit I - Introduction to Cloud Computing (07 Hours)

Cloud Fundamentals: Definition, Importance of cloud computing, Advantages and Disadvantages of Cloud Computing, Characteristics, Categories of Clouds: Private clouds, public clouds Cloud Ser- vice Models: SaaS, PaaS, IaaS, Cloud Architecture, Cloud Storage: Distributed Data Storage, Data management, Cloud Deployment Models

Case Study: Cloud Computing Model of Amazon

Unit II Virtualization in Cloud Computing (09 Hours)

Virtualization: What's virtualization, Benefits of Virtualization, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para virtualization, and Device virtualiza- tion, Virtual Clustering, Virtualization Architecture, Containerization and orchestration, Understand- ing importance of Hypervisors, Virtualization Applications, Issues with Virtualization, Virtualization and Cloud Computing: Virtualizations in Cloud, Virtual Infrastructure, CPU Virtualization, Network and Storage Virtualization

Case Study of VMware: Full virtualization, Xen: Para Virtualization, Microsoft HyperV

Unit III Cloud Platforms and Applications - (09 Hours)

Industrial Cloud Platforms: Amazon Web Services (AWS)- AWS infrastructure, Components, Ama- zon Simple DB, Elastic Cloud Computing (EC2), Amazon Storage System, Amazon Database Services. Microsoft Azure: Azure core concepts, SQL Azure, and Application Services for managed runtimes. Open Source Platforms: Overview of OpenStack, Cloud Stack, and Eucalyptus for private cloud deployment. Cloud Applications: Data-Intensive & Emerging Applications: Smart Cities & IoT: Integrating sensor data from traffic, waste management, and power grids into a centralized cloud dashboard. AI/ML in the Cloud: Case study on Google Photos (image recognition) or Alexa (Natural Language Processing) using cloud-based TPU/GPU instances. Healthcare & Biology: Gene sequencing, protein folding and ECG analysis in the cloud. Geoscience: Satellite image processing and seismic data analysis using cloud clusters.

Case Study: The Google Case Study Data Processing: The evolution from MapReduce to Dremel and Big Query. Storage Innovation: Understanding the Google File System (GFS) and Bigtable as the backbone of global search.

Unit IV - Security in Cloud Computing - (09 Hours)

Risks in Cloud Computing: Risk Management, Enterprise-Wide Risk Management, Types of Risks in Cloud Computing. Data Security in Cloud: Security Issues, Challenges, advantages, Disadvantages, Cloud Digital persona and Data security, Content Level Security. Cloud Security Services: Confidential- ity, Integrity and Availability, Security Authorization Challenges in the Cloud, Secure Cloud Software Requirements, Secure Cloud Software Testing

Case Study: Cloud Security Tool: Acunetix

Unit V - Modern Cloud Environment & Emerging Technologies (09 Hours)

Future Trends in cloud Computing, Mobile Cloud, Comet Cloud, Multimedia Cloud: IPTV, Energy Aware Cloud Computing, Distributed Cloud Computing Vs. Edge Computing, Containers, Dockers, Kubernets, Pod Management Green Cloud & Sustainability: Sustainable Cloud Architecture, Energy-efficient data centre design and carbon footprint tracking.

Case Studies on DevOps: DocuSign, Forter, Gengo

Learning Resource:

Text Book:

1. A. Srinivasan, J. Suresh, “Cloud Computing: A Practical Approach for Learning and Implementa- tion”, Pearson, ISBN: 978-81-317-7651-3
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill Education, ISBN-13:978-1-25-902995-0

MOOC / NPTEL/YouTube Links:-

1. James Bond ,“The Enterprise Cloud”, O’Reilly Media, Inc. ISBN: 9781491907627

8. Dr. Kris Jamsa, “Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more”, Wiley Publica- tions, ISBN: 978-0-470-97389-9
9. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, 2010, The McGraw-Hill.
10. Gautam Shrof, “ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications”, Cam- bridge University Press, ISBN: 9780511778476
11. Tim Mather, Subra K, Shahid L.,”Cloud Security and Privacy”, Oreilly, ISBN-13 978-81-8404-815
12. Ronald L. Krutz, Russell Dean Vines, “Cloud Security: A Comprehensive Guide to Secure Cloud Computing”, Wiley- India,2010
13. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Editors: Wile, 2011

E-Books:-

1. http://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/CLOUD%20COMPUTING%20Princip
2. https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem_2/DECAP470_CLOUD
3. https://studytm.wordpress.com/wp-content/uploads/2014/03/hand-book-of-cloud-computing.pdf
4. https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf
5. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pdf

Mooc Course:-

1. https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. http://www.ndl.gov.in/he_document/nptel/nptel/N_C_S_A_E_C_C_A_D_S_N_I_T_C_C_536752663
3. https://onlinecourses.nptel.ac.in/noc26_cs29/preview?
4. https://onlinecourses.nptel.ac.in/noc21_cs15/preview?

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC321CCAI: Soft Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

Prerequisite Courses, if any :

1. Mathematics for Computing / Engineering Mathematics.
2. Basics of Artificial Intelligence.
3. Programming Fundamentals

Course Objectives: The course aims to:

1. Introduce the fundamental concepts and techniques of soft computing.
2. Explain the principles of genetic algorithms and optimization techniques.
3. Provide knowledge of fuzzy logic, neural networks, and evolutionary computing methods.
4. Develop the ability to apply soft computing methods to real-world engineering and decision-making problems.
5. Familiarize students with hybrid soft computing techniques for solving complex problems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain the basic concepts and importance of soft computing techniques.
- CO2: Apply fuzzy logic concepts for reasoning and decision-making problems.
- CO3: Demonstrate the working principles of artificial neural networks.
- CO4: Analyze and implement genetic algorithms and evolutionary optimization methods.
- CO5: Develop solutions for real-world problems using hybrid soft computing approaches.

Course Contents

Unit I – Introduction to Soft Computing and Fuzzy Logic (9 hours)

Introduction to Soft Computing, Need for Soft Computing, Characteristics of Soft Computing, Hard Computing vs Soft Computing, Components of Soft Computing, Applications of Soft Computing in engineering and real-world problems, Introduction to Fuzzy Logic, Crisp sets and Fuzzy sets, Properties of fuzzy sets, Membership functions, Linguistic variables, Fuzzy logic operations.

Case Study: Application of Fuzzy Logic in Temperature Control System (e.g., air conditioner or washing machine control).

Unit II – Fuzzy Logic Systems and Controller Design (09 hours)

Fuzzy Logic System Architecture, Fuzzy rule-based systems, Fuzzification and Defuzzification methods, Fuzzy inference systems (Mamdani and Sugeno models), Design of fuzzy controller, Applications of fuzzy logic in decision making and control systems, Advantages and limitations of fuzzy logic systems.
Case Study: Fuzzy Traffic Signal Control System for managing congestion in urban areas.

Unit III – Fundamentals of Artificial Neural Networks (09 hours)

Introduction to Artificial Neural Networks (ANN), Biological neuron vs artificial neuron model, Characteristics of neural networks, Types of neural networks, Single-layer and multilayer neural networks, Activation functions, Perceptron model, Learning rules, Supervised and unsupervised learning, Backpropagation algorithm, Applications of neural networks in pattern recognition and prediction.
Case Study: Using Neural Networks for Handwritten Digit Recognition.

Unit IV – Fundamentals of Evolutionary Computing and Generic Algorithms (09 hours)

Introduction to Evolutionary Computing, Genetic Algorithms (GA) concepts, Basic terminology in genetic algorithms, Genetic operators: Selection, Crossover, Mutation, Fitness function, Genetic algorithm flow, Applications of genetic algorithms in optimization problems, Introduction to swarm intelligence concepts (overview), Advantages and limitations of evolutionary algorithms.
Case Study: Application of Genetic Algorithms for Optimal Resource Allocation or Scheduling Problems.

Unit V – Hybrid Soft Computing Systems and Applications (09 hours)

Content: Hybrid Soft Computing Systems, Neuro-Fuzzy systems, Genetic-fuzzy systems, Adaptive neuro-fuzzy inference system (ANFIS) overview, Optimization using hybrid techniques, Applications of soft computing in engineering, data mining, robotics, and intelligent systems, Recent trends in soft computing such as deep learning integration and intelligent decision systems.

Case Study: Application of Hybrid Soft Computing Techniques in Medical Diagnosis Systems.

Learning Resources:

Text Books:

1. S. Rajasekaran, G. A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", 2nd Edition, PHI Learning, 2017.
2. N. P. Padhy, S. P. Simon, "Soft Computing with MATLAB Programming", Oxford University Press, 2015.
3. Himanshu Singh, Bhanu Prasad, "Soft Computing: Concepts and Applications", PHI Learning, 2014.

Reference Books:

1. Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2020.

MOOC / NPTEL/YouTube Links:

4. NPTEL – Introduction to Soft Computing https://nptel.ac.in/courses/106105173; Covers fuzzy logic, neural networks, genetic algorithms, and soft computing applications.
5. NPTEL – Fuzzy Logic and Neural Networks https://nptel.ac.in/courses/127105006; Explains fuzzy sets, neural network models, clustering, and neuro-fuzzy systems.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC321DCAI: Gaming and Animation		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Mathematics, Programming

Course Objectives: The course aims to:

1. Provide foundational knowledge of computer graphics concepts relevant to animation and gaming applications.
2. Introduce fundamental techniques of 2D and 3D animation used in digital media.
3. Familiarize students with the stages and workflow of the animation production pipeline.
4. Develop an understanding of game design principles, mechanics, and development tools.
5. Encourage creative thinking and logical problem-solving in designing basic animated and gaming applications.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain and analyze computer graphics concepts used in animation and gaming
- CO2: Apply animation techniques for 2D/3D content creation.
- CO3: Analyze and describe animation production workflows.
- CO4: Design structured game mechanics and gameplay systems.
- CO5: Develop basic game prototypes integrating animation and gaming concepts.

Course Contents

Unit I - Fundamentals of Computer Graphics (09 Hours)

Introduction to Computer Graphics, Transformations: 2D and 3D transformations (Translation, Rotation, Scaling), Projection: Orthographic vs. Perspective projections, Cohen-Sutherland Line Clip-ping, Lighting Basics: Ambient, Diffuse, and Specular reflection models; Shading Algorithms: Phong Shading and Garraud shading. Color Models (RGB, CMY, HSV)

Case Study: Graphics Transformations in Angry Birds -Study the use of 2D transformations such as translation, rotation, and scaling in character movement and object interaction. Discuss basic pro-jection and collision representation used in simple 2D game environments.

Unit II - Principles and Techniques of Animation (09 Hours)

Definition and History of Animation, 12 Principles of Animation, Types of Animation: Traditional, Stop Motion, Computer Animation, 2D Animation Techniques: Keyframes and

Tweening (Motion paths, Interpolation Techniques, Tim-ing charts), Layer based Animation,3D Animation Techniques: Modeling, Rigging, Skinning, Motion Capture and Simulation,

Case Study: 12 Principles of Animation in Frozen - Analyze the application of squash and stretch, anticipation, follow-through, and timing in character animation. Identify how keyframes and rigging techniques enhance realism in animated sequences.

Unit III - Animation Production Pipeline (09 Hours)

Pre-Production: Storyboarding, Scriptwriting, Concept Art, Production: Modeling, Texturing, Lighting, Rendering, Post-Production: Compositing, Editing, Sound Integration, Visual Effects (VFX) and Motion Graphics, Rendering Engines and File Formats, Role of Physics and AI in Modern Anima-tion,

Case Study: Case Studies: Short Animation Film Workflow, Workflow of Big Buck Bunny -Examine the stage of animation production including storyboarding, modeling, lighting, rendering, and post-production. Understand how structured workflow ensures quality output.

Unit IV - Fundamentals of Game Design (09 Hours)

Introduction to Gaming: History and Genres, Game Design Elements: Story, Character, Level, and Interface Design, Gameplay Mechanics and Game Balancing, Game Flow, Rules, and Objectives, Game Art and Audio Design, Introduction to Game Engines: Unity, Unreal Engine, Godot, Prototyping and Testing of Games

Case Study: Game Mechanics in Tetris -Study core gameplay mechanics, scoring system, difficulty progression, and player engagement model. Analyze how simple design principles create addictive gameplay.

Unit V - Game Development and Integration with Animation (09 Hours)

2D and 3D Game Development Process, Physics: Motion, Gravity, Collision Detection, and User Interactions, Integration of Animated Assets in Games, Game AI Basics (Pathfinding, Decision Making), Optimization and Performance in Games, Introduction to XR (AR/VR/MR) in Gaming

Case Study: NPC Behavior in Pac-Man -Analyze ghost movement patterns as an example of basic rule-based AI behavior. Discuss state-based logic and simple decision-making in early game develop-ment.

Learning Resources:

Text Books:

1. D. Hearn, M. Baker, “Computer Graphics with OpenGL”, 4th Edition, Pearson Education, 2014, ISBN 978-93-325-1871-1
2. Andrew Hogue, Rick Parent, "Computer Animation Algorithms and Techniques", 4th Edition, Morgan Kaufmann Publishers, 2022
3. Jason Gregory, “Game Engine Architecture”, 4th Edition, A K Peters/CRC Press, 2023.
4. K. L. Murdock, Autodesk Maya 2024 Basics Guide. Boston: SDC Publications, 2023.

Reference Books:

1. C. Briggs, An Essential Introduction to Maya Character Rigging, 2nd ed. Abingdon: Routledge, 2021.
2. “Understanding 3D Animation Using Maya” by John Edgar Park
3. “Animated Storytelling: Simple Steps For Creating Animation and Motion Graphics” by Liz Blazer
4. “Gpu Gems 2: Programming Techniques for High – Performance Graphics and General – Purpose Computation” by Matt Pharr and Randima Fernando (Series Editor)
5. “Creating 3-D Animation: The Aardman Book of Filmmaking” by Peter Lord and Brian Sibley
6. “3D Animation Essentials (Essentials (John Wiley))” by Andy Beane
7. “Exploring 3D Animation with Maya 6 (Design Exploration)” by Peter Young and Patricia Beck-mann
8. “Blender 3D by Example” by Romain Caudron and Pierre-Armand NicqImations By Dr. Abhishek Kumar & Dr. Achintya Singhal.

MOOC / NPTEL/YouTube Links:

1. https://onlinecourses.swayam2.ac.in/ learning/preview/cec26_ge03
2. https://youtu.be/4ZSWwHk4AOQ?si=N_6CFC5RmOh0T3DO
3. https://www.youtube.com/watch?v=6XBq8_iNlxY

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC322ACAI: Cyber security Laboratory-I		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks

Companion Course if any: Software Engineering and Project Management

Course Objectives: The course aims to:

1. To provide hands-on experience in applying software engineering principles, tools, and methodologies throughout the software development life cycle.
2. To develop skills in requirements analysis, system design, coding, testing, and documentation using industry-standard practices.
3. To strengthen the ability to plan, manage, and execute software projects using project management techniques, tools, and teamwork strategies.
4. To expose students to version control, software modeling, and quality assurance practices in a collaborative development environment.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Apply software engineering methodologies to analyze requirements and design software solutions.
- CO2: Develop, test, and document software applications using appropriate programming tools and development environments.
- CO3: Use project management techniques such as project planning, scheduling, risk management, and effort estimation for software projects.
- CO4: Work effectively in teams using version control systems and collaborative development practices.
- CO5: Evaluate software quality attributes such as reliability, maintainability, and usability, and apply testing strategies to improve software performance.

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should in- clude prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of

Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Comple- tion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if appli- cable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribu- tion towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. Dur- ing practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, trans- parent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction - Cyber Security and Data Privacy Lab

List of Assignment - Group A

1. A data science team is designing an AI based student analytics portal. Demonstrate the application of cyber security and data privacy principles by explaining the CIA
2. A legacy banking system still relies on DES for certain secure transactions. Implement a program to perform encryption and decryption using DES to simulate secure data exchange.
3. TriadA secure messaging app requires text encryption using classical ciphers. Develop a program to encrypt and decrypt user messages using the Playfair Cipher to demonstrate secure communication basics.
4. A system administrator needs to restrict access to a server. Configure basic firewall rules using iptables to allow HTTP traffic while denying SSH connections.
5. A software vendor wants to ensure authenticity of updates. Implement digital signature generation and

verification using RSA to validate sender identity and message integrity.

6. An online examination system must ensure that uploaded answer sheets are not tampered with; generate MD5 and SHA 1 hash values to verify data integrity.
7. A social media platform detects multiple fake accounts spreading malicious links. Simulate a social engineering attack using fake profiles and demonstrate preventive measures such as user awareness training, link filtering etc
8. A banking application needs to establish a secure communication channel between client and server; simulate Diffie Hellman key exchange to generate a shared secret key over an insecure network
9. A college student receives threatening messages on social media. To seek justice, the student must use official cybercrime reporting platforms such as the National Cyber Crime Reporting Portal. Demonstrate how to access the platform, file a complaint, and track the case status to ensure timely action
10. A college admission portal collects personal details of students such as name, address, and academic records. To comply with the DPDP Act, the portal must ensure that sensitive data is securely stored, transmitted only with consent, and protected from unauthorized access.

Mini project: Implement any one of the following

1. Secure Chat Application: Build a messaging app that integrates Diffie Hellman for key exchange, hashing for integrity, RSA signatures for authentication, and firewall rules for traffic control.
2. Digital Document Vault: Create a secure repository where files are hashed, signed, and verified before storage or retrieval, with firewall protection for access control.
3. IoT Device Security Framework: Design a lightweight security protocol for IoT devices using key exchange, hashing, and signatures, plus firewall rules to isolate devices from external threats.
4. Incident Response Simulator: Develop a simulation tool where students respond to cyber incidents by applying cryptography, hashing, signatures, and firewall defense in real time scenarios.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC322BCAI: Cloud Computing and DevOps Laboratory-I		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks

Guidelines for Student's Laboratory Conduction

1. To study fundamental concepts of cloud, understand components of Cloud Architecture and Deployment models for Cloud.
2. To study the concept of Storage as a Service (SaaS) and implement cloud storage using a cloud platform.
3. Install VirtualBox/VMware Workstation with different flavors of Linux or windows OS on top of windows 7 or 8.
4. Install a C compiler in the virtual machine created using virtual box and execute Simple Pro-grams.
5. Implement procedure to transfer the files from one virtual machine to another virtual machine
6. Google App Engine (GAE) Setup: Install Google App Engine. Create hello world app or any other simple web applications using python/java.
7. Case Study on PaaS (Google App Engine). Use GAE launcher to launch the web applications.

List of Assignment - Group B (Any THREE from 1 to 5, 6th and 7th is mandatory)

1. Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.
2. Creating an application in Salesforce.com using Apex Programming Language
3. To study creating a warehouse application in Salesforce.com
4. To study installation and Configuration of Hadoop.
5. Install Hadoop single node cluster and run simple application like wordcount.
6. To study the Cloud Computing Security Issue
7. Case study on Amazon or Microsoft Azure Cloud

List of Assignment - Group C One Mini-Project

1. Setup your own Cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. Write your own code for Cloud Controller using Open Source Technologies to implement with HDFS.
2. Implement the basic Operations such as divide the file in segments /blocks and download file from cloud in encrypted form. Host a portfolio or documentation site using Amazon S3 (AWS), Azure Blob Storage,

or Google Cloud Storage. Integrate a Content Delivery Network (CDN) like AWS Cloud Front to learn about global content distribution and HTTPS.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC322CCAI: Soft Computing Laboratory-I		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks

Companion Course if any: Soft computing

Course Objectives: The course aims to:

1. To understand the fundamentals of soft computing, its characteristics, components, and differences between hard computing and soft computing through practical experiments.
2. To apply fuzzy logic concepts such as fuzzy sets, membership functions, linguistic variables, and fuzzy operations for solving control and decision-making problems.
3. To design and implement fuzzy inference systems including fuzzification, rule-based systems, and defuzzification techniques for real-world applications.
4. To implement artificial neural network models including perceptron, multilayer networks, activation functions, and back propagation algorithm for pattern recognition and prediction problems.
5. To apply evolutionary computing techniques such as genetic algorithms and explore hybrid soft computing approaches like neuro-fuzzy systems for optimization and intelligent system applications.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain the concepts of soft computing, fuzzy sets, membership functions, linguistic variables, and fuzzy logic operations through practical demonstrations.
- CO2: Design and implement fuzzy logic systems including fuzzification, rule-based inference, and defuzzification methods for solving control and decision-making problems.
- CO3: Apply artificial neural network models such as perceptron and multilayer neural networks using supervised learning and backpropagation algorithms for pattern recognition tasks.
- CO4: Implement evolutionary computing techniques such as genetic algorithms for solving optimization problems and analyze their performance.
- CO5: Explore hybrid soft computing techniques such as neuro-fuzzy systems and analyze their applications in intelligent systems and real-world problem solving.

Course Contents

Guidelines for Instructor's Manual

The instructor's manual/Lab Manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of course, conduction and Assessment guidelines, topics un- der consideration-concept, objectives, outcomes, set of typical applications/assignments/guidelines, references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal con- sists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy.

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Students programs maintained on cloud or college server by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purpose.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be done based on overall performance and Labora- tory assignments performance of student. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall as- sessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent stu- dents. The instructor may set multiple sets of assignments and distribute them among batches of students.

It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged. In addition to these, instructors may assign one real life ap- plication in the form of a mini-project based

on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to respective branch beyond the scope of the syllabus.

Set of suggested assignment lists is provided in groups- A, B, C, D, and E. Each student must perform at least 9 assignments (All assignments for group A are compulsory, 2 from group B, 2 from group C, 1 from group D and 1 from group E).

• All assignments should be implemented using the Python programming language. Python libraries such as NumPy, scikit-learn, and scikit-fuzzy may be used for implementing soft computing algorithms.

•Operating System Recommended: 64-bit Open Source Linux or its derivatives such as Ubuntu or Fedora.

• Programming Tools Recommended**:** Open-source Python interpreter and package managers such as Anaconda or Manikonda for installing required libraries and managing environments.

• Development environments or text editors: Visual Studio Code, Jupiter Notebook, PyCharm Community Edition, or terminal-based editors such as Vim or Emacs.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Learning Resources

Virtual Laboratory:

https://scikit-fuzzy.readthedocs.io/en/latest/

Suggested List of Laboratory Experiments/Assignments

Sl. No.	Practical List
1	Study and implementation of basic Soft Computing concepts by comparing hard computing and soft computing approaches and demonstrating simple problem solving using approximate reasoning.
2	Implementation of crisp sets and fuzzy sets and verification of basic fuzzy set properties such as union, intersection, and complement using suitable programming tools.
3	Design and visualization of different membership functions (triangular, trapezoidal, Gaussian, and bell-shaped) for representing fuzzy variables in a fuzzy system.
4	Implementation of linguistic variables and fuzzy logic operations to model uncertain data and

	perform fuzzy reasoning for a simple decision-making problem.
5	Design and implementation of a Fuzzy Inference System (FIS) using fuzzification and defuzzification techniques to solve a basic control problem.
6	Development of a fuzzy logic-based temperature control system (e.g., air conditioner or washing machine) using fuzzy rules and membership functions.
7	Implementation of a simple Artificial Neural Network (ANN) to demonstrate the perceptron model and activation functions for classification of input patterns.
8	Implementation of a multilayer neural network using the backpropagation algorithm for solving a pattern recognition problem such as handwritten digit classification.
9	Implementation of a Genetic Algorithm including selection, crossover, mutation, and fitness evaluation to solve an optimization problem such as resource allocation or scheduling.
10	Design and implementation of a hybrid soft computing system (e.g., Neuro-Fuzzy or Genetic-Fuzzy model) for solving a real-world problem such as medical diagnosis or intelligent decision support.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC322DCAI: Gaming and Animation Laboratory-I		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks

Guidelines for Laboratory Conduction - Gaming and Animation

Group A

1. Implement basic 2D geometric transformations (translation, rotation, scaling) using OpenGL.
2. Implement Cohen Sutherland Line clipping using OpenGL.
3. Create a short storyboard representing an animated sequence.
4. Design a 2D animation using keyframes and twining techniques.
5. Model and rig a simple 3D object or character
6. Develop a short animation sequence demonstrating lighting, textures, and rendering.
7. Prepare a Game Design Document (GDD) for a small game concept.
8. Develop a simple 2D interactive game prototype using Unity or Godot.
9. Integrate animated 3D assets within a game scene and control them via scripts.

Group B - Mini Project

1. Design and develop a short interactive game/animated scene combining animation and gameplay logic.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
MDM331-CAI: Internet of Things		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial: 01 Hours/Week **Practical :** 02 Hours/Week	03	**Term Work**: 25 Marks **Oral** : 25 Marks

Prerequisite Courses: Digital Electronics, Programming and Problem Solving, Embedded Systems

Course Objectives: The course aims to:

1. Explore the fundamentals, history, and evolution of Raspberry Pi, Arduino, and other microcontrollers.
2. Interface sensors (IR, temperature, gas) with Arduino/Raspberry Pi and develop real-time embedded applications.
3. Implement machine learning models for prediction, classification, and anomaly detection using IoT sensor data.
4. Design smart IoT-based systems using sensors.
5. Develop secure IoT systems for secured data transmission.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Elaborate the architecture and working principles of Raspberry Pi, Arduino, and microcontrollers.
- **CO2:** Build real-time IoT applications using various sensors.
- **CO3:** Develop predictive models using machine learning techniques.
- **CO4:** Implement classification and anomaly detection models for IoT applications.
- **CO5:** Design secure IoT solutions ensuring authentication, privacy, and secure data communication.

Course Contents for Tutorials:-

Unit I - Introduction to IoT and System Architecture (02 Hours)

Introduction to IoT: IoT Components, IoT Design, Role of cloud in IoT, Applications of IoT, Physical & Logical design of IoT, IoT setup using Arduino Uno or Raspberry Pi.

IoT Architecture: M2M & Simplified IoT Architecture, IoT protocol stack, Introduction to M2M.

Unit II - IoT Communication, Networking and Protocols (03 Hours)

IoT Communication: various IoT Communication Models, Communication Patterns: D2D,D2C,D2G Back-end Data Sharing. IoT Networks: TCP/IP Model ,6LoWPAN, IPv4 vs IPv6, Short-Range vs long

range Communication Protocols: RFID, LoRa & LoRaWAN, Cellular IoT, Wi-Fi (802.11 variants), AI-enabled Networking.

Unit III - IoT Data Acquisition, Cloud and Edge Platforms (03 Hours)

IoT Data Acquisition: Types of sensors and actuators, hardware and protocols -ADC, DAC, GPIO, UART, SPI, I2C, Real-time data streaming and buffering, Challenges in IoT data acquisition. Cloud Platforms for IoT: IoT cloud platforms (AWS IoT Core, Microsoft Azure IoT Hub, Google Cloud), Cloud services for IoT. IoT Data Management and Compute Stack: Cloud IoT;Edge Computing in IoT- Concept of edge computing and fog computing, Edge devices and gateways, optimization for latency, bandwidth optimization, enhanced privacy.

Unit IV - Intelligent IoT Applications (03 Hours)

AI in IoT- Artificial Intelligence and IoT integration (AIoT), Need for intelligent analytics in IoT systems, Various IoT Applications. Challenges: noisy, incomplete and heterogeneous data. Predictive Analytics using machine learning.

Unit V - Security and Privacy in IoT (03 Hours):

Introduction to IoT Security: IoT Threats, Vulnerabilities & Risk Model, IoT Security Mechanisms; IoT Security and Communication Protocols: TLS/SSL, Risk involved ,DTLS ,authentication & Authorization, Encryption Techniques,5G IoT; Protocols: HTTP and HTTPS in IoT, MQTT, Publish/Subscribe model, QoS Levels, CoAP, RESTful communication, AMQP Protocol ,XMPP, DDS (Data Distribution Service) Privacy in IoT Systems: Privacy Principles for IoT, Types of Data Collected by IoT Devices, Privacy Risks and Challenges in IoT, Privacy Preservation Techniques in IoT

Learning Resources

Text Books:

1. Ovidiu Vermesan, Peter Friess, “Internet of Things: Converging Technologies for Smart Environ-ments and Integrated Ecosystems”, River Publishers, ISBN: 978-87-92982-73-5 (Print)
2. Volker Ziemann, “A Hands-On Course in Sensors Using the Arduino and Raspberry Pi”, 2018, 1st Edition, CRC Press, United States. ISBN: 978-1-032-37748-3 (hbk)ISBN: 978-1-032-37619-6 (pbk).
3. Nathan Ida, “Sensors, Actuators and their Interfaces - A Multidisciplinary Introduction”, 2020, 2 nd Edition, IET, United Kingdom. ISBN: 978-1-78561-835-2, e-ISBN: 978-1-78561-836-9
4. Internet of Things: A Hands-on Approach by Arshdeep Bahga and Vijay Madisetti, ISBN 978 81 7371 954 7

5. Olivier Hersent, David Boswarthick, Omar Elloumi, “The Internet of Things – Key Applications and Protocols”, ISBN: 978-1-119-99435-0, January 2012, 376 pages Wiley.

Mooc / NPTEL / You Tube Links:-

1. Arduino Official Documentation (Reference for microcontroller interfacing and code) https://www.ardui
2. All About Circuits - Textbooks (Fundamentals of electronics, circuits, op-amps) https://www.allaboutcirc
3. SparkFun/Adafruit Sensor & Actuator Tutorials (Practical interfacing guides for common components) https://learn.sparkfun.com/tutorials
4. https://www.youtube.com/watch?v=KsL_uwa0ekY
5. https://onlinecourses.nptel.ac.in/noc22_cs53/preview
6. https://onlinecourses.swayam2.ac.in/ntr25_ed21/preview
7. https://pg.its.edu.in/sites/default/files/KCA043%20Internet%20of%20things%20-IoT%20by%20Raj%
8. https://jcer.in/jcer-docs/E-Learning/Digital%20Library%20/E-Books/Internet-of-things-a-hands-on-approach-%20Arshadeep.pdf

Guidelines for Instructor’s Manual:

The instructor‘s manual is to be developed as a reference and hands-on resource

It should include prologue (about University/program/ institute/ department/foreword/ pref-ace), curriculum of the course, conduction and Assessment guidelines, topics under consider-ation, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student’s Lab Journal:

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor’s sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Lab / TW Assessment:

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, punctuality and performance on time submission, oral Q & A

Guidelines for Laboratory Conduction:

List of laboratory assignments is provided below for reference. The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of coding style, proper indentation and comments. Use of open-source software and recent versions is to be encouraged. In addition to this, instructors may assign one real life application in the form of a mini-project. Based on the concepts learned. Instructors may also set one assignment or mini project.

Suggested List of Laboratory Experiments/Assignment – Part A:

1. Develop an application for the connectivity of the Arduino UNO/Raspberry Pi circuit with sensors.
2. Implement an application to detect obstacles and notify users using LEDs to understand the connectivity of Raspberry-Pi / Arduino with IR sensor.
3. Implement a simple IoT application using an Arduino board to control an LED through GPIO pins in order to understand the physical design of IoT devices and device actuation.
4. Develop a program to detect the gas leakage in the surrounding environment
5. Configure & implement an IoT device (Arduino UNO IoT / ESP module / Raspberry Pi) to connect to a Wi-Fi network and transmit sensor data to a remote computer using TCP/IP communication.
6. Implement an edge computing model where sensor data is processed locally on an edge device (Raspberry Pi) before being transmitted to the cloud in order to reduce latency and bandwidth usage.
7. Design an RFID based identification system that reads tag information and displays the ID on a monitoring system to demonstrate short-range communication technologies in IoT.

Suggested List of Laboratory Experiments/Assignments – Part B:

1. Develop an IoT application that uses the communication protocol to transmit sensor data be-tween an IoT device and a server.
2. Perform data cleaning, normalization, and visualization on IoT sensor dataset(any dataset can be considered)
3. Implement an IoT application to collect real-time temperature data using an IoT sensor and apply a machine learning model to predict future temperature values.(Any Machine Learning model)
4. To design and implement a password-protected smart sensor system that allows access to sen-sor data only after

successful authentication, demonstrating basic IoT security concepts. (Any security Algorithm can be used

5. To implement secure data transmission in an IoT system using a sensor and any communication security protocol
6. Develop & Implement a machine learning model to perform classification (e.g., activity recog-nition or device state detection) and anomaly detection (e.g., fault detection in industrial IoT sensors).
7. Case study for Data Minimization Using Smart Threshold Alert by providing Privacy-by-Design

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
ELC- 342- CAI: Technical Seminar		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Oral** : 25 Marks

Course Objectives: The course aims to:

1. To develop research orientation and technical communication skills in emerging Computer Engineering and Artificial Intelligence domains.
2. To enable students to critically review, analyze, and synthesize contemporary research papers, white papers, patents, and technical standards.
3. To promote interdisciplinary thinking aligned with NEP-2020 multidisciplinary philosophy.
4. To inculcate ethical awareness, sustainability perspective, and societal impact analysis of AI systems.
5. To prepare students for industry, higher education, entrepreneurship, and innovation ecosystems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Identify and select emerging and relevant technical topics through literature survey.
- CO2: Analyze and synthesize information from research papers, journals, and credible sources.
- CO3: Demonstrate effective technical communication skills through oral presentation.
- CO4: Prepare a structured technical report following academic writing standards.
- CO5: Use modern tools (presentation software, plagiarism checkers, referencing tools) for semi- nar preparation.

Guidelines to Conduct Technical Seminar

Topic Selection Guidelines

The Technical Seminar shall be research-oriented and domain-specific, focusing strictly on recent development in Computer Engineering.

- Topic must be from emerging Computer Engineering and Artificial Intelligence domains (last 3–5 years).
- Must involve a minimum of 5 recent research papers (IEEE, ACM, Elsevier, Springer etc). They should summarizing paper – Reading abstracts and finding ideas, conclusion, Advantages of Their approach, and the drawbacks of the papers. Generalize results from a research paper to related research problems. Comparing the approach - Identify weaknesses and strengths in recent research articles in the subject. Practical sessions on how to read, analyze and summarize research papers.

- Should not be a basic textbook topic.
- Must include: Problem statement, State-of-the-art analysis, Comparative study, Ethical & societal impact, Interdisciplinary themes aligned with NEP encouraged.
- Topic approval by a faculty panel

Seminar Process

Stage 1: Orientation & Topic Finalization (Week 1–2)

- Conduct an orientation session explaining: Objectives of the technical seminar, Evaluation criteria and expected outcomes
- Each student must submit: Title of the seminar, Problem statement, Relevance to current technology trends, Approval by guide is mandatory before proceeding

- **Literature Survey & Problem Understanding (Week 2–4)**

- Students must: Refer minimum 5–8 recent research papers (IEEE, Springer, Elsevier, ACM etc.)
- Use scholarly databases like IEEE Xplore, Google Scholar, ScienceDirect
- Prepare a literature survey matrix, including:

∗ Author/year
∗ Methodology used
∗ Key findings
∗ Limitations

- Identify: Research gaps and Challenges in existing approaches

- **Synopsis Preparation & Presentation (Week 4–5)**

- Submit a 2–3 page synopsis including: Introduction, Literature insights, Objectives, Pro- posed seminar scope
- Conduct a Synopsis Presentation (5–7 minutes): Evaluate clarity of understanding, Receive feedback for improvement
- Approval required before proceeding to full report

- **In-depth Study & Content Development (Week 5–8)**

- Students should: Deeply analyze concepts, models, architectures, or case studies, Include diagrams, flowcharts, and comparative tables
- Weekly review meetings with guide: Track progress, Ensure conceptual clarity,

- Emphasis on: Critical analysis (not just description), Real-world applications

- **Draft Report Submission & Review (Week 8–10)**

- Submit first draft of the report
- Guide provides feedback on: Technical content quality, Structure and coherence, Referenc- ing and plagiarism,
- Students must revise based on suggestions by the guide

- **Pre-Seminar Presentation (Mock Evaluation) (Week 10–11)**

- Conduct a mock presentation simulating final evaluation
- Focus on: Presentation skills, Time management, Handling questions
- Peer and faculty feedback should be incorporated

- **Final Report Submission (Week 11–12)**

- Submit: Final hard copy (if required), Soft copy (PDF format)
- Ensure: Proper formatting, Plagiarism compliance (<20%), Correct referencing using reference managers like Zotero and Mendeley Desktop

- **Final Seminar Presentation & Viva Voce (Week 12–13)**

- Presentation duration: 10–15 minutes, Followed by Q&A session (5–10 minutes)
- Evaluation based on: Depth of understanding, Analytical ability, Communication skills

Method of Evaluation

- During the seminar session each student is expected to prepare and present a topic on engineer- ing/ technology, for duration of about 12to 15 minutes.
- Each student is expected to present at least twice during the semester and the student is evaluated based on that.
- At the end of the semester, he / she can submit a report on his / her topic of seminar and marks are given based on the report.
- A Faculty guide is to be allotted and he / she will guide and monitor the progress of the student and maintain attendance.

Third Year Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern - Level 5.0

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Oral	Practical	Total	Theory	Tutorial	Practical	Total
PCC351-CAI	Programme Core Course 1	Software Engineering and Project Management	3	-	-	30	70	-	-	-	100	3	0	-	3
PCC352-CAI	Programme Core Course 2	Deep Learning	2	-	-	30	70	-	-	-	100	2	0	-	2
PCC353-CAI	Programme Core Course Lab 1	Deep Learning Laboratory	-		4	-	-	25	-	25	50	0	0	2	2
PCC354-CAI	Programme Core Course Lab 2	Software Engineering and Project Management Laboratory	-	-	2	-	-	25	25	-	50	0	0	1	1
PEC361-CAI	*Programme Elective Course	Programme Elective-II	3	-	-	30	70	-	-	-	100	3	0	-	3
PEC362-CAI	*Programme Elective Course	Programme Elective-III	3	-	-	30	70	-	-	-	100	3	0	-	3
PEC363-CAI	Programme Elective Course Lab 3	Programme Elective-II Laboratory	-	-	2	-	-	25	-	25	50	0	0	1	1
MDM371-CAI	Multidisciplinary Minor	Robotics and Automation	-	1	2	-	-	25	25	-	50	0	1	1	2
VSE372-CAI	Vocational and Skill Enhancement Course	Solar Technology and Development	-	-	2	-	-	50	-	-	50	0	0	1	1
ELC381-CAI	Experiential Learning Course (Internship/OJT)	Internship	-	-	8	-	-	-	50	-	50	0	0	4	4
Total			**11**	**1**	**20**	**120**	**280**	**150**	**100**	**50**	**700**	**11**	**1**	**10**	**22**

Programme Elective Course II	
PEC361ACAI	Blockchain
PEC361BCAI	Big Data engineering
PEC361CCAI	High Performance Computing
PEC361DCAI	AR/VR

Programme Elective Course III	
PEC362ACAI	Ethical Hacking
PEC362BCAI	Quantum AI
PEC362CCAI	Prompt Engineering
PEC362DCAI	UI/UX

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC351-CAI : Software Engineering and Project Management		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE: 30 Marks End-Semester: 70 Marks

,

Prerequisite Courses, if any:

Programming and Problem Solving. Data Structures, Fundamentals of Computer Systems.

Course Objectives: The course aims to:

1. Introduce the fundamental concepts, principles, and practices of Software Engineering and their role in developing high-quality software systems.
2. Provide comprehensive understanding of software development life cycle models, requirements engineering, and software design methodologies.
3. Enable students to apply software project management techniques including estimation, scheduling, risk management, and quality assurance.
4. Develop the ability to analyse, model, design, test, and maintain software systems using standard engineering practices.
5. Foster professional ethics, teamwork, and responsibility required for successful software development projects.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain fundamental software engineering concepts, software process models, and the evolving role of software in modern applications.
- CO2: Elicit, analyse, document, and validate functional and non-functional requirements using standard requirement engineering techniques and SRS documentation.
- CO3: Apply software design principles, architectural styles, UML modelling, and design patterns to develop effective software solutions.
- CO4: Utilize software project management techniques for project planning, effort estimation, scheduling, risk management, and monitoring of software projects.
- CO5: Apply software quality assurance, testing strategies, configuration management, and maintenance practices to ensure reliable and maintainable software systems

.

Course Contents

Unit I - Introduction to Software Engineering (6 Hours)

Introduction to Software Engineering and its importance, Software characteristics and software crisis, Software applications and evolving role of software, Generic software process framework, Software Development Life Cycle (SDLC) overview, Waterfall, Incremental, V-Model and Spiral models – comparison, Agile software Development principles and manifesto, Software engineering ethics and professionalism.

Case Study: Development of an Online Library Management System to compare SDLC models and demonstrate the importance of structured software engineering practices.

Unit II - Software Requirements and Analysis (6 Hours)

Requirements engineering process, Types of requirements: functional and non-functional, Requirement elicitation Techniques, Requirement analysis and validation, Software Requirements Specification (SRS) – structure and standards, Feasibility study – technical, economic, operational, Use case modeling and requirement traceability, Introduction to requirements management tools.

Case Study: Requirement analysis and SRS preparation for a Hospital Management System including functional, non-functional requirements and feasibility study.

Unit III Software Design and Architecture (6 Hours)

Software design concepts and principles, Modularity, abstraction, cohesion and coupling, Architectural design and architectural styles, Component-level and interface design, User interface design principles, Unified Modeling Language (UML) diagrams, Design patterns – overview and significance, Introduction to software prototyping.

Case Study: Case Study: Designing the architecture and UML models for an E-Commerce Website applying design principles, patterns, and user interface concepts.

Unit IV: Project Management, Estimation and Scheduling (6 Hours)

Introduction to software project management, Project planning activities and project life cycle, Software project estimation techniques, Lines of Code (LOC) and Function Point (FP) analysis, COCOMO model – basic concepts, Project scheduling techniques, Gantt chart, PERT and CPM, Risk management – identification, analysis and mitigation.

Case Study: Case Study: Project planning, cost estimation (COCOMO), scheduling (Gantt/PERT), and risk management for developing a Mobile Banking Application.

Unit V Software Quality, Testing and Maintenance (6 Hours)

Software quality concepts and quality assurance, Software quality models (ISO, CMMI overview), Verification and validation, Software testing fundamentals, Levels of testing – unit, integration, system, acceptance, Testing strategies and test case design, Software configuration management, Software maintenance and re-engineering.

Case Study: Implementing testing strategies, quality assurance practices, and configuration management for an online examination System.

Learning Resources

Text Books:

1. Ian Sommerville – Software Engineering, 10th Edition, Pearson, 2021.
2. Roger S. Pressman & Bruce R. Maxim – Software Engineering: A Practitioner's Approach, 8th Edition, McGraw-Hill Education, 2014.

Reference Books:

1. Project Management Institute – A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 7th Edition, Project Management Institute, 2021.Pankaj Jalote – An Integrated Approach to Software Engineering.
2. Rajib Mall – Fundamentals of Software Engineering, 5th Edition, PHI Learning, 2018.

MOOC / NPTEL/YouTube Links:

1. Software Engineering – Prof. Rajib Mall (IIT Kharagpur).
2. Software Project Management – Prof. K. K. Aggarwal / Prof. Rajib Mall.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC352-CAI : Deep Learning		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	02	CCE: 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Linear Algebra, Probability and Statistics, Machine Learning

Course Objectives: The course aims to:

1. Introduce the fundamental concepts of artificial neural networks
2. Familiarize students with the fundamentals of deep learning.
3. Provide knowledge of Convolutional Neural Networks architectures.
4. Acquaint with the need for sequence models in processing sequential data.
5. Enable students to gain knowledge of the fundamentals of deep generative models.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply neural network concepts for solving real-world problems.
- **CO2:** Make use of appropriate activation and loss functions in model design.
- **CO3:** Utilize Convolutional Neural Networks techniques for solving real-world problems.
- **CO4:** Analyze the need for sequence models in handling sequential data problems.
- **CO5:** Distinguish between different deep generative models such as GANs and autoencoders.

Course Content

Unit I - Fundamentals of Artificial Neural Networks (09 Hours)

Introduction to Neural Network, Biological neuron vs artificial neuron. Artificial Neuron Model: Structure, mathematical representation. Introduction to activation functions. Neural Network Architextures: Single-layer feedforward networks, multi-layer feedforward networks, network representation. Learning in Artificial Neural Network: Supervised learning, Unsupervised learning, training vs testing, loss (error) function. Perceptron Model: Single-layer perceptron, perceptron learning rule, limitations.

Multilayer Perceptron: Need for hidden layers. Overview of forward propagation and backpropagation, advantages over perceptron, applications of Artificial Neural Neywork.

Case Study: Handwritten Digit Recognition using Artificial Neural Network

Unit II - Deep Learning and Training Mechanisms (09 Hours)

Introduction, difference between Machine Learning and Deep Learning. Deep Neural Networks (DNN): Architecture and representation. Activation functions: ReLU, Sigmoid, Tanh, SoftMax. Forward propagation. Loss functions: MSE, Cross-Entropy. Backpropagation. Gradient Descent: Batch, Stochastic, Mini-batch. Optimization Algorithms: SGD, AdaGrad, RMSProp, Adam, applications of deep learning. Challenges in training deep networks: Vanishing and exploding gradients, weight initialization techniques, Overfitting and underfitting. Regularization techniques: L1/L2 regularization, Dropout.

Case Study: Predict house prices based on features like area, location, and number of rooms using a Deep Neural Network.

Unit III - Convolutional Neural Networks (09 Hours)

Introduction, convolution operation and feature maps, kernel and convolution, filter size calculations, padding and stride, pooling layers, CNN architecture design. CNN architectures: LeNet, AlexNet, VGG, ResNet. Transfer learning and fine-tuning. Object detection technique in deep learning: YOLO (You Only Look Once). Applications

Case Study: Traffic sign recognition system using CNN-based transfer learning for automated de-tection and classification of road signs.

Unit IV - Recurrent Neural Networks (RNN) (09 Hours)

Need for sequence models, Recurrent Neural Networks (RNN) architecture, Bidirectional RNNs, Deep Recurrent Networks, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), introduc-tion to Transformers. Text embeddings: Word2Vec, GloVe. Applications of sequence models: Natural language processing, speech recognition, time series forecasting, video analysis, and healthcare signal processing.

Case Study: Build a sequence model using GRU/LSTM to predict future stock prices or weather conditions based on historical time-series data.

Unit V - Deep Generative Models (09 Hours)

Introduction, Generative Adversarial Networks. Autoencoders: Traditional autoencoders, varia-tional autoencoders. Model compression and pruning, model deployment using REST APIs, Tensor-Flow Lite, Edge AI, ethical considerations in deep learning, and energy efficiency and sustainability in deep learning models.

Case Study: Develop a Generative Adversarial Network (GAN) based system for classifying images

as real or fake, focusing on detecting synthetic or manipulated images.

Learning Resources:

Text Books:

1. Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron, "Deep Learning," 1st Edition, MIT Press, 2016, ISBN 978-0262035613.
2. Chollet, François, "Deep Learning with Python," 2nd Edition, Manning Publications, 2021, ISBN 978-1617296864.
3. Géron, Aurélien, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," 3rd Edition, O'Reilly Media, 2022, ISBN 978-109812597.

Reference Book:

1. Aggarwal, Charu C., "Neural Networks and Deep Learning: A Textbook," 1st Edition, Springer, 2006, ISBN 978-3319944623.
2. Bishop, Christopher M., "Pattern Recognition and Machine Learning," 1st Edition, Springer, 2006, ISBN 978-0387310732.
3. Zhang, Aston, Lipton, Zachary C., Li, Mu, and Smola, Alexander J., "Dive into Deep Learning," 1st Edition, Cambridge University Press, 2023, ISBN 978-1009387422.

MOOC / NPTEL / You Tube Link:

1. https://onlinecourse
2. NPTEL course: "Deep Learning" by Prof. P. K. Biswas, IIT Kharagpur. https://onlinecourses.nptel.ac.in/n
3. Dive into Deep Learning (Interactive): Link: https://d2l.ai

E-Books:

1. Dive into Deep Learning: https://arxiv.org/pdf/2106.11342 https://d2l.ai/?utm_source=chatgpt.com
   Deep Learning: An Introduction for Applied Mathematicians, Catherine F. Higham, Desmond J. Higham, 2018. https://arxiv.org/pdf/1801.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC353-CAI: Deep Learning Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	**Term Work :** 25 Marks **Practical :** 25 Marks

Prerequisite Courses: Artificial intelligence, Natural Language Processing, Machine Learning
Companion Course: Deep Learning

Course Objectives:

1. To develop the ability to set up a Python-based deep learning environment and work with a single-layer perceptron.
2. To enable students to implement an MLP using backpropagation and understand activation and loss functions.
3. To provide knowledge of optimization algorithms and regularization techniques for improving neural network performance.
4. To familiarize students with CNN design and transfer learning using pretrained models.
5. To familiarize students with object detection and generative models using YOLO, autoencoders, and GANs.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Demonstrate tensor operations, matrix manipulations, and the functioning of a single-layer perceptron for binary classification on linearly separable data.
- **CO2:** Apply an MLP on a standard dataset to observe the effect of activation and loss functions on model performance.
- **CO3:** Make use of gradient descent variants and regularization techniques to improve model convergence and performance.
- **CO4:** Analyze CNN architectures and transfer learning approaches by comparing pretrained models with custom CNNs for image data.
- **CO5:** Examine object detection and generative models by assessing detection performance and output quality.

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Comple-tion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and

assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and

program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on

Guidelines for Practical Examination

parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Problem statements must be decided jointly by the internal examiner and external examiner. Dur-ing practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, trans-parent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

A list of laboratory assignments is provided below for reference. The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment-framing policy should address the average student and include an element to attract and promote intelligent students. The instructor may set multiple assignment sets and distribute them across student batches. It is appreciated if the assignments are based on real-world problems/applications. Encourage students to use coding style, proper indentation and comments appropriately.

Use of open-source software and recent versions is to be encouraged. Python is preferably used.

In addition, instructors may assign a real-world application as a mini-project. based on the concepts learned.

Suggested List of Laboratory Experiments/Assignments:

1. Environment Setup and Tensor Operations: Demonstrate the setup of a Python-based deep learning environment using TensorFlow or PyTorch, and perform basic tensor operations, matrix manipulations,

and data visualization

2. Implementation of Perceptron: Demonstrate a single-layer perceptron for binary classification on linearly separable data and measure its performance.
3. Multilayer Perceptron with Backpropagation: Build a Multi-Layer Perceptron (MLP) using back-propagation for classification on a standard dataset (e.g., Iris/MNIST).
4. Activation Functions and Loss Analysis: Identify the impact of different activation functions (ReLU, Sigmoid, Tanh) and loss functions on model convergence and accuracy.
5. Optimization Algorithms Comparison: Develop a neural network using different Gradient De-scent variants (SGD, Mini-batch, Adam, RMSProp) and observe convergence behavior.
6. Regularization Techniques: Apply Dropout, Batch Normalization, and Early Stopping to control overfitting and measure model performance.
7. Convolutional Neural Network (CNN) Design and Implementation: Design a Convolutional Neu-ral Network for image data (e.g., MNIST/CIFAR-10) and visualize feature maps.
8. Transfer Learning using Pretrained CNN: Examine a pretrained model (e.g., ResNet/VGG) for image classification and compare its performance with a custom CNN.
9. Object Detection using YOLO: Analyze object detection using a pretrained YOLO model on im-ages/videos and assess detection performance.
10. Auto encoder / GAN Implementation: Design an autoencoder or GAN for image reconstruction or generation and assess output quality.

Mini Project: Students may select any one topic for the mini-project from the following list (not limited to these).

- Disaster Detection using Satellite Images
- Sentiment Analysis for Social Awareness
- Crop Disease Detection System
- Air Quality Prediction using Deep Learning

Learning Resources:

Virtual Laboratory (links):

1. https://vlab.spit.ac.in/ai/
2. https://scte-iitkgp.vlabs.ac.in/

MOOC / Web Links:

1. NPTEL course: "Deep Learning" by Prof. Mitesh M. Khapra, IIT Madras / IIT Ropar. https://onlinecourse

2. NPTEL course: "Deep Learning" by Prof. P. K. Biswas, IIT Kharagpur. https://onlinecourses.nptel.ac.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PCC354-CAI: Software Engineering and Project Management Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Term Work** : 25 Marks **Oral :** 25 Marks

Companion Course if any: Software Engineering and Project Management

Course Objectives: The course aims to:

1. To provide hands-on experience in applying software engineering principles, tools, and methodologies throughout the software development life cycle.
2. To develop skills in requirements analysis, system design, coding, testing, and documentation using industry-standard practices.
3. To strengthen the ability to plan, manage, and execute software projects using project management techniques, tools, and teamwork strategies.
4. To expose students to version control, software modeling, and quality assurance practices in a collaborative development environment.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply software engineering methodologies to analyze requirements and design software solutions.
- **CO2:** Develop, test, and document software applications using appropriate programming tools and development environments.
- **CO3:** Use project management techniques such as project planning, scheduling, risk management, and effort estimation for software projects.
- **CO4:** Work effectively in teams using version control systems and collaborative development practices.
- **CO5:** Evaluate software quality attributes such as reliability, maintainability, and usability, and apply testing strategies to improve software performance.

Course Contents

Guidelines for Instructor's Manual

The instructor's manual or laboratory manual shall be developed as a comprehensive hands-on resource and reference material. It should include a prologue describing the university, institute, department, and program, along with a foreword and preface. The manual must clearly define the course curriculum, laboratory conduction procedures, assessment guidelines, and detailed topics covered in the laboratory. Each topic should outline the

underlying concept, objectives, expected outcomes, typical applications, sample assignments, implementation guidelines, and relevant references related to cloud computing technologies and recent trends.

Guidelines for Student's Laboratory Journal

Students are required to submit laboratory assignments in the form of a journal. The journal should consist of a prologue, certificate, table of contents, and handwritten write-ups for each assignment. Each assignment write-up must include the title, objectives, problem statement, outcomes, software and hardware requirements, date of completion, assessment grade or marks with assessor's signature, a brief theory concept, algorithm, flowchart, test cases, test datasets if applicable, mathematical model if applicable, and a conclusion or analysis. Program codes along with sample outputs of all performed assignments must be submitted in softcopy format.

As a conscious effort toward Green IT and environmental sustainability, students are encouraged to avoid attaching printed papers as part of the journal or program listings. Maintaining programs on cloud platforms or college servers under the supervision of the laboratory in-charge is highly encouraged. For accreditation purposes, one or two reference journals with printed program listings may be maintained in the laboratory.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work shall be carried out based on the overall performance of the student and the quality of laboratory assignments. Each assignment shall be evaluated using appropriate weightage parameters such as timely completion, innovation, efficiency of implementation, correctness, punctuality, and neatness. These parameters should be consistently applied for both individual assignment assessment and overall laboratory performance.

Guidelines for Laboratory Conduction

The instructor is expected to design laboratory assignments by considering prerequisites, technological relevance, utility, and recent advancements in cloud computing. Assignments should be inclusive, addressing average learners while incorporating elements that challenge and motivate advanced students. The instructor may create multiple sets of assignments and distribute them among different student batches. Preference should be given to assignments based on real-world cloud applications. Students should be encouraged to follow coding best practices, including proper naming conventions, indentation, documentation, and use of open-source tools. In addition, instructors may assign a mini-project or real-life application based on cloud computing concepts learned during the course, including interdisciplinary or branch-specific applications beyond the prescribed syllabus.

A structured set of suggested laboratory assignments shall be provided in **Groups A, B, C, D, and E**. Each student must complete a **minimum of nine assignments**, including **all assignments from Group A**, **two assignments from Group B**, **two assignments from Group C**, **one assignment from Group D**, and **one assignment from Group E**.

The laboratory assignments focus on the **practical application of software engineering and project management concepts**, including **requirements analysis, software design, coding standards, software testing, version control, project planning, scheduling, risk management, and documentation**. Assignments may involve **case studies, software development tasks, project planning exercises, and team-based implementation activities**.

The **recommended operating system** for this laboratory is **64-bit Microsoft Windows**, which provides a widely used, user-friendly, and industry-relevant development environment. Students are encouraged to use **Java** as the primary programming language due to its platform independence, strong object-oriented support, and extensive use in software engineering education and industry applications.

Programming and project management tools may include **Java Development Kit (JDK)**, **version control systems such as Git**, **software modeling and documentation tools**, and **Integrated Development Environments (IDEs)** such as **Eclipse** or **IntelliJ IDEA Community Edition**. These tools enable students to **design, develop, test, manage, and document software projects effectively** while supporting collaborative and team-based development practices.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Learning Resources

- **Oracle Java Documentation** – Java language and API reference.
- **Eclipse Documentation** – IDE usage, debugging, and project management.

Suggested List of Laboratory Experiments/Assignments

Sr. No.	Group A - (All)
1	Write menu based program- Requirements Engineering and SRS Preparation Select a real-world software system (Library Management, Hospital System, Student Portal, etc.) and: Identify stakeholders. Elicit functional and non-functional requirements. Prepare a Software Requirements Specification (SRS) as per IEEE standard.

2	Software Design Using UML Based on the prepared SRS: Draw **Use Case Diagram.** Draw **Class Diagram.** Draw **Sequence Diagram.** Use any UML tool and explain design decisions.
3	Implement a small module of the selected system in **Java** following coding standards. Use **Git** to: Create a repository. Commit changes. Demonstrate version history.
Group B - Assignments (Any TWO)	
4	Software Testing and Test Case Design For the implemented module: Design **test cases** (black-box and white-box). Perform unit testing. Record test results and defects.
OR	
	Prepare a **test plan and test case document** for a given software application and perform manual or automated testing.
5	Software Testing and Test Case Design For the implemented module: Design **test cases** (black-box and white-box). Perform unit testing. Record test results and defects.
OR	
	Prepare a **test plan and test case document** for a given software application and perform manual or automated testing.
Group C- Assignments (Any TWO)	
6	Given an existing codebase:

	Identify code smells. Refactor the code. Compare readability and performance before and after refactoring.
	OR
	Perform a **software maintenance activity** by implementing enhancements or correcting defects in an existing system and documenting the changes.
7	Project Planning and Scheduling For the selected project: Create a **Work Breakdown Structure (WBS).** Prepare **Gantt Chart** and **PERT Chart.** Identify milestones and deliverables.
	OR
	Prepare a **detailed project plan** for a given software project, including schedule, milestones, and resource allocation.
	Group D- Assignments (Any ONE)
8	Identify: Technical, schedule, and resource risks. Risk mitigation strategies. Resource allocation plan (people, tools, time).
	OR
	Perform a **risk analysis case study** on a failed or delayed software project and propose mitigation strategies.
	Group E - Assignments (Any ONE)
9	Simulate an **Agile/Scrum process** by: Creating a product backlog. Planning a sprint. Conducting sprint review and retrospective.
	OR
	Compare **Agile and Waterfall models** for a given project scenario and justify the most suitable development approach.
	Group F – Assignments
10	Design and implement a mini project to simulate **task scheduling for a software project**. Use appropriate data structures (arrays, linked lists, or priority queues) to manage tasks.

	Include task attributes: Task ID, assignee, estimated time, priority, and status. Implement scheduling algorithms (e.g., priority-based, FCFS, or deadline-based). Generate reports: pending tasks, completed tasks, and resource utilization.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC361A-CAI: Blockchain		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester**: 70 Marks

Prerequisite Courses: Data Structures, Network and Security, Machine Learning

Course Objectives: The course aims to:

1. Identify the fundamentals of blockchain technology and its applications in AI and data science
2. Examine cryptocurrency, smart contracts, and decentralized data systems
3. Analyze consensus algorithms and their role in distributed ledger systems
4. Construct smart contracts using Ethereum and Solidity for data-driven applications
5. Evaluate blockchain solutions for real-world AI and data science challenges

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Describe the fundamentals of blockchain technology
- CO2: Demonstrate the use of cryptocurrency wallets to perform blockchain transactions.
- CO3: Analyze appropriate consensus mechanisms for distributed systems
- CO4: Construct smart contracts using Ethereum and Solidity for data-driven applications
- CO5: Develop blockchain-based solutions for applications in AI and data science

Course Content

Unit I - Introduction to Blockchain and Cryptographic Foundations (09 Hours)

Cryptography: Symmetric Key Cryptography and Asymmetric Key Cryptography, Elliptic Curve Cryptography (ECC), Cryptographic Hash Functions: SHA256, Digital Signature Algorithm (DSA), Merkle Trees **Blockchain Basics:** History, Limitation of Centralized System, Decentralized Systems,

Layers in Blockchain **Types of Blockchain:** Public, Private and Consortium **Case Study:** Hash-based integrity verification

Unit II - Cryptocurrency and Bitcoin (09 Hours)

Introduction: Bitcoin and the Cryptocurrency, Cryptocurrency, Bitcoin Transactions and Scripts, Wallets and Keys, Mining Process, Blockchain Security Mechanisms, Cryptocurrency Economics **Types**

of Cryptocurrencies: Cryptocurrency Usage, Crypto wallets: MetaMask, Coinbase, Binance

Types of Blockchain Platforms: Bitcoin, Ethereum, Hyperledger, IoTA, Corda, and R3.

Case Study: Create your own wallet for cryptocurrency using any of the blockchain platforms.

Unit III Consensus Mechanism (09 Hours)

Blockchain Architecture: Layers Of Blockchain: Application Layer, Execution Layer, Semantic Layer, Propagation Layer, Consensus Layer **Consensus Algorithms:** Proof of Work, Byzantine General Problem, Proof of Stake, Proof of Elapsed Time, Proof of Activity, Proof of Burn, Proof of Authority, Scalability and Performance

Case Study: Select an appropriate consensus mechanism (such as PBFT or PoA) for a blockchain-based decentralized energy trading system

Unit IV - Ethereum and Smart Contracts (09 Hours)

Ethereum Platform Architecture: Need of Ethereum, Type of Ethereum Platforms, Ethereum Vir-tual Machine (EVM), Gas and Transaction Fees, Smart Contract Fundamentals, Solidity Programming Language, Deploying and Interacting with Smart Contracts using Remix, Decentralized Applications (Dapps), Decentralized Storage (IPFS) Web3.Js Basics

Case Study: Deploy a smart contract for the supply chain

Unit V - Blockchain Applications (09 Hours)

Blockchain For Data Provenance and Integrity, Decentralized Data Marketplaces, Blockchain In Healthcare (Medical Records, Drug Traceability), Supply Chain Transparency And Tracking, Financial Services and Defi, Identity Management Systems, Voting Systems, NFTs And Digital Assets, Integration of Blockchain with AI/ML Models

Case Study: Secure model sharing using blockchain

Learning Resources:

Text Books:

1. Imran Bashir, "Mastering Blockchain," 2nd Edition, Packt Publishing, 2018
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, "Bit-coin and Cryptocurrency Technologies", Princeton University Press, ISBN:9780691171692 1st Edition, 2016

Reference Books:

1. Andreas M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, 2nd Edition, O'Reilly Media, Sebastopol, California, USA, 2017

2. Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, Sebastopol, California, USA, 2015.

3. Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress, New York, USA, 2017.

4. William Mougayar, The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, Wiley, Hoboken, New Jersey, USA, 2016.

5. Arshdeep Bahga and Vijay K. Madisetti, Blockchain Applications: A Hands-On Approach, VPT Publications, Blacksburg, Virginia, USA, 2017.

Mooc / NPTEL / You Tube Link:

1. NPTEL Course on "Introduction to Blockchain Technology & Applications" Link: https://nptel.ac.in/cours

2. NPTEL Course on "Blockchain and its Applications "Link: https://nptel.ac.in/courses/106/105/1061051

3. Coursera: "Blockchain Basics" by University at Buffalo

4. edX: "Blockchain for Business" by The Linux Foundation Quantum Computing by Prof .Debabrata Goswami, IIT Kanpur Link: https://nptel.ac.in/courses/104104082

E-Books:

1. https://www.asau.ru/files/pdf/1789486.pdf
2. https://users.cs.fiu.edu/~prabakar/cen5079/Common/textbooks/Mastering_Blockchain_2nd _Edition.pdf
3. https://www.lopp.net/pdf/princeton_bitcoin_book.pdf
4. https://www.blockchainexpert.uk/book/blockchain-book.pdf

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC361B-CAI: Big Data engineering		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Basic understanding of Data Structures, Database Management Systems (DBMS), Operating Systems, and programming knowledge in Java or Python is essential.

Course Objectives: The course aims to:

1. Interpret key Big Data principles and the challenges encountered during implementation.
2. Describe the architecture of Hadoop, including HDFS and the MapReduce programming model.
3. Utilize Big Data processing tools and analytics frameworks.
4. Implement Big Data techniques for real-world applications such as web and social network ana-lytics.
5. Assess ethical issues and promote responsible use of Big Data.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Analyze the key characteristics of Big Data along with its architecture and distributed computing concepts.
- **CO2:** Operate within the Hadoop ecosystem, including HDFS and MapReduce.
- **CO3:** Apply Hadoop tools and process frameworks for data analytics tasks.
- **CO4:** Design and develop Big Data analytics solutions for practical applications.
- **CO5:**Develop secure solutions that uphold data privacy and meet compliance requirements

Course Contents:

Unit I - Introduction to Big Data & Its Ecosystem (09 Hours)

Introduction to Big Data, Evolution of Big Data, Characteristics of Big Data (Volume, Velocity, Variety, Veracity, Value), Types of Data – Structured Data, Semi-Structured Data, Unstructured Data, Limitations of Traditional Data Processing Systems, Distributed Computing Concepts, Parallel Processing, Big Data Architecture Layers, Overview of Hadoop Ecosystem, Applications of Big Data.

Case Study: A college generates large amounts of data such as attendance, exam results, LMS logs,

and feedback forms. The traditional database system becomes slow during result processing. How can Big Data concepts help in handling Volume, Velocity, and Variety of campus data

Unit II Hadoop and HDFS - (09 Hours)

Hadoop Architecture, Core Components of Hadoop, Hadoop Distributed File System (HDFS), NameNode, DataNode, Secondary NameNode, Block Storage Mechanism, Data Replication Strategy, Fault Tolerance in Hadoop, MapReduce Programming Model, Map Phase, Reduce Phase, Combiner Function, YARN Architecture, Resource Management in Hadoop.

Case Study: Design a solution using the Hadoop Distributed File System (HDFS) to resolve storage capacity, scalability constraints, and fault tolerance issues for an e-commerce company storing millions of daily transaction records on a single server.

Unit III -Apache Spark and Data Processing (09 Hours)

Basics of Apache Spark, Spark Architecture – Driver, Executor, Cluster Manager, Resilient Distributed Datasets (RDD), RDD Operations – Transformations and Actions, DataFrames, Spark SQL, Spark Streaming Basics, Comparison between Hadoop MapReduce and Spark, In-Memory Processing Concept.

Case Study: Enable a social media platform to efficiently manage unstructured data in various formats (user posts, images, comments, likes) by leveraging NoSQL databases like MongoDB for flexible schema design and high scalability.

Unit IV - Big Data Tools and Applications (09 Hours)

Data Ingestion Techniques, Apache Flume, Apache Sqoop, Batch Processing, Real-Time Processing, Data Pipeline Architecture, Cloud-Based Big Data Platforms, Big Data Security Basics, Data Governance Concepts, Industry Applications – Healthcare, Banking, E-commerce, Smart Cities.

Case Study: Leverage Big Data analytics techniques to enable a bank to detect fraudulent transactions in real time, analyze customer spending behavior, and support predictive decision-making for fraud prevention.

Unit V - Big Data Security & Governance (09 Hours)

Security: Security challenges in Big Data, authentication and authorization, data encryption (at rest and in transit), access control mechanisms (RBAC and ABAC), audit and compliance monitoring, risk management and disaster recovery

Governance: Data privacy and compliance, data governance and lifecycle, data anonymization and masking, ethical considerations

Case Study: An organization handling sensitive data requires Big Data security and governance mechanisms like encryption, RBAC/ABAC access control, and data masking to ensure privacy, compliance, and secure data management.

Learning Resources:

Text Books:

1. Diego Rodrigues, “Fundamentals of Big Data: With Hadoop and Spark”, Self/Academic Press, 2024 Edition, ISBN: 978-XXXXXXX (latest 2024 integrated Hadoop–Spark framework text covering modern big data systems)
2. Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee, “Learning Spark: Lightning-Fast Big Data Analysis”, O’Reilly Media, 2nd Edition, 2020, ISBN: 9781492050049.
3. Ambrish Kumar Sharma, J Jegan Amarnath, Dr. G. Vadivel, and S. Suganya, “A Brief Guide to Big Data & Hadoop”, AG Publishing House, 2022, ISBN: 978-93-90593.
4. Mayank Bhushan, “Big Data Analytics: Introduction to Hadoop, Spark and Machine Learning”, BPB Publications, 2024 Edition, ISBN: 978-93-5551

Mooc / NPTEL / You Tube Link:

1. https://www.coursera.org/specializations/packt-big-data-foundations-with-hadoop-and-spark?- Includes Hadoop, Spark, Scala, and Kafka concepts with hands-on labs.
2. https://www.educative.io/courses/introduction-to-big-data-and-hadoop?- beginner-friendly Big Data fundamentals and Hadoop basics.
3. YouTube/Video Links: 1. https://www.youtube.com/watch?v=YHiS441Bk1E
4. https://www.overdrive.com/media/3001853/big-data-analytics?utm_source=chatgpt.com

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC361C-CAI: High Performance Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester**: 70 Marks

Course Objectives: The course aims to:

1. To understand the fundamentals of High Performance Computing.
2. To learn parallel computing concepts and performance evaluation techniques.
3. To apply basic parallel programming models.
4. To understand GPU and accelerator-based computing.
5. To explore cloud-based HPC for AI applications.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1 :** Explain fundamental HPC concepts and performance metrics.
- **CO2 :** Apply parallel programming using shared and distributed memory models.
- **CO3 :** Describe GPU and accelerator-based computing for AI workloads.
- **CO4 :** Analyze HPC solutions for AI and data-intensive applications.
- **CO5 :** Evaluate cloud-based HPC systems for scalable computing.

Course Contents:

Unit I - Fundamentals of Modern HPC (09 Hours)

Introduction to HPC: Evolution from sequential to parallel computing, need for HPC in modern workloads, limitations of sequential systems. Parallel vs Distributed Computing. Shared vs Distributed Memory. Basics of Parallel Computing: Concurrency vs Parallelism, Data Parallelism, Task Parallelism, Flynn's Classification (SISD, SIMD, MIMD). Performance Metrics: Execution time, Speedup, Efficiency, Throughput, Scalability (Strong and Weak Scaling). Performance Laws: Amdahl's Law and Gustafson's Law. Applications of HPC in AI, ML, Big Data and Scientific Computing.

Case Study: Speedup Analysis of Deep Learning Model Training using Multi-core Processors

Unit II - Parallel Programming Models (09 Hours)

Introduction to Parallel Programming and "Thinking Parallel". Shared Memory Programming using OpenMP (parallel for, reduction, tasks – basic concepts). Distributed Memory Programming using MPI (send, receive, broadcast – basic communication). Comparison of OpenMP and MPI. Introduction to

Hybrid Programming. Synchronization concepts: barriers and basic race conditions.

Case Study: Parallel Matrix Multiplication using OpenMP vs MPI.

Unit III - Accelerators and GPU Computing (09 Hours)

Need for hardware accelerators in modern HPC and limitations of CPU-centric systems for AI workloads. Overview of accelerators such as GPUs, FPGAs and TPUs. Fundamentals of GPU architecture: streaming multiprocessors, thread hierarchy (threads, blocks, grids), SIMT execution model, and GPU memory hierarchy. Basic GPU execution workflow. Introduction to CUDA programming model: kernel launch, host-device interaction, and basic memory management concepts. Role of GPUs in AI and deep learning: matrix operations, training and inference acceleration. Overview of optimized libraries such as cuBLAS and cuDNN. Basic performance considerations: memory bandwidth, latency, and idea of multi-GPU systems with CPU–GPU coordination.

Case Study: Performance measurement, bottleneck identification, optimization (kernel fusion, Tensor Cores, batch size tuning), Comparison with CPU-only baseline & with FPGA-based inference (optional)

Unit IV - HPC for AI and Data-Intensive Applications (09 Hours)

Introduction to HPC for AI: Why AI needs HPC, computational challenges in deep learning, Key aspect of HPC in AI, Architecture, Overview of AI workloads: training vs. Inference, Data Parallelism vs Model Parallelism, Examples of AI frameworks and their HPC integration Distributed Deep Learn-ing, Application of HPC for AI. Introduction to Data-Intensive Applications: Big Data Architectures, Distributed File Systems, Data Locality, MapReduce Paradigm, Spark-based Distributed Processing. Performance challenges in large-scale AI systems: communication overhead and scalability issues.

Case Study: Distributed Training of a Deep Learning Model using Data Parallelism.

Unit V - Cloud and Modern HPC Systems (09 Hours)

Introduction to Cloud-based HPC and virtual clusters. On-premises HPC vs Cloud HPC.

Infrastructure as a Service (IaaS) for HPC workloads. Containers (Docker – basic concept) for portable HPC applications. Introduction to Kubernetes for workload orchestration. Elastic scaling and auto-scaling in cloud environments. Cost-aware computing and use of spot instances. Overview of Energy efficiency and green computing in HPC. Future trends: Exascale computing, AI-driven HPC optimization.

Case Study: Deployment of AI Model Training on Cloud Cluster – Cost and Scalability Analysis.

Learning Resources:

Text Book:

1. Grama, Ananth, Gupta, Anshul, Karypis, George, and Kumar, Vipin, Introduction to Parallel Computing, Pearson Education, 2nd Edition, 2003, ISBN: 978-0201648652.

2. Quinn, Michael J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004, ISBN: 978-0072822564.

3. Hwang, Kai, Dongarra, Jack, and Fox, Geoffrey, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 1st Edition, 2012, ISBN: 978-0123858801.

Reference Books:

1. Kirk, David B., and Hwu, Wenmei W., Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 3rd Edition, 2016, ISBN: 978-0128119860.

2. Rauber, Thomas and Rünger, Gudula, Parallel Programming for Multicore and Cluster Systems, Springer, 2nd Edition, 2013, ISBN: 978-3642378010.

MOOC / NPTEL / You Tube Link:

1. High Performance Computing by Prof. S. Gopalakrishnan, IIT Madras https://nptel.ac.in/courses/10610

2. Parallel Computer Architecture and Programming – IIT Kanpur (NPTEL) https://nptel.ac.in

3. Introduction to High Performance Computing https://www.youtube.com/watch?v=KsL_uwa0ekY

E-Books:

1. https://dl.acm.org/doi/pdf/10.5555/3455710 NVIDIA CUDA Programming Guide (Official Documentation)

2. https://docs.nvidia.com/cuda/ OpenMP API Specification (Official Documentation)

3. https://www.openmp.org/specifications/ MPI Standard Documentation

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC361DCAI: AR/VR (Augmented Reality & Virtual Reality)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE**: 30 Marks **End-Semester**: 70 Marks

Prerequisite Courses: Linear Algebra, Probability and Statistics, Machine Learning

Course Objectives: The course aims to:

1. To understand the fundamentals and evolution of Augmented Reality (AR) and Virtual Reality (VR) technologies.
2. To learn basic 3D graphics and scene representation concepts required for immersive environments.
3. To study interaction techniques and user interface design principles used in immersive systems.
4. To understand augmented reality tracking, spatial mapping and mobile AR frameworks.
5. To analyze applications, challenges and ethical considerations of AR/VR systems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain the concepts, components and applications of AR/VR systems. (L2 – Understand)
- CO2: Apply basic 3D scene representation and transformation concepts in immersive environments. (L3 – Apply)
- CO3: Describe interaction techniques and user interface design principles in virtual environments. (L2/L3 – Understand/Apply)
- CO4: Explain tracking, registration and spatial mapping techniques used in augmented reality systems. (L2 – Understand)
- CO5: Analyze real-world applications, performance issues and ethical considerations in AR/VR systems.

Course Contents

Unit I - Introduction to AR and VR (09 Hours)

Introduction and evolution of immersive technologies, definitions of AR, VR, MR and XR, components of AR/VR systems, hardware devices such as head-mounted displays (HMDs), controllers and sensors, human perception, immersion and presence, field of view and frame rate considerations, motion sickness issues in VR, applications of AR/VR in gaming, healthcare, education and industry, challenges and future trends.

Case Study: Virtual Reality in Medical Training, Study of VR-based surgical simulation systems

used in medical education to train surgeons in a safe and controlled virtual environment.

Unit II - Representing the Virtual World (8 Hours)

Basics of 3D graphics for immersive environments, representation of virtual environments, 3D geo- metric objects and scene structures, coordinate systems (world, local, camera), geometric transformations (translation, rotation, scaling), lighting and materials in virtual scenes, texture mapping concepts, spatial audio and 3D sound, haptic feedback concepts, avatars and virtual objects.

Case Study: Design of Virtual Environments in Game Engines, Study of how game engines such as Unity create immersive 3D environments using models, lighting, textures and spatial audio.

Unit III Interaction and Navigation in Virtual Environments (8 Hours)

Interaction techniques in virtual environments, navigation and locomotion methods, gesture-based interaction, controller-based interaction, natural user interfaces, VR user interface design principles, immersive UX design, cybersickness reduction techniques, performance and latency considerations

Case Study: Interaction Design in VR Gaming, Analysis of interaction mechanisms in the VR rhythm game Beat Saber focusing on gesture-based interaction a nd user engagement.

Unit IV Augmented Reality Systems and Tracking (8 Hours)

Principles of augmented reality, marker-based tracking techniques, marker-less tracking approaches, concept of simultaneous localization and mapping (SLAM), overview of sensor fusion, spatial mapping and plane detection, AR content registration and anchoring, mobile AR frameworks such as ARCore and ARKit.

Case Study: Augmented Reality in Retail Visualization, Study of AR applications used by furniture retailers that allow customers to visualize products in their homes using mobile devices and marker- less AR.

Unit V Advanced Immersive Systems and Applications (8 Hours)

Mixed reality systems, WebXR and browser-based immersive applications, collaborative virtual environments, industrial and healthcare AR/VR applications, ethical, privacy and security issues in immersive systems, future trends in spatial computing.

Case Study: Virtual Reality for Industrial Training, Analysis of VR-based safety and equipment training used in manufacturing industries to simulate real-world operations.

Learning Resources:

Text Books:

1. Understanding Virtual Reality: Interface, Application and Design — William R. Sherman and Alan B. Craig, Morgan Kaufmann.
2. Augmented Reality: Principles and Practice — Dieter Schmalstieg and Tobias Hollerer, Addison- Wesley.

Reference Books:

1. Learning Virtual Reality — Tony Parisi, O'Reilly Media.
2. Augmented Reality for Developers — Jonathan Linowes, Packt Publishing.
3. Virtual Reality Technology — Grigore Burdea and Philippe Coiffet, Wiley.
4. 3D User Interfaces: Theory and Practice — Doug Bowman, Addison-Wesley.

Mooc / NPTEL / You Tube Links:

1. Foundation for Virtual and Augmented Reality Systems, IIT Guwahati : https://onlinecourses.nptel.ac.in learning/preview/noc26_cs03
2. Foundations of Virtual Reality, IIT Madras, Prof. M. Manivannan : https://onlinecourses.nptel.ac.in/e- learning/preview/noc25_cs87
3. Mobile Virtual Reality and Artificial Intelligence : https://onlinecourses.nptel.ac.in/e-learning/preview/

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC362ACAI : Ethical Hacking		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE**: 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Computer Networks, Operating Systems, Cyber Security Fundamentals

Course Objectives: The course aims to:

1. Equip students with core knowledge of ethical hacking, its application scope, relevant legal frameworks, and the evolving cyber threat landscape
2. Develop understanding of reconnaissance techniques and attack surface analysis using OSINT and exposure assessment methods.
3. Build knowledge of common vulnerabilities, exploitation concepts, and attack techniques used in real-world scenarios.
4. Analyze identity-based attacks, social engineering methods, and post-compromise activities in modern systems.
5. Explore security challenges in modern environments including web, API, cloud, and AI-driven systems along with ethical responsibilities

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply knowledge of ethical hacking frameworks, lifecycle stages, and professional guidelines to assess modern threat landscapes.
- **CO2 :** Utilize reconnaissance methodologies and tools to identify exposed assets and map attack surfaces effectively
- **CO3:** Identify common vulnerabilities and demonstrate basic exploitation concepts in controlled environments.
- **CO4:** Analyze identity-based attacks, credential misuse, and post-compromise activities to understand attacker behavior.
- **CO5:** Evaluate security risks in web, API, cloud, and AI systems and understand ethical responsibilities in cybersecurity practices.

Course Contents

Unit I - Ethical Hacking Approach & Modern Threat Landscape (09 Hours)

Ethical Hacking – definition, purpose, and scope; Difference between penetration testing, vulnerability assessment, and red teaming; Ethical hacking lifecycle – reconnaissance, scanning, exploitation, post-compromise, reporting; Rules of Engagement – authorization, scope, ethical boundaries; At-tacker mindset – targeting weakest link, identity-focused attacks, attack chaining; Overview of MITRE ATT&CK – understanding attacker tactics.

Modern Threat Landscape: Modern attack trends – identity-based attacks, cloud misconfiguration, API abuse, supply chain attacks, Phishing evolution – AiTM phishing, session/token theft, Zero Trust Security Model and identity-centric security principles, AI-augmented attacks – LLM prompt in-jection, adversarial ML, deepfake-enabled social engineering.

Case Study: Modern identity-based breach involving credential misuse, AiTM phishing, and unauthorized cloud access (e.g., analysis of a real-world Microsoft 365 compromise scenario).

Unit II Reconnaissance & Vulnerability Discovery (09 Hours)

Reconnaissance – definition, importance, passive vs active; OSINT – public sources; Digital foot-print and attack surface; DNS and domain intelligence; Service exposure; Vulnerability scanning con-cept; WHOIS, ASN lookups, BGP peering analysis, Google Dorking and advanced search operators, GitHub/GitLab OSINT – exposed secrets, API keys, credentials in public repos

Scanning & Enumeration: Network scanning – Nmap techniques: SYN, UDP, version detection, OS fingerprinting ,Service enumeration :banner grabbing, version identification, Vulnerability scanning concept – Nessus, OpenVAS, Cloud attack surface :exposed S3 buckets, Azure blobs, GCP storage misconfigurations ,Internet-wide exposure awareness : Shodan queries, Censys facets, exposed IoT devices ,Attack surface management (ASM) platforms .

Case Study: Data exposure due to misconfigured cloud storage: full reconnaissance chain from OSINT to bucket discovery, impact analysis, and responsible disclosure process.

Unit III Vulnerabilities & Exploitation Concepts (09 Hours)

Vulnerabilities : types; Difference between vulnerability and exploit; CVE, CVSS scoring, NVD, Exploit-DB, VulnHub, Metasploit Framework : architecture, modules Web vulnerabilities – OWASP Top 10 (2021 & 2025 updates) – overview of each category ,broken access control, authentication weaknesses; Common attack concepts – SQL Injection, XSS; Password attacks; Credential attacks; Exploitation concept.Network & System Attacks: Password attacks – brute force, dictionary, credential stuffing, rainbow tables

Network sniffing: ARP poisoning, passive capture (Wireshark), Wireless attacks: WPA2 cracking, Evil Twin, Buffer overflow concept – stack vs. heap

Emerging: IoT & OT Attack Surface: IoT attack vectors, OT/ICS/SCADA security fundamentals and attack scenarios, Embedded device exploitation awareness (UART, JTAG)

Case Study: Web application compromise due to chained vulnerabilities – SQL injection to authentication bypass, privilege escalation, and data exfiltration in a simulated e-commerce portal.

Unit IV : Identity Attacks & Post-Compromise Overview (09 Hours)

Identity as Attack Surface: Identity-centric security, Active Directory attacks,Azure AD / Entra ID attack paths Credential-based attacks, MFA bypass techniques.

Social Engineering: Social engineering, Phishing campaigns, AiTM (Adversary-in-the-Middle) phishing Business Email Compromise (BEC) , AI-enhanced social engineering – voice cloning, deepfake video calls, Physical security – badge cloning, social engineering for physical access.

Post-Compromise Activities: Post-exploitation objectives, Session/token abuse Data exfiltration methods and C2 (Command & Control) frameworks overview, Covering tracks.

Detection Awareness: SIEM alerts and EDR detection for attacker techniques, Attacker OPSEC – avoiding detection in red team engagements, Threat hunting basics from attacker perspective.

Case Study: Identity-based attack chain: spear phishing, AiTM credential capture, MFA bypass lateral movement via Pass-the-Hash, data exfiltration from SharePoint. Analysis includes detection opportunities at each stage.

Unit V - Modern Attack Domains: Web, API, Cloud & AI Threats (09 Hours)

Web Penetration Testing: Web pretesting methodology , Advanced web attacks – HTTP request smuggling, web cache poisoning, clickjacking ,Tools: Burp Suite Pro (intruder, repeater, active scan), OWASP ZAP,**API Security Testing**: REST, GraphQL, gRPC API attack techniques , API fuzzing, rate limit bypass, parameter pollution , JWT attacks ,Tools: Postman, Insomnia **Cloud Penetration Testing**: Cloud pentesting scope – AWS, Azure, GCP specific attack paths,, Cloud metadata exploitation (SSRF to IMDS), Container security – Docker escape, Kubernetes RBAC misconfigurations Tools: Pacu (AWS), ScoutSuite, Prowler, CloudSploit AI/LLM Security (Emerging Area): AI attack surface – LLM prompt injection, jailbreaking, adversarial inputs Poisoning attacks on ML training data, Securing AI/ML pipelines – model hardening, input validation, output filtering. **Ethical & Legal Framework**: IT Act 2000 & Amendments – Sections 43, 66, 66C, 66F relevant to hacking

Case Study: Cloud/API breach simulation: SSRF to cloud metadata service, IAM token extraction privilege escalation S3 data exfiltration. Full penetration test report writing exercise.

Learning Resources:

Text Books:

1. Patrick Engebretson – The Basics of Hacking and Penetration Testing, 3rd Ed., Syngress/Elsevier, (ISBN number 978-0443438868). March 2026
2. Rafay Baloch – Ethical Hacking and Penetration Testing Guide, Auerbach/CRC Press, 2014 (ISBN: 978-1482208917).
3. Matt Walker – CEH Certified Ethical Hacker All-in-One Exam Guide, 5th Ed., McGraw-Hill, 2022 (ISBN: 978-1264274567).

4. Georgia Weidman – Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press, 2014 (ISBN: 978-1593275648

Reference Books:

1. Jon Erickson – Hacking: The Art of Exploitation, 2nd Ed., No Starch Press, 2008 (ISBN: 978-1593271442).
2. 2Peter Kim – The Hacker Playbook 3: Practical Guide to Penetration Testing, Secure Planet, 2018 (ISBN: 978-1980901754). [Red teaming, advanced TTPs]
3. Christopher Hadnagy – Social Engineering: The Science of Human Hacking, 2nd Ed., Wiley, 2018 (ISBN: 978-1119433385).
4. Chris Anley et al. – The Shellcoder's Handbook: Discovering and Exploiting Security Holes, 2nd Ed., Wiley, 2007 (ISBN: 978-0470080238).
5. Michael Sikorski & Andrew Honig – Practical Malware Analysis, No Starch Press, 2012 (ISBN: 978-1593272906).
6. Daniel Miessler – The Real Internet of Things, Leanpub

NPTEL / Web Link:

1. Ethical Hacking by Prof. Indranil Sen Gupta, IIT Kharagpur (NPTEL)
2. Cyber Security and Privacy by Prof. Saji K Mathew, IIT Madras (NPTEL)
3. Introduction to Cyber Security – Dr. Jeetendra Pande, Uttarakhand Open University
4. Practical Ethical Hacking – TCM Security (Heath Adams) – Udemy/TCM Academy
5. Web Application Hacking & Bug Bounty – TryHackMe and HackTheBox Academy
6. CEH v12 Preparation – EC-Council Official (paid) / free preview modules

You Tube Links:

1. The Cyber Mentor (TCM Security) – Full Ethical Hacking Course
2. NetworkChuck – Ethical Hacking Playlist (Beginner-friendly)
3. IppSec – HackTheBox Walkthroughs (Advanced Pentesting)
4. John Hammond – CTF & Malware Analysis

5. David Bombal – Network Hacking & Ethical Hacking
6. LiveOverflow – Binary Exploitation & Web Hacking
7. STOK – Bug Bounty Hunting & Web App Security

E-Books:

1. Alana Maurushat – Ethical Hacking, University of Ottawa Press (Open Access, free PDF)
2. PCI Security Standards Council – Penetration Testing Guidance (Free PDF from pcisecuritystandards.org)
3. OWASP Testing Guide v4.2 (Free PDF from owasp.org) – comprehensive web pentesting methodology
4. NIST SP 800-115 – Technical Guide to Information Security Testing (free from nist.gov)
5. Zaid Sabih – Learn Ethical Hacking from Scratch (Packt eBook / Udemy course)
6. CREST – Penetration Testing & Security Assessment Guides (free PDFs from crest-approved.org)
7. HackTricks Book (free, gitbook.io/hacktricks) – continuously updated red team techniques

Web Links:

1. TryHackMe (tryhackme.com) – Guided learning paths for ethical hacking
2. HackTheBox Academy (academy.hackthebox.com) – Professional penetration testing labs
3. PentesterLab (pentesterlab.com) – Web application security exercises
4. PortSwigger Web Security Academy (portswigger.net/web-security) – Free, world-class web hacking labs
5. VulnHub (vulnhub.com) – Downloadable vulnerable VMs for offline practice
6. DVWA, WebGoat, Metasploitable – Local lab environments for classroom use

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC362BCAI : Quantum AI		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Engineering Mathematics, Engineering Physics

Course Objectives: The course aims to:

1. Introduce the fundamental concepts and motivation behind quantum computing.
2. Develop the ability to use vector spaces, Hilbert spaces, operators, and tensor products in quan-tum mechanics.
3. Enable students to evaluate the advantages and limitations of quantum algorithms in solving computational problems.
4. Provide insight into quantum supervised and unsupervised learning techniques.
5. Familiarize with Quantum AI approaches for decision-making and problem-solving in diverse domains.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply principles of superposition, entanglement, and quantum interference to represent and manipulate qubits.
- **CO2:** Utilize the postulates of quantum mechanics to describe quantum measurements, expec-tation values, and time evolution of quantum systems.
- **CO3:** Make use of fundamental quantum algorithms to address computational problems effi-ciently.
- **CO4:** Analyze quantum learning techniques and quantum circuits for solving complex problems.
- **CO5:** Examine Quantum AI techniques for decision-making and problem-solving in diverse domains.

Course Contents

Unit I - Fundamentals of Quantum Computing (09 Hours)

Overview of Quantum Computing: Motivation and evolution of quantum computing. Limitations of Classical Computing: Computational complexity, parallelism limits, and the need for quantum advantage. Quantum Mechanics for Computing: Superposition, entanglement, and quantum interference. Qubits vs Classical Bits: Representation of qubits and Bloch sphere visualization. Quantum Gates: Pauli gates (X, Y, Z), Hadamard gate, Controlled-NOT gate (CNOT). Quantum Circuits and Measurement: Quantum circuit model and measurement in quantum systems.

Case Study: Classical Bit vs Quantum Bit: Superposition Experiment

Unit II Mathematical Foundations of Quantum Mechanics (09 Hours)

Linear Algebra Framework for Quantum Mechanics: Vector spaces, complex Hilbert spaces, inner products, orthogonality, linear operators, eigenvalues and eigenvectors, Hermitian and unitary opera-tors, tensor products. Postulates of Quantum Mechanics: Quantum states and state space, observables as operators, measurement postulate and Born rule, expectation values, time evolution of quantum

states, quantum states in Hilbert space, qubits and Bloch sphere representation, quantum measure-ment theory, density operator formalism, composite quantum systems. Fundamentals of Quantum Principle: No-cloning theorem and implications for quantum information and quantum AI.

Case Study: Detection of Eavesdropping in a Quantum AI Communication System using the No-Cloning Principle.

Unit III - Quantum Algorithms (09 Hours)

Overview of quantum algorithms and quantum speedup, Phil's algorithm, David Deutsch's algo-rithm, and David Deutsch–Jozsa algorithm, Daniel Simon's algorithm, Lov Grover's Search algorithm, Quantum Fourier Transform (QFT), Peter Shor's algorithm for integer factorization, introduction to quantum walk and search application.

Case Study: Classical vs Quantum Search: Understanding Grover's Algorithm.

Unit IV : Quantum Machine Learning (09 Hours)

Introduction to quantum-classical hybrid learning, data encoding and quantum feature maps, vi-brational quantum circuits (VQC) and parametrized quantum circuits, quantum optimization and the parameter shift rule. Quantum Supervised Learning: Quantum support vector machine (QSVM) and quantum neural networks. Quantum Unsupervised Learning: Quantum K-Means and Quantum Princi-pal Component Analysis (QPCA), quantum generative models, and the barren plateau problem.

Case Study: Conceptual Study of Hybrid Quantum-Classical Machine Learning Models

Unit V : Applications of Quantum Artificial Intelligence (09 Hours)

Quantum AI for Optimization Problems: Quantum-enhanced combinatorial optimization, logistics and supply chain optimization, resource allocation and scheduling, smart grid, and traffic optimization. Quantum AI in Finance and Business Analytics: Portfolio optimization, risk analysis and modeling, fraud detection, and financial forecasting. Quantum AI in Healthcare and Life Sciences: Drug d iscovery optimization, molecular modeling, genomic data analysis. Quantum AI in Cybersecurity: Quantum-enhanced anomaly detection, intrusion detection systems, AI-assisted post-quantum cryptanalysis. Quantum AI in Intelligent Systems: Quantum machine learning applications, intelligent decision systems, and advanced AI architectures.

Case Study: Quantum AI-Based Smart Traffic Signal Optimization System

Learning Resources:

Text Books:

1. ***Wichert, Andreas, "Quantum Artificial Intelligence with Qiskit," 1st Edition, Chapman & Hall / CRC Press, 2025, ISBN 978-1032448978***
2. Bernhardt, Chris, "Quantum Computing for Everyone," 1st Edition, The MIT Press, 2019, ISBN 978-0262039253.
3. Kaye, P., Laflamme, R., and Mosca, M., "An Introduction to Quantum Computing," 1st Edition, OUP Oxford, 2006, ISBN 978-0191524615.
4. Schuld, Maria, Petrucciani, Francesco, "Supervised Learning with Quantum Computers", 1st Edition, Springer International Publishing, 2018, ISBN 978-3319964249.

Reference Books:

1. Nielsen, M. A., Chuang, I. L., "Quantum Computation and Quantum Information", 10th Edition, Cambridge University Press, 2010, ISBN 978-1107002173.
2. Wittek, Peter, "Quantum Machine Learning: What Quantum Computing Means to Data Mining", 1st Edition, Elsevier Science, 2014, ISBN 978-0128010990.
3. Yanofsky, Noson S., Mannucci, Mirco A., "Quantum Computing for Computer Scientists", 1st Edition, Cambridge University Press, ISBN 978-0-521-87996-5.

Mooc / NPTEL / You Tube Links:

1. NPTEL course: "Quantum Computing" by Prof. Debabrata Goswami, IIT Kanpur https://www.youtube.c Gm0yRYwTj7Fs6jyzYa83HErSrpXgPQ
2. NPTEL course: "Introduction to Quantum Computing: Quantum Algorithms and Qiskit" by Prof. Prabha Mandayam, Prof. Anupama Ray, Prof. Sheshashayee Raghunathan, IIT Madras & IBM Research.
3. https://onlinecourses.nptel.ac.in/noc26_cs89/preview
4. Quantum Computation Fundamentals: https://www.youtube.com/watch?v=KsL_uwa0ekY
5. Quantum machine learning: https://qiskit.org/learn/course/machine-learning-course/

1. Center for Excellence in Quantum Technology: https://research.ibm.com/blog/next-wave-quantum-centric-supercomputing

E- Books:

1. Mermin, N. D., "Quantum Computer Science: An Introduction," Cambridge University Press, 2007, ISBN 978-0521876572

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC362C-CAI: Prompt Engineering		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE :** 30 Marks **End-Semester**: 70 Marks

Prerequisite Courses, if any:

1. Artificial Intelligence

Course Objectives: The course aims to:

1. To introduce the fundamentals of Generative AI and the role of prompt engineering in interacting with AI systems.
2. To understand the principles of designing clear, structured, and effective prompts.
3. To explore various prompting techniques for improving AI-generated responses.
4. To apply prompt engineering methods in programming, technical documentation, and data analysis tasks.
5. To analyze ethical issues, limitations, and responsible use of AI-generated content.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain the concepts of prompt engineering and its role in modern AI systems.
- CO2: Design effective prompts to obtain accurate and relevant outputs from AI tools.
- CO3: Apply advanced prompting techniques to solve engineering and programming problems.
- CO4: Evaluate the quality and reliability of AI-generated responses.
- CO5: Demonstrate responsible and ethical use of Generative AI tools in academic and professional applications.

Course Contents

Unit I -: Introduction to Generative AI and Prompt Engineering (09 Hours)

Content: Introduction to Generative AI and Prompt Engineering, overview of Artificial Intelligence, Machine Learning and Deep Learning, concept of large language models, role of prompts in AI interaction, components of a prompt (instruction, context, input data, output format), introduction to AI tools such as ChatGPT and Google Gemini, applications of prompt engineering in education, research and software development.

Case Study: A university wants to develop an AI assistant for students to explain difficult engineering concepts. The assistant currently gives generic or overly technical answers that students cannot understand. Task for Students: Design prompts that help the AI provide simple explanations with examples suitable for undergraduate students.

Unit II - Fundamentals of Prompt Design (09 Hours)

Content: Fundamentals of prompt design, structure of effective prompts, prompt clarity and specificity, context based prompting, instruction based prompting, prompt constraints and output formatting, types of prompting techniques such as zero shot prompting, few shot prompting and role based prompting, prompt iteration and refinement methods, designing prompts for explanation, summarization and classification tasks.

Case Study: A startup wants to use AI to summarize research papers for engineers. When using a generic prompt, AI outputs either incomplete or overly long summaries. Task for Students: Design structured prompts ncluding context, instructions, and output format to get concise, informative summaries for different engineering domains.

Unit III - Advanced prompting techniques (09 Hours)

Content: Advanced prompting techniques, reasoning based prompts, step by step reasoning methods such as Chain-of-Thought Prompting, concept of multi-step prompts, prompt templates and reusable prompt structures, prompt optimization and debugging, controlling tone, style and format of AI responses, prompt engineering for complex problem solving and analytical tasks.

Case Study: An engineering company needs AI to analyze sensor data and provide step-by-step explanations of anomalies. The AI fails when asked in simple instructions. Task for Students: Apply Chain-of-Thought or Tree-of-Thought prompting to guide AI in reasoning and producing stepwise analytical solutions.

Unit IV- Prompt Engineering for Engineering Applications (09 Hours)

Content: Applications of prompt engineering in engineering domains, prompt based code generation and debugging, use of prompts for software development tasks, technical documentation generation using AI tools, prompts for data analysis and report generation, prompts for research assistance and literature review, automation of engineering workflows using generative AI systems.

Case Study: A software development team wants to generate reusable code snippets and documentation using AI. When prompts are vague, generated code is incomplete or inconsistent.Task for Students: Design prompts for automated code generation, commenting, and documentation, ensuring correctness and usability.

Unit V - Ethics, Limitations and Future of Prompt Engineering (09 Hours)

Content: Ethical and responsible use of generative AI, bias and hallucination problems in AI generated responses, verification and evaluation of AI outputs, privacy and security considerations while using AI systems, responsible prompt design practices, limitations of generative AI technologies, future trends of prompt engineering and AI assisted engineering applications.

Case Study: A healthcare organization uses AI to generate patient treatment recommendations. Some AI outputs are biased or factually incorrect, posing ethical risks. Task for Students: Create prompts and verification steps to minimize bias, ensure factual accuracy, and maintain ethical AI usage. Discuss limitations of AI in sensitive applications.

Learning Resources:

Text Books

2. Vladimir Geroimenko, “The Essential Guide to Prompt Engineering: Key Principles, Techniques, Challenges, and Security Risks”, SpringerBriefs in Computer Science (2025), ISBN: 978-3-031-86205-2.

Reference Books:

1. Suraj Singh, “Prompt Engineering Handbook: Unlocking the Power of Generative AI”, Notion Press (2025), ISBN: 9798897777969.
2. James Phoenix, Mike Taylor, “Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs”, O’Reilly Media, Inc. (2024), ISBN: 9781098153434.

MOOC / NPTEL/YouTube Links:

1. https://cloud.google.com/discover/what-is-prompt-engineering
2. https://www.udemy.com/course/prompt-engineering-certification-master-ai/

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC362DCAI : UI/UX (User Interface & User Experience Design)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester**: 70 Marks

Course Objectives: The course aims to

1. To understand principles of user-centered interface design.
2. To apply visual and interaction design techniques for digital interfaces.
3. To analyze user requirements using UX research methods.
4. To develop wireframes and prototypes for applications.
5. To explore modern UI/UX tools and industry practices.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain fundamental concepts of UI/UX and user-centered design principles.
- CO2: Apply human-computer interaction and visual design principles in interface design.
- CO3: Analyze user needs using UX research techniques and information architecture.
- CO4: Develop wireframes and interactive prototypes using modern design tools.
- CO5: Evaluate modern UI/UX trends, tools, and industry practices for designing effective digital products.

Unit I -Introduction to UI/UX Design - (09 Hours)

What is UI/UX Design: Importance of User-Centric Design, Goals of User Interface Design , Design Thinking Process, Core Principles, Role of UX in Product Development Lifecycle, mental and conceptual model.

Usability Design Principles: 4 Design Principles, Schneiderman's Golden Rules, Gestalt Principles of Design, Visual Design Principles, Form versus Function, Metaphors, Idioms and Affordances in UI design, User Interface Elements: Input Controls, Navigation Components, Information Components, Containers User Research: Qualitative and Quantitative User Research, Behavioral and Attitudinal User Research, Use of Personas, User Stories and Scenarios, Affinity Mapping.

Case Study: A good and a bad User Interface Design

Unit II - Usability Engineering, Evaluation and Testing (09 Hours)

Usability Engineering: Concept of usability, usability principles, benefits of usability for users and organizations, internationalization and localization, human errors and their impact on usability. Us-ability Evaluation: Human information processing and memory, Fitts's Law and Hick's Law, usability inspection methods such as Heuristic Evaluation and Cognitive Walkthrough, user studies and field studies. Usability Testing: Planning and conducting usability testing, Think-Aloud testing, A/B test-ing, use of heatmaps, and basic UX metrics for evaluating user experience. UX Design Foundations: Ideation and research in UX design, content and interaction mapping, paper prototyping, introduction to wireframes and interface layout, applying Nielsen's usability heuristics, refining UI based on user feedback.

Unit III - WEB DESIGN: STRATEGIES AND INFORMATION ARCHITECTURE (09 Hours)

User-centric design - The UX Phases - Waterfall vs. Agile - Web vs. App. Determining Strategy: User Research - Inspiration - Analytics - User Needs and Client Needs - Target Audience - What is in and What is Out: Outlining Scope - Content and Functionality. The Sitemap: Introduction to Sitemaps - Information Architecture - Sitemap Concerns - annotated process - Elements - Treejack Introduction - Treejack Analysis.

Unit IV - Wireframing Fundamentals - (09 Hours)

Introduction to wireframes, purpose and importance in UX design, low-fidelity vs high-fidelity wireframes, sketching interfaces, layout grids, design hierarchy, and responsive layouts. Prototyping Concepts: Introduction to prototypes, types of prototypes (paper prototypes, digital prototypes, interactive prototypes), fidelity levels, advantages of prototyping in UX design, iterative design process. Inter-action Design: Principles of interaction design, micro-interactions, feedback mechanisms, navigation flows, usability considerations in interaction design, designing intuitive interfaces. UI Design Tools: Introduction to modern UI/UX design tools such as Figma, Adobe XD, Sketch, and InVision. Creating interactive prototypes, collaborative design workflows, design handoff to developers. Design Docu-mentation: User flow diagrams, storyboards, design specifications, and design system basics.

Unit V - Designing UX for Tomorrow - (09 Hours)

Emerging technologies in UX Design: Voice UI, Touchless gesture control, Intelligent UX, Conversa-tional UX, Immersive Media and Fluid UX Designing for Web and Mobile Interfaces, IoT applications, Industry Specific UX Design: FinTech, Education, Health Care, E-commerce and Industrial Websites, designing for Wearable Devices, designing for Augmented Reality, Virtual Reality and Mixed Reality, Tomorrow's challenges in UX design.

Learning Resources:

Text Books:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos – Designing the User Interface: Strategies for Effective Human-Computer Interac-tion, Pearson.
2. Don Norman – The Design of Everyday Things, Basic Books.
3. Russ Unger and Carolyn Chandler – A Project Guide to UX Design, New Riders.

Reference Books:

1. Steve Krug – Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders.
2. Alan Cooper, Robert Reimann, David Cronin – About Face: The Essentials of Interaction Design, Wiley.
3. Jesse James Garrett – The Elements of User Experience, New Riders.
4. Jeff Gothelf & Josh Seiden – Lean UX, O'Reilly.

Mooc / NPTEL / You Tube Link:

1. Google UX Design Professional Certificate – Coursera
2. Interaction Design Specialization – Coursera (UC San Diego)
3. User Experience Design Fundamentals – Udemy
4. Introduction to User Experience Design – Georgia Tech (Coursera)
5. Figma UI/UX Design Essentials – Udemy

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC363ACAI : Blockchain Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	01	**Term Work:** 25 Marks **Practical** : 25 Marks

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. Dur-ing practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, trans-parent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So,

adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction - Block chain

List of Assignment - Group A

1. Design and implement a Python program to demonstrate the working of the SHA-256 hashing algorithm and analyze the avalanche effect.
2. To implement Elliptic Curve Cryptography for generating and verifying digital signatures
3. To implement and simulate a basic blockchain data structure using Python in order to understand the concept of block creation, SHA-256 hashing, and linking of blocks to form a secure and immutable blockchain.
4. To design and implement a blockchain-based system for tracking different versions of an Artifi-cial Intelligence (AI) model in order to ensure secure, transparent, and tamper-resistant model updates.
5. To implement a blockchain system with a Proof of Work consensus mechanism to understand the process of mining and block validation.
6. Installation of MetaMask, create your own wallet using MetaMask for crypto transactions and study spending Ether per transaction
7. Write a smart contract on a test network for the Bank account of a customer for following oper-ations.

(a) Deposit Money

(b) Withdraw Money

(c) Show Balance

8. Write a program in solidity to create Student data. Use the following constructs:

(a) Structures

(b) Arrays

(c) Fallback

Deploy this as a smart contract on Ethereum and Observe the transaction fee and Gas value.

9. Develop a Blockchain based application dApp (de- centralized app) for e-voting system.

Mini Project (Open Ended) - Title: Blockchain-Based Decentralized Application

Concepts to be Implemented: Students must incorporate at least four of the following:

1. SHA-256 or cryptographic hashing for data integrity,
2. Smart contract development using Solidity deployed on Ethereum testnet,
3. Consensus mechanism simulation (PoW, PoS, or Snow/Avalanche),
4. Wallet creation and transaction management using MetaMask,
5. Decentralized data storage using IPFS
6. Integration of blockchain with an AI/ML model for secure data provenance or model versioning.

7. Sample Application Domains (not limited to): Decentralized voting system, Supply chain prod-uct tracking, Blockchain-based academic certificate verification, Healthcare record management, NFT-based digital asset marketplace.

   Deliverables: Problem statement, system design, source code, deployed contract address on testnet, demonstration, and a brief report. Tools: Solidity, Remix IDE / Hardhat / Truffle, MetaMask, Python, Web3.js, IPFS (optional)

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC363BCAI : Big Data engineering Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	**01**	**Term Work:** 25 Marks **Practical :** 25 Marks

1. Big Data Characteristics Analysis: Design a system that collects college data such as attendance, results, LMS logs, and feedback and analyzes it using Big Data tools to handle Volume, Velocity, and Variety of data efficiently.
2. Large-Scale Social Media Data Analyzer: Develop a system that processes structured, semi-structured, and unstructured social media data and classifies posts based on topics or sentiment using dis- tributed processing.
3. Distributed File Storage System using HDFS: Implement a Hadoop-based distributed storage system to store large datasets and demonstrate data replication, fault tolerance, and block storage mechanism.
4. E-Commerce Transaction Analysis using MapReduce: Build a MapReduce application that pro- cesses large e-commerce transaction datasets to generate insights such as top-selling products, customer purchase patterns, and sales statistics.

List of Assignment - Group B (Any TWO)

1. Real-Time Log Analysis using Apache Spark: Create a Spark-based system that analyzes server or application logs in real time to detect errors, anomalies, and system performance issues.
2. Big Data Movie Recommendation System Using Spark: Develop a Spark-based recommendation engine that analyzes large movie rating datasets and suggests movies based on user preferences.
3. Real-Time Fraud Detection System for Banking: A food delivery app receives thousands of live orders per minute. Design a Kafka-based data ingestion pipeline to handle high-velocity streaming data.
4. Smart City Data Analytics Platform: Develop a system that integrates transport, energy, and environmental datasets and uses Big Data analytics to provide insights for smart city planning and resource management.
5. Design and implement a secure Big Data storage system where sensitive data is stored in encrypted format and accessed through controlled authorization mechanisms. Apply authentication and authorization (RBAC/ABAC)
6. Develop a Big Data governance model that ensures data privacy, compliance, and ethical usage of data.

List of Assignment - Group C

1. Large-Scale Data Deduplication System: Design a Big Data application that processes large datasets stored in HDFS and identifies duplicate records efficiently using distributed processing frameworks such as MapReduce or Spark.

2. Distributed Data Processing for Weather Data Analysis: Develop a system that collects and analyzes large historical weather datasets to identify climate trends and patterns using Hadoop and Spark analytics tools.

3. Big Data Based Online Advertisement Analysis: Create a system that processes large-scale online advertisement clickstream data to analyze user engagement, click-through rates, and advertising performance.

4. Energy Consumption Analytics using Big Data: Design a platform that collects and processes energy usage data from multiple smart meters and analyzes consumption patterns to support efficient energy management.

5. Big Data Pipeline for Data Integration and Processing: Develop a data pipeline architecture that collects data from multiple sources, performs ingestion using tools such as Flume or Sqoop, stores it in HDFS, and processes it using Apache Spark.

Mini Projects:

1. Real-Time Social Media Sentiment Analysis: Develop a Big Data application that collects social media data (tweets or posts) and analyzes user sentiment (positive, negative, neutral) using Apache Spark and streaming tools.

2. E-Commerce Product Recommendation System: Design a scalable system that analyzes large cus- tomer purchase datasets and recommends products based on user behavior and purchase history using Spark analytics.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC363CCAI: High Performance Computing Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory: 02 Hours/Week	01	**Term work**: 25 Marks **Practical**: 25 Marks

Laboratory sessions should be conducted in alignment with the approved HPC theory syllabus.

1. Required software tools (OpenMP, MPI, CUDA, Spark, etc.) must be properly installed and tested before conducting practical's.
2. Each experiment should include brief explanation of theory followed by hands-on implementation.
3. Students should compare sequential and parallel versions of programs wherever applicable to analyze performance.
4. Emphasis should be given to performance metrics such as execution time, speedup, and efficiency.
5. Continuous assessment should be carried out based on implementation, understanding, and timely submission of assignments

List of Assignment

1. Install and configure OpenMP/MPI environment. Write a simple C/Python program to measure execution time of a sequential program and analyze performance.
2. Implement a CPU-based parallel program (e.g., array sum or matrix addition using OpenMP). Calculate speedup and efficiency for different number of threads.
3. Implement parallel matrix multiplication or sorting using OpenMP directives. Compare sequential vs parallel execution time.
4. Implement parallel matrix addition or multiplication using MPI (send, receive, broadcast). Analyze communication overhead.
5. Implement a computational problem (e.g., sorting, numerical computation, or array processing) using both OpenMP and MPI. Analyze and compare execution time, speedup, and scalability of the two programming models.
6. Write a simple CUDA program (vector addition or matrix addition). Compare CPU vs GPU execution time.
7. Implement a simple deep learning operation (matrix multiplication or convolution using Py-Torch/TensorFlow with GPU). Measure training time with and without GPU.
8. Implement a basic MapReduce task (word count or log analysis) using Apache Spark. Analyze performance for different data sizes.
9. Mini Project: HPC-based AI or Data-Intensive Application: Develop and implement a small project using HPC concepts such as parallel programming, GPU computing, or distributed processing.

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
PEC363DCAI : AR/VR Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	01	**Term work**: 25 Marks **Practical**: 25 Marks

1. To understand the working of immersive hardware devices (Study HMDs, VR controllers, depth sensors, motion trackers.)
2. Creating a Basic 3D Scene in Unity :
    - Install Unity and required SDK
    - Create a 3D virtual room/environment.
    - Add geometric objects, lighting and textures.
    - Apply camera controls.
3. Geometric Transformations and Camera Navigation - Implement translation, rotation and scaling of objects.
4. Virtual Object Interaction using Controllers - To implement interaction techniques in VR.
5. Gesture-Based Interaction System - To design natural user interfaces (Capture hand gestures using webcam/sensors)
6. Marker-Based Augmented Reality Application - To implement basic AR tracking

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
MDM371-CAI : Robotics and Automation		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 01 Hours/Week **Practical:** 02 Hours/Week	02	**Term Work:** 25 Marks **Oral :** 25 Marks

Prerequisite Courses: Engineering Physics, Engineering Mathematics, Basics of Electrical Engineering, Basics of Electronics Engineering & Engineering Graphics.

Course Objectives: The course aims to:

1. To introduce various types of Robots and the functional elements of Robotics.
2. To impart knowledge of robot drive systems & educate on various sensors used in Robotic automation.
3. To introduce various types the end effectors.
4. To impart knowledge of basics of Robot Programming and robotic Applications

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Understand basic concepts of robotics.
- CO2: Select appropriate drive & sensors for Robotic applications.
- CO3: To Compare and Select robot and end effectors as per application.
- CO4: To know about the fundamentals of robot programming and applications.

Course Content

Unit I - Fundamentals of Robotics (03 Hours)

Definitions of Industrial Robot, Type and Classification of Robots, Asimov ‘s laws of robotics, Robot configurations, Robot Components, Robot Degrees of Freedom, Work volume and work envelope, Robot Joints and symbols, Robot Coordinates

Unit II - Sensors (03 Hours)

Transducers and sensors, Sensors in robotics, Principles and applications of the following types of sensors- Proximity Sensors, Photoelectric Sensors, Position sensors – Piezoelectric Sensor, LVDT, Resolvers, Encoders – Absolute and Incremental: - Optical, Magnetic, Capacitive, pneumatic Position Sensors, Range Sensors- Range Finders, Laser Range Meters, Touch Sensors, Force and torque sensors

Unit III - Industrial Automation and AI in Robotics (03 Hours)

Introduction to industrial automation PLC basics and SCADA overview

IoT in automation systems

Basics of AI in robotics (computer vision, ML concepts) Case studies: smart factories, autonomous robots

Unit IV Autonomous Navigation & Path Planning (03 Hours)

Environment Representation: Occupancy grids, topological maps, and configuration spaces. Path Planning Algorithms: Dijkstra's Algorithm, A* (A-star) search, and RRT (Rapidly exploring Random Trees). Localization: Odometry, sensor fusion (Kalman Filters), and particle filters.

Obstacle Avoidance: Potential field methods and reactive control loops.

Unit V Robotic Middleware & Industry 4.0 (03 Hours)

Robot Operating System (ROS): Architecture (Nodes, Topics, Services, Messages) and Workspace management. Simulation Environments: Working with Gazebo and RViz for testing and validation. Industrial Automation: Introduction to PLC (Programmable Logic Controllers) and SCADA systems. Industry 4.0: Impact of IoT, Big Data, and AI in smart manufacturing and automation.

Learning Resources:

Text Book:

1. Robotics: Control, Sensing, Vision and Intelligence – McGraw Hill
2. Introduction to Robotics – Tata McGraw Hill
3. Robotics Engineering – Prentice Hall
4. Programmable Logic Controllers – McGraw Hill

Guidelines for Laboratory Conduction

List of Assignment - Group A (Any SIX)

1. Study of different robotic components, joints, links, and end effectors
2. Interfacing and control of DC motors using Arduino
3. Servo motor position control using PWM signals

4. Obstacle detection robot using ultrasonic sensor
5. Design and implementation of line follower robot
6. Interfacing IR sensors and proximity sensors with microcontroller
7. Stepper motor control for robotic arm movement
8. Basic PLC programming using ladder logic for industrial automation
9. IoT-based automation system using sensors and cloud monitoring

Group B - Mini-project: Design of a simple autonomous robotic or automation application - Suggested List

1. Smart warehouse robot
2. Automated parking system
3. IoT-based industrial monitoring system
4. Pick-and-place robotic arm
5. Smart conveyor automation
6. Home automation using sensors and actuators

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
VSE372-CAI: Solar Technology and Development		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical: 02 Hours/Week	02	**Term Work:** 50 Marks

Prerequisites: Basic knowledge of Physics (especially topics like electricity, magnetism semiconductors, light/energy concepts, Basic Electrical Engineering or Basic Electronics, Engineering Mechanics. Heat and energy concepts

Course Objectives: The course aims to:

1. Apply Safely install, wire, and commission basic solar PV systems while measuring key performance parameters.
2. Analyze Break down the impact of environmental and operational factors on solar system efficiency and diagnose common faults.
3. Evaluate Judge the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
4. Create Develop simple practical solutions or documentation for improving solar system performance in mini projects.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply safe installation, wiring, commissioning, and performance measurement of basic solar PV systems.
- **CO2:** Analyze the impact of environmental/operational factors on solar PV efficiency and diagnose common faults.
- **CO3:** Evaluate the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
- **CO4:** Create simple practical solutions or documentation for improving solar system performance via mini projects.

Practical Assignments

1. Experiments No 1, 2 and 10 are compulsory.
2. Perform 2 Experiments from 3 to 5
3. Perform any 3 Experiments from 6 to 9

1. Measurement of solar irradiance using pyrometer/lux meter at different times/angles. Real-World Assignment: Survey irradiance on your college rooftop for one day. Calculate daily energy generation for a 100W panel and suggest best installation time/angle for maximum output.
2. Plot I-V and P-V characteristics of solar PV module under varying light & temperature. Real-World Assignment: Simulate cloudy/rainy day conditions. Calculate module efficiency and estimate annual energy loss in Pune climate.

3. Survey and Comparative Analysis of Solar PV Installation Systems: Grid-Tied, Hybrid, and Off-Grid Configurations. Real-World Assignment: Survey 2–3 real solar installations (e.g., college rooftop, nearby home/business, or online/virtual
4. Series and parallel connection of PV modules, observe mismatch issues. Real-World Assignment: Design a small array for 12V/24V system (e.g., for laptop charging or lab fan). Calculate total power and suggest fuse/ diode protection for mismatch in a multi-panel rooftop installation.
5. Installation and wiring of standalone solar PV system (PV \Charge controller Battery (Load/Inverter) Real-World Assignment: Prepare a complete wiring diagram and BOM for a 100W system to power a college water cooler or hostel room. Include safety and cable sizing as per real IEC standards.
6. Preventive maintenance: Cleaning, visual inspection, corrosion/loose connection check. Real-World Assignment: Inspect any existing solar panel in college/hostel. Prepare a 6-month maintenance schedule with cost estimation (dust cleaning, tightening)
7. Grid-Related Maintenance Checks for Grid-Tied Solar PV Systems: Inverter Health, Performance Monitoring, and Fault Diagnosis. Real-World Assignment: Survey a real grid-tied installation, Prepare a maintenance schedule: Monthly inverter check, quarterly visual, annual professional inspection.
8. Mounting structure assembly: Rooftop/ground mount, tilt adjustment, stability check Real-World Assignment: Design a simple mounting frame for windy Pune conditions. Calculate wind load and suggest material/cost for a 5kW residential installation.
9. IoT-Based Real-Time Solar PV System Monitoring and Performance Dashboard.
10. Industrial Visit to Solar Energy Facility in Pune Region: Hands-On Learning of Solar PV System Operations and Maintenance

Learning Resources:

Textbook:

1. S.P. Sukhatme, Solar Energy
2. C.S. Solanki, Solar Photovoltaics
3. D.P. Kothari et al., Renewable Energy Sources
4. G.D. Rai, Non-Conventional Energy Sources
5. H.P. Garg, Solar Energy Utilization

Reference Books:

1. Solar PV System: Design, Installation, Operation and Maintenance Authors: L. Ashok Kumar and
   K. Mohana Sundaram.
2. Solar Engineering of Thermal Processes, Photovoltaics and Wind (5th Edition) Authors: John A. Duffie, William A. Beckman (updated with Nathan Blair).
3. Principles of Solar Engineering (3rd Edition) Authors: D. Yogi Goswami, Frank

Kreith, Jan F. Kreide .

NPTEL/MOOC Courses:

1. Solar Photovoltaics: Fundamentals, Technology and Applications: https://onlinecourses.nptel.ac.in/noc
2. SkillCat or Other Free Solar Training (Installation Focus). https://www.skillcatapp.com/solar-installation-training

Savitribai Phule Pune University Third Year of Computer Science and Engineering (Artificial Intelligence) - 2024 Pattern		
ELC381-CAI: Internship		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 08 Hours/Week	04	**Oral :** 50 Marks

Course Objectives: The course aims to:

1. To expose students to real-world industry practices.
2. To bridge the gap between academic learning and practical implementation.
3. Develop professional competency, ethics, communication, and teamwork skills.
4. To encourage self-learning and problem-solving abilities.
5. Encourage innovation, entrepreneurship, and research aptitude.

Course Outcomes: Upon successful completion of this course, students will be able to:

1. CO1: Apply theoretical knowledge to solve real-world engineering problems.
2. CO2: Demonstrate technical competency in tools/technologies used in industry.
3. CO3: Exhibit professional ethics and teamwork.
4. CO4: Prepare technical reports and deliver effective presentations on industrial training experience
5. CO5 Analyze industrial processes and suggest feasible improvements or innovations.

Student Guidelines

Students should opt for a internship/JOT that would provide them to gain ample field knowledge in the relevant field of engineering such that theoretical knowledge gained in the class can be applied to solve the practical/ field problem.

1. Students must have to opt for technical internship after VI semester and before VII semester, preferably during summer break.
2. Undergoing a training program / Course at a particular organization for specified dura- tion is NOT

considered as summer internship

3. However, students can attend such programs mentioned above to learn new tools for short duration that would help to solve the problem undertaken in the internship

4. Students should take a challenging task, maybe a small portion, and apply the knowledge gained to solve it.

5. Internships can also involve data collection from different sources, including generating experimental data, collection of data from field etc. The data may be analyzed later on.

6. Different central and state government organizations, CSIR labs, premier institutions like IITs and IIMs, DRDO, public sector undertaking organizations, and top IT companies may be considered for internships.

7. Students need to submit Synopsis, Permission letter and offer letters to Internship coordinator before proceeding to internship.

8. Internship completion will be considered only after submission of valid documents at the end of internship like Completion certificate, Report and presentation of work done, feedback from industry etc.

9. Student will appear for term work evaluation where he/she will present the work done before mentor(s) at the end of internship.

Suggested Internship Activities:

- Students are expected to perform the following activities during internship:
- Phase I – Orientation and Requirement Study
  - Understanding organization structure
  - Study of workflow and operational processes
  - Requirement analysis and project allocation
  - Understanding tools and technologies used
- Phase II – Technical Learning and Development
  - Coding and implementation
  - Database design and integration
  - Software testing and debugging
  - API integration and deployment
  - Use of version control systems
  - Documentation practices

- Phase III – Project Execution
  - Module development
  - Testing and validation
  - Performance optimization
  - Client interaction (if applicable)
  - Team collaboration
- Phase IV – Documentation and Presentation
  - Preparation of internship report
  - Preparation of project demonstration
  - Final presentation and viva voce

Deliverables

- Internship Joining Report
- Weekly Logbook
- Mid-term Progress Report
- Supervisor Feedback (Initial)

Internship Structure

The internship may be carried out in any one of the following domains:

- Software Development
- Artificial Intelligence and Machine Learning
- Data Science and Analytics
- Cloud Computing and DevOps
- Cyber Security
- Web and Mobile Application Development
- IoT and Embedded Systems
- Networking and System Administration
- Automation and Robotics Software
- Research and Development
- Entrepreneurship and Startup Projects
- Government/NGO Technical Projects

Nature of Internship

Students shall undergo internship/training in one of the following:

- Registered companies / startups
- Government organizations
- Research institutions
- Recognized industry-academic collaborative projects
- Internships may be conducted in offline, online, or hybrid mode, subject to proper approval and verification.

Guidelines for Internship Report Writing

1.Preliminary Pages

- Cover Page
- Certificate from Organization
- Certificate from Department
- Acknowledgement
- Abstract
- Table of Contents

2.Chapter 1 – Organization Profile

- Company overview
- Vision and mission
- Products/services
- Organizational structure

3.Chapter 2 – Problem Statement and Objectives

- Project title
- Need of project
- Objectives
- Scope

4.Chapter 3 – Technologies and Methodology

- Software/hardware tools used
- Development methodology
- System architecture
- Database design

5.Chapter 4 – Work Carried Out

- Tasks completed

- Screenshots/results
- Challenges faced
- Solutions implemented

6.Chapter 5 – Learning Outcomes

- Technical learning
- Professional skills acquired
- Industry exposure
- Future scope

7.Chapter 6 – Conclusion

- Summary of work
- Achievements
- Suggestions

References: IEEE format references preferred Appendices

- Source code snippets
- Certificates
- Additional screenshots

Learning Resources

Text Books:

1. W. J. King and James G. Skakoon , The Unwritten Laws of Engineering , ASME Press
2. Stuart Walesh, Engineering Your Future: The Professional Practice of Engineering
3. Eliyahu M. Goldratt, The Goal: A Process of Ongoing Improvement
4. AICTE Internship policy : AICTE Internship Policy: Guidelines & Procedures
5. AICTE Internship Portal : https://internship.aicte-india.org

Savitribai Phule Pune University, Pune

Maharashtra, India

Task Force for Curriculum Design and Development Third Year Computer Science and Engineering (Artificial Intelligence)- 2024 Pattern

Programme Coordinator

Dr. Sachin D. Babar - Member, Board of Studies – Computer Science Engineering (Artificial Intelligence)

Core Committee Members

Dr. Sachin Babar	Indira College of Engineering & Management, Pune (ICEM)
Dr. Sagar Shinde	Nutan Maharashtra Institute of Engineering and Technology
Prof. Priyanka Abhay Vyas	Nutan Maharashtra Institute of Engineering and Technology
Dr. Prashant Ahire	Nutan Maharashtra Institute of Engineering and Technology
Dr.Dhanashree Kulkarni	Nutan Maharashtra Institute of Engineering and Technology
Dr.Dikshendra Sarpate	ISBM College Of Engineering Pune
Dr. Girish Potdar	PICT,Pune

Compiled By

Prof. Priyanka Abhay Vyas, Nutan Maharashtra Institute of Engineering and Technology.

Team Members for Course Design

Networks & Security	
Dr. Vishal Puranik	Ajeenkya DY Patil School of Engineering
Dr. Sagar Shinde	Nutan Maharashtra Institute of Engineering and Technology
Dr Mohammed S Baig	Maulana Mukhtar Ahmad Nadvi technical Campus
Prof. Chandan Wagh	Dr D.Y.Patil College Of Engineering and Innovation, Talegaon, Pune
Prof. Pooja Prasad Kajale	DVVP COE Ahilyanagar
Prof. Dipali Shinde	PVG COE & SSD IOM Nashik
Mr. Tushar B. Kute	MITU Research, Pune

Natural Language Processing	
Dr. Vinay S. Nalawade	S B Patil College of Engineering,Indapur
Dr. Rama Gaikwad	Anantrao Pawar college of Engineering and Research
Dr. Shwetal Patil	MMIT
Dr. Sujeet More	Trinity College of Engineering and Research
Prof. Abha Pathak	Dr. D. Y. Patil College of Engineering and Innovation Varale

Prof.Anjali Phaltane	Dr.Vithalrao Vikhe Patil College of Engineering, Ahilyanagar
Mr. Anshuman Jadhav	Independent Consultant

Internet of Things	
Dr. Kalpana Metre	ITMBU,Vadodara,Gujrat
Dr. Dhanashree Kulkarni	Nutan Maharashtra Institute of Engineering & Technology
Prof. Charushila Patil	GGS College of Engineering And Research Centre, Nashik
Dr.Pramod Dhamdhere	MMIT Lohegaon Pune
Prof. Manasi D. Karajgar	DYPCOE, AKURDI, PUNE
Dr.Saloni N.Shah	SPCOET,Someshwar
Mr.Piyush Sonewar	Pixaflip Technologies Pvt.Ltd.,Pune

Deep Learning	
Prof. Vaibhav Suryawanshi	Nutan Maharashtra Institute of Engineering & Technology
Dr. Minakshi Pradeep Atre	PVG,Pune
Dr. Shraddha Pandit	Modern COE,Pune
Dr.Jitendra Musale	Anantrao Pawar college of Engineering and Research
Dr. Dipti D. Patil	Cummins College of Engineering for Women,Pune
Mr.Kedar Kulkarni	Apple Inc.,Austin, Texas

Cyber Security	
Dr. Chhaya Gosavi	Cummins COEW,Pune
Dr. Bhagyashree Dhakulkar	Ajeenkya DY Patil School of Engineering, Lohegaon Pune
Dr. Bajirao Subhash Shirole	Loknete Gopinathji Munde Institute of Engineering Education and Research, Nashik
Prof. Sarita Charakha	Nutan Maharashtra Institute of Engineering and Technology,Pune
Mr.Venkatta Satish Gutulla	Independent Consultant

Computer Vision	
Dr. Vinay S. Nalawade	S B Patil College of Engineering,Indapur
Dr Aarti Dandavate	G.H.Raisoni College of Engineering & Management,Pune
Dr. Neeta Maitre	Cummins College of Engineering for Women,Pune
Dr. Suvarna Pawar	Trinity Academy of Engineering,Pune
Prof. D.S.Bhadane	Ajinkya D Y Patil University,Pune
Prof. S.D.Pawar	Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering
Mr. Pradip D.Kulkarni	Visdin Solutions

Big Data Engineering	
Prof. Priyanka Vyas	Nutan Maharashtra Institute of Engineering and Technology,Pune
Dr. Dikshendra Sarpate	ISBM College Of Engineering Pune
Dr. Shraddha Khonde	Wadia College of Engineering,Pune
Prof. Mangal Kale	Dr. Vithalrao Vikhe Patil COE, Ahmednagar
Prof. Vivek Badgujar	JES Institute of Technology Management & Research, Nashik
Prof. Alka Kumbhar	GSMCOE balewadi
Dr.Javed Shaikh	Capital Numbers Infotech PVT LTD

High Performance Computing	
Dr. Girish Potdar	PICT,Pune
Prof. Seema Mahalungkar	Nutan Maharashtra Institute of Engineering and Technology, Pune
Dr. Dewanand Meshram	RMD Sinhgad school of Engineering, Pune
Prof.Anand Nandlal Gharu	MET's Institute of Engineering, Bhujbal Knowledge City, Nashik
Prof. Prasad Ashokrao Lahare	PVG,s COE & SSD IOM,Nashik
Prof. Sandeep Jadhav	Sanghavi College of Engineering, Nashik
Trambak Ramesh Pawar	Telus Digital

Ethical Hacking	
Dr. Dikshendra Sarpate	ISBM College Of Engineering, Pune
Dr Prakash Gadekar	Marathwada Mitra Mandals College of Engineering,Pune
Prof.Suchitra S.Pakale	Cummins College of Engineering for Women,Pune
Prof. Sagar Dhanake	Dr. D Y Patil College of Engineering and Innovation
Prof. Vishal Naykwadi	VIIT Pune
Prof Meenal Kamlakar	Cummins College of Engineering for Women,Pune
Mr.Sachin Kulkarni	flydubai

Gaming & Animation	
Dr. Nuzhat Faiz Shaikh	Wadia COE
Prof. Harsha Sonune	Cummins College of Engineering for Women,Pune
Dr.Shabnam Sayyad	AISSMS COE,Pune
Dr. Poonam Railkar	SKNCOE,Pune
Prof.Mahesh Suhas Shinde	VIT,Pune

Quantum AI	
Dr. Shraddha Pandit	Modern COE,Pune
Prof. Prashant Bhagirath Koli	R. H. Sapat College of Engineering,Nashik
Dr.Archana Chaudhari	Symbiosis Institute of Technology
Dr.Ritu Dudhamal	MIT,Alandi
Dr. Darshan Medhane	Kakasaheb Wagh Institute of Engineering,Nashik
Dr.Manjusha Tatiya	Indira College of Engineering and Management,Pune

Reinforcement Learning	
Dr. Minakshi Pradeep Atre	PVG,Pune
Prof. Geeta M Kodabagi	Ajeenkya DY Patil school of engineering Lohegaon pune
Dr. Prajwal Sopan Gaikwad	AISSMS IOIT
Dr. Ravindra Apare	PVG's COETM,Pune
Prof. Hitendra S. Khairnar	Cummins College of Engineering for Women,Pune
Prof. Dipika Paranjape	Nutan Maharashtra Institute of Engineering and Technology,Pune
Mr. Sushrut Guruji	Wipro Technology Pune

Machine Learning	
Dr. Shraddha Pandit	Modern COE,Pune

Prof. Priyanka Vyas	Nutan Maharashtra Institute of Engineering and Technology,Pune
Prof. Dipika Paranjape	Nutan Maharashtra Institute of Engineering and Technology,Pune
Dr. Girisha Bombale	SNDCOE & RC,Yeola
Prof. Pradnya Bachhav	Guru Gobind Singh College of Engineering & Research Centre
Prof. Varsha P.Chavan	Anantrao Pawar College of Engineering and Research
Mr. Nachiket Kulkarni	AutoDesk, Pune

Chairman

Dr. Nilesh Uke - Board of Studies Computer Science Engineering (Artificial Intelligence)

Savitribai Phule Pune University, Pune

Dean

Dr. Raosaheb Latpate- Dean – Science and Technology

Savitribai Phule Pune University, Pune.