Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Deep Learning -II
Course Code	AM-601
Semester	6
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

- Proficiency in Python programming and common ML libraries (e.g., NumPy, PyTorch/TensorFlow)

- Fundamental understanding of linear algebra, calculus, and probability/statistics

- Introductory machine learning concepts including supervised learning, basic neural networks, and back-propagation

2. Course Learning Objectives

- Equip students with a deep conceptual understanding of modern sequence-modeling techniques--from traditional word embeddings to advanced Transformer architectures--and the ability to articulate why and when each approach is appropriate.

- Develop students' practical skills in designing, implementing, and debugging recurrent and attention-based neural networks using contemporary deep-learning frameworks, including hands-on experience with preprocessing, training tricks, and model optimisation.

- Enable learners to apply sequence-modeling and generative-AI methods to real-world problems across text, vision, and audio domains, and to critically evaluate model performance, interpretability, and deployment constraints.

• Introduce foundational reinforcement-learning concepts and ethical considerations, fostering the ability to build and assess deep RL agents while recognizing societal impacts, fairness, and safety issues.

• Cultivate students' capacity to communicate technical ideas effectively, collaborate on AI projects, and adopt responsible AI practices throughout the model development lifecycle.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Foundations of Sequence Modeling & Word Representations** - Introduction to sequence modeling and why it matters for AI - Text preprocessing and tokenization (including	6

	sub-word units such as BPE) - One-hot encoding vs. dense distributed embeddings - Word2Vec (skip-gram & CBOW) and GloVe fundamentals - Contextual word representations (e.g., ELMo) - concept only - Positional embeddings - intuition and simple implementations	
2	**Recurrent Neural Networks and Gated Architectures** - Basic RNN architecture and unfolding over time - Conceptual view of back-propagation through time (BPTT) - Vanishing / exploding gradient intuition and simple mitigation (gradient clipping) - Long Short-Term Memory (LSTM): gates and practical training tips - Gated Recurrent Units (GRU) - Bidirectional RNNs and when to use them - Stacked (deep) RNNs and regularisation techniques (drop-out, layer norm)	7
3	**Sequence-to-Sequence, Attention, and Transformers** - Encoder-decoder (seq2seq) framework and its applications - Teacher forcing and beam-search decoding basics - Additive (Bahdanau) and multiplicative (Luong) attention mechanisms - Self-attention and alignment models - Transformer architecture: key, query, value vectors - Multi-head attention, residual connections, layer-norm, and positional encoding - Practical advantages of Transformers over RNNs and common implementation tips	8
4	**Applications and Interpretability of Sequence Models** - Machine translation and text summarisation use-cases - Vision Transformers for image classification and detection - Audio-signal processing with Transformers (e.g., speech recognition) - Interpretability techniques: attention visualisation, saliency maps, probing classifiers - Real-world case studies that link model design choices to problem requirements	6

	- Deployment considerations (latency, model size, on-device inference)	
5	**Generative Modeling Techniques** - Overview of generative modelling goals and evaluation criteria - Autoencoders: encoder-decoder structure and reconstruction loss - Variational Autoencoders - intuitive view of latent space and regularisation (no heavy math) - Generative Adversarial Networks: generator-discriminator game and minimax idea - Popular GAN variants: DCGAN, Wasserstein GAN, Conditional GAN - Practical training tricks: stabilisation, mode-collapse mitigation, evaluation metrics - Emerging lightweight generative ideas (e.g., diffusion models - high-level concept only)	7
6	**Reinforcement Learning and Ethical AI** - Markov Decision Processes: states, actions, rewards, policies (conceptual) - Value functions and the Bellman idea - intuition without detailed derivations - Policy-based vs. value-based methods: Q-learning, SARSA, policy gradients, actor-critic - Deep Reinforcement Learning: Deep Q-Networks (DQNs) and policy-gradient with neural nets - Exploration strategies and safety considerations (e.g., reward shaping) - Ethics, fairness, and societal impact of AI systems - Interpretability for RL agents and responsible AI practices	8

6. References

Textbooks:

1. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into Deep Learning. Cambridge University Press.

2. Prince, S. J. D. (2023). Understanding Deep Learning. MIT Press

Reference Books:

1. Foster, D. (2023). Generative Deep Learning (2nd ed.). O'Reilly Media.

2. Chollet, F., & colleagues. (2019). Advanced Deep Learning with Python. Packt Publishing.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-601.1	Recall and describe the fundamental concepts of sequence modeling, including tokenization, one-hot encoding, dense embeddings, and the intuition behind contextual word representations such as ELMo.	Recall	Remember
AM-601.2	Apply preprocessing pipelines to raw text data by implementing sub-word tokenization (e.g., BPE) and generating dense word embeddings (Word2Vec or GloVe) for use in downstream models.	Apply	Apply
AM-601.3	Analyze the strengths and limitations of basic RNNs, LSTMs, and GRUs, and select the most appropriate gated architecture for a given sequential classification problem, justifying the choice with gradient-flow and performance considerations.	Analyze	Analyze
AM-601.4	Design and implement a Transformer-based encoder-decoder model with multi-head self-attention and positional encodings to solve a sequence-to-sequence task, and evaluate its accuracy and latency against an equivalent RNN baseline.	Design	Create
AM-601.5	Evaluate generative models--including Variational Autoencoders and GAN variants--by training them on image or text datasets, measuring reconstruction quality, diversity, and stability, and recommending the most suitable model for a specified application.	Evaluate	Evaluate
AM-601.6	Synthesize reinforcement-learning concepts with ethical AI principles to propose a responsible RL solution, outlining the MDP	Synthesize	Create

	formulation, policy-learning approach, safety mechanisms, and fairness assessment for a real-world scenario.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	1	-	-	-	-	1	-	1
CO2	2	2	-	1	3	-	-	-	1	1	-	2
CO3	1	3	2	2	2	-	-	-	1	1	-	1
CO4	2	2	3	2	3	-	-	-	1	2	1	1
CO5	1	2	2	3	2	1	1	1	1	2	1	1
CO6	1	2	2	2	2	3	1	3	1	2	1	1

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	2	3	1
CO3	3	2	1
CO4	3	3	1
CO5	3	3	2
CO6	3	3	3

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Software Engineering
Course Code	AM-602
Semester	6
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

• Fundamental programming skills and data structures (e.g., Python/Java, arrays, linked lists, trees)

• Introductory software engineering concepts including the software development lifecycle, version control, and basic project planning

• Basic statistics and linear algebra concepts needed for understanding AI/ML components

2. Course Learning Objectives

• Guide students to develop a holistic understanding of modern software engineering processes, including traditional life-cycle models, Agile/DevOps practices, and emerging AI-enabled system considerations.

• Equip learners with the ability to systematically capture, analyze, specify, and manage requirements--both functional and non-functional--while addressing ethical, data-quality, and bias issues specific to AI/ML components.

• Enable students to design robust, maintainable, and secure software architectures using appropriate styles, patterns, and documentation techniques, and to articulate design decisions through UML and lightweight artifacts.

• Foster competence in planning, executing, and controlling software projects through effective estimation, scheduling, configuration management, and DevOps-driven development practices.

• Develop proficiency in applying comprehensive testing, verification, validation, and quality-assurance strategies--including automated testing and AI-specific evaluation--to ensure reliable, ethical, and high-quality software products.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full marks**	**Assignment Full marks**	**Attendance Full marks**	**Total Marks**
CIA-1	25	10	5	40
CIA-2	25	10	5	40
End Semester Examination (ESE)	_	_	_	60
Total				100

5. Course Modules

Module	**Topics**	**Hours**
1	**Foundations of Software Engineering** - Introduction to software and software engineering - Role of the software engineer, software	5

	characteristics, and the software crisis - Core quality attributes (reliability, usability, security, maintainability, portability) - Classic life-cycle models (Waterfall, Incremental, Evolutionary, Spiral, RAD) and their modern counterparts (Agile, DevOps) - Generic process framework and layered technology view - Process assessment basics (CMMI concepts, KPAs, maturity levels) - high-level overview only - Emerging concerns for AI-enabled systems (ethical considerations, data-driven quality) - Software ethics and professional responsibility	
2	**Requirement Engineering** - Types of requirements: functional, non-functional, user, system, interface - Elicitation techniques: interviews, workshops, brainstorming, user stories, story mapping - Analysis artifacts: use-case models, data-flow diagrams, ER diagrams, simple object models - Specification standards: IEEE 830-1993 SRS, IEEE 1016-1998 design description, lightweight Agile specs - Requirements validation, traceability, and change-management basics - Requirements for AI/ML components (data quality, bias, privacy, explainability) - Tool support (requirements management tools, versioned repositories)	6
3	**Design and Architecture** - Design fundamentals: modularity, cohesion, coupling, design metrics - Design strategies: top-down, bottom-up, hybrid, functional-oriented, object-oriented - Architectural styles & patterns: layered, client-server, micro-services, MVC, event-driven, AI-centric pipelines - Design documentation standards (IEEE 1016-1998) and lightweight alternatives (architecture decision records) - UML modeling basics: class, sequence, collaboration, component, deployment diagrams - Common design patterns (Factory, Observer, Strategy, Repository) and their relevance to AI systems - Security-by-design and privacy-by-design considerations	7
4	**Development Practices & Project Planning**	8

	- Project initiation: scope definition, stakeholder identification, team organization - Scheduling and planning techniques (Gantt, critical path, Agile sprint planning) - Effort and cost estimation basics (COCOMO overview, story-point sizing) - Staffing models and personal development planning - Configuration management: version control, baseline management, branching strategies - CASE tools, integrated development environments, and CI/CD pipelines - Software metrics & measurement (process, product, code quality, test coverage) - Coding standards, code review practices, static analysis, and test-driven development - Introduction to DevOps culture and continuous delivery	
5	**Testing, Verification & Validation** - Testing fundamentals: testability, test process, test strategies - White-box testing: unit testing, integration testing, test drivers & stubs, test coverage - Black-box testing: system testing, acceptance testing, regression testing, exploratory testing - Test automation frameworks and CI-integrated test suites - AI/ML testing concerns: data validation, model performance, bias detection, explainability checks - Verification vs. validation concepts (practical distinction, no heavy formal methods) - Test metrics: defect density, pass/fail rates, coverage percentages, AI model confidence metrics	8
6	**Quality Assurance, Risk Management, Maintenance & Project Control** - Software quality assurance fundamentals and QA planning - Industry standards overview (ISO 9001, ISO 25010, SEI-CMM basics) - Review techniques: peer reviews, walkthroughs, inspections - Risk management lifecycle: identification, analysis, mitigation, monitoring (focus on practical techniques) - Maintenance models: corrective, adaptive, perfective, and AI model retraining cycles - Project monitoring & control: key performance indicators, earned value, dashboards - Process improvement: continuous improvement	8

	loops, retrospectives, metrics-driven refinement - Security and ethical risk considerations for AI-enabled software - Post-deployment monitoring of AI systems (drift detection, performance alerts)	

6. References

Textbooks:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, BPB Publications.

2. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6 edition, Mc Graw Hill International Edition.

3. Software Engineering - A practitioner's approach, R. S. Pressman, 6th ed., McGraw Hill Int. Ed

Reference Books:

1. Software Engineering, Ian Sommerville, Pearson Education; Tenth edition.

2. Roger S Pressman, Software Engineering-A Practitioners Approach, McGraw Hill Publications.

3. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-602.1	Identify and describe the fundamental concepts of software engineering, classic and modern life-cycle models, core quality attributes, and ethical considerations specific to AI-enabled systems.	Identify	Remember
AM-602.2	Explain the various requirement-engineering techniques, artifacts, and standards, and illustrate how to capture functional, non-functional, and AI/ML-specific requirements with traceability.	Explain	Understand
AM-602.3	Apply design principles, architectural styles, and common	Apply	Apply

	design patterns to develop modular, secure, and AI-centric software architectures, producing appropriate UML diagrams and design documentation.		
AM-602.4	Analyze project planning, effort estimation, and risk-management methods for AI/ML projects, selecting suitable scheduling techniques, staffing models, and metrics to support effective project control.	Analyze	Analyze
AM-602.5	Evaluate testing and verification approaches for both conventional and AI/ML components, including test automation, bias detection, and model-performance metrics, and recommend improvements based on quantitative test results.	Evaluate	Evaluate
AM-602.6	Design a comprehensive quality-assurance and maintenance strategy for an AI-enabled software system that integrates continuous improvement, security-by-design, ethical risk mitigation, and post-deployment monitoring for model drift.	Design	Create

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	2	1	3	1	1	1	2
CO2	3	3	2	2	2	1	1	1	2	3	2	2
CO3	3	2	3	2	3	2	1	2	2	3	2	2
CO4	2	3	2	3	2	2	1	1	2	2	3	2
CO5	2	3	2	3	3	2	1	3	2	2	2	2
CO6	2	3	3	3	3	3	2	3	2	3	3	2

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	2
CO2	2	1	1
CO3	3	2	1
CO4	2	2	1

CO5	3	2	2
CO6	3	2	3

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Introduction to Entrepreneurship
Course Code	AM-603
Semester	6
Course Category	Humanities and Social Science
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

- Introductory Business/Management concepts (e.g., basic economics, value creation, and market fundamentals)

- Fundamental digital literacy and familiarity with common productivity and data-analysis tools (e.g., spreadsheets, presentation software, basic internet research)

- Basic understanding of technology trends and AI fundamentals (e.g., what AI is, its capabilities, and ethical considerations)

2. Course Learning Objectives

- Guide students to develop an entrepreneurial mindset that integrates ethical decision-making, AI-driven innovation, and value-creation principles across diverse venture contexts.

- Equip learners with systematic methods for identifying, evaluating, and validating market opportunities using design-thinking, AI analytics, and rigorous feasibility criteria.

- Enable students to construct and communicate comprehensive, lean business plans--including legal, financial, operational, and marketing components--tailored to the needs of early-stage technology ventures.

• Develop proficiency in financial management, funding strategies, and AI-enhanced performance monitoring to support sustainable growth and responsible capital acquisition.

• Foster the ability to design scalable operations, build high-performing teams, and leverage ecosystem resources while applying AI, sustainability, and inclusive entrepreneurship practices.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Foundations of Entrepreneurship** - Definition of an entrepreneur and the entrepreneurial mindset - Core entrepreneurial traits and motivations - Entrepreneur vs. manager vs. intrapreneur - Types and roles of entrepreneurs in the economy - Value creation, value proposition, and problem-solving focus - Overview of entrepreneurship's impact on economic development - Common myths and realities about entrepreneurship	6

	- Brief history and evolution of entrepreneurship - Introduction to Lean Startup & Business Model Canvas concepts - Relevance of AI and digital technologies for modern ventures - Ethical foundations: values, attitudes, and responsible innovation	
2	**Opportunity Identification and Ideation** - Sources of new ideas (personal, market, technology trends, AI-driven insights) - Structured idea-generation techniques (brainstorming, SCAMPER, design thinking) - Creative problem-solving frameworks - Opportunity identification process and screening criteria - Feasibility assessment: market size, customer need, competitive landscape - Using AI tools for market analysis and early validation - Customer discovery and validation basics - Pitfalls and the "dark side" of entrepreneurship (bias, over-optimism, ethical risks)	7
3	**Business Planning, Legal Framework & Ethics** - Components of a business plan (executive summary, problem, solution, market, model, team, finances) - Writing and presenting a concise business plan / Lean Canvas - Project formulation, appraisal, and detailed project report basics - Legal formalities: registration, licensing, intellectual property basics - Government policies & support schemes for SMEs and startups - Business ethics, corporate social responsibility, and AI ethics (bias, transparency, data privacy) - Risk management and compliance considerations for digital ventures	8
4	**Financing & Financial Management of New Ventures** - Sources of capital: bootstrapping, angel investors, venture capital, government schemes, fintech platforms - Fundamentals of financial statements and cash-flow management - Budgeting, cost control, and break-even analysis for early-stage firms - Working-capital management and financing short-	7

	term needs - Financial controls and performance dashboards using AI-enabled tools - Basics of valuation and preparing for investor pitches - Ethical considerations in fundraising and financial reporting	
5	**Operations, Human Resources, Marketing & Technology** - Operations basics: production planning, inventory control, and quality management (TQM) - Lean operations and process automation with AI/IoT - Human-resource fundamentals: recruitment, team motivation, leadership in startups - Marketing fundamentals for startups: segmentation, positioning, pricing, promotion - Digital marketing, e-commerce, and AI-driven customer analytics - Technology adoption roadmap and cybersecurity basics for entrepreneurs - Record-keeping and data-driven decision making	6
6	**Growth, Scaling, Ecosystem & Specialized Entrepreneurship** - Growth strategies: market expansion, product line extension, and scaling operations - Business model innovation and go-to-market planning - Building and leveraging startup ecosystems: incubators, accelerators, E-Cells - Policy environment and support mechanisms for SMEs and tech-driven ventures - Specialized entrepreneurship: women, rural, and social entrepreneurship - AI-focused entrepreneurship: data-driven products, ethical AI, and responsible innovation - Sustainable practices and impact measurement for growing ventures	8

6. References

Textbooks:

1. Poornima M. Charantimath - "Entrepreneurship Development and Small Business Enterprise" (Pearson)

2. Paul Burns - "Entrepreneurship and Small Business" (Palgrave / Bloomsbury imprint)

Reference Books:

1. NCERT / Open Text - "Introduction to Entrepreneurship" (Open Textbook Library / KPU Pressbooks)

2. NCERT / Other Indian HE materials - "Entrepreneurship and Small Business Management"

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-603.1	Recall and describe the fundamental concepts of entrepreneurship, including the entrepreneurial mindset, value creation, and the impact of AI and digital technologies on new ventures.	Describe	Understand
AM-603.2	Apply structured ideation techniques and AI-driven market-analysis tools to generate, screen, and preliminarily validate at least three viable business ideas.	Apply	Apply
AM-603.3	Analyze the elements of a business plan, relevant legal requirements, and ethical considerations--including AI ethics--to produce a critical assessment report of a proposed venture.	Analyze	Analyze
AM-603.4	Evaluate various financing sources and AI-enabled financial-management tools to recommend an optimal capital-structure and budgeting plan for an early-stage startup.	Evaluate	Evaluate
AM-603.5	Design a complete Lean Canvas and prototype an AI-powered product/service, integrating operations, marketing, and sustainability strategies, and present the solution in a pitch deck.	Design	Create
AM-603.6	Formulate a comprehensive growth and scaling strategy that leverages startup-ecosystem resources, AI/ML innovations, and impact-measurement frameworks,	Formulate	Synthesize

	and defend the strategy in a simulated investor panel.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	1	1	2	1	1	1	2	1	2
CO2	2	3	3	3	3	2	1	2	2	2	2	2
CO3	1	3	2	3	2	3	2	3	2	3	2	2
CO4	2	2	2	3	3	2	1	2	2	2	3	2
CO5	3	2	3	3	3	2	3	2	3	3	2	2
CO6	2	3	3	3	3	3	2	2	3	3	3	2

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	2	1	1
CO2	2	3	1
CO3	2	2	3
CO4	2	3	1
CO5	2	3	2
CO6	3	2	2

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Optimization Techniques in Machine Learning
Course Code	AM-604
Semester	6
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

• Linear algebra and multivariate calculus (vectors, matrices, gradients, Hessians, eigen-values)

• Proficiency in Python programming with experience using NumPy (and basic familiarity with PyTorch or similar scientific libraries)

• Fundamental understanding of probability/statistics and basic machine-learning concepts (e.g., supervised learning, loss functions)

2. Course Learning Objectives

• Equip students with a solid theoretical foundation in convex analysis and optimization, enabling them to understand and derive first- and second-order optimality conditions for a wide range of continuous and discrete problems.

• Develop practical proficiency in implementing and selecting appropriate optimization algorithms--from basic gradient methods to advanced quasi-Newton and large-scale deep-learning optimizers--using modern Python tools such as NumPy, PyTorch, and scientific-computing libraries.

• Enable learners to critically evaluate and apply optimization techniques within classical machine-learning models and modern deep-learning architectures, including considerations of regularization, scalability, and computational efficiency.

- Foster the ability to design, execute, and communicate end-to-end optimization projects, emphasizing reproducibility, ethical AI practices, and effective presentation of results to technical and non-technical audiences.

- Prepare students to bridge theory and practice by diagnosing optimization challenges (e.g., ill-conditioning, non-convexity, resource constraints) and employing appropriate strategies such as preconditioning, adaptive methods, and distributed training.

3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

4. Evaluation System

Activities	Class Test Full marks	Assignment Full marks	Attendance Full marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
			Total	100 marks

5. Course Modules

Module	Topics	Hours
1	**Foundations of Optimization & Convex Analysis** Mathematical background: Linear algebra (vectors, matrices, eigenvalues) and multivariate calculus (gradients, Hessians, Taylor expansion), Rate of convergence (Chapter 4.2): linear, sublinear, superlinear convergence, Convex sets and convex functions: definitions, examples, and properties, First- and second-order optimality conditions,	6

	KKT conditions (intuitive introduction) and duality concepts, Common optimization problem types: LP, QP, simple convex programs.	
2	**First-Order Optimization Methods** Gradient Descent: fundamentals, convergence analysis, and step-size selection, Acceleration methods: Polyak momentum, Nesterov acceleration, Projected Gradient Descent for constrained problems, Subgradient and Proximal methods for non-smooth optimization, Frank–Wolfe algorithm for constrained convex problems, Mirror Descent and its interpretation in dual spaces.	8
3	**Second-Order and Practical Optimization Algorithms** - Newton's method: intuition, step-size, when to use it - Quasi-Newton methods: BFGS and limited-memory BFGS (L-BFGS) - Simple line-search and trust-region ideas for robustness - High-level view of interior-point concepts (no deep theory) - Cutting-plane and active-set ideas for small-scale problems	7
4	**Optimization for Classical Machine-Learning Models** - Empirical risk minimization and common loss functions - Linear regression and ridge regression (closed-form & gradient-based) - Logistic regression and Support Vector Machines (primal intuition, dual sketch) - Regularization & sparsity: Lasso and basic feature-selection techniques - Simple discrete optimization in ML (knapsack for feature budgeting, matching) - High-level submodular ideas for data summarization and DPP intuition - Case study overview: model compression & structured prediction	5
5	**Deep Learning and Large-Scale Optimization** Optimization in deep neural networks: backpropagation and loss minimization,	8

	Large-scale SGD: batch-size scaling, warm-up, and gradient clipping, Adaptive optimizers: AdaGrad, RMSProp, Adam, AdamW, Practical convergence challenges: ill-conditioning and preconditioning, Distributed and memory-efficient optimization (overview only)	
6	**Implementation foundation of optimization in machine learning** - NumPy and PyTorch fundamentals for optimization experiments - Implementing core algorithms from scratch (GD, SGD, Adam, L-BFGS) - Project workflow: data preprocessing, model selection, evaluation metrics - Real-world case studies (image classification, recommender systems) - Reproducibility practices: random seeds, version control, experiment tracking - Ethical considerations in AI optimization (bias, resource usage, sustainability)	8

6. References

Textbooks:

1. A. Beck, First-Order Methods in Optimization, MOS-SIAM Series on Optimization, 2017.
2. S. Bubeck, Convex Optimization: Algorithms and Complexity, Foundations and Trends in Optimization, 2015.
3. F. Bach, “Learning with Submodular Functions: A Convex Optimization Perspective”, Foundations and Trends in Machine Learning, Now Publishers Inc.

Reference Books:

1. Roberto Battiti, Mauro Brunato. The LION Way: Machine Learning plus Intelligent Optimization. Lionsolver, Inc. 2013.

2. S. Boyd, N. Parikh, and E. Chu,“ Distributed optimization and statistical learning via the alternating direction method of multipliers”, Foundations and Trends in Machine Learning, Now Publishers Inc.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-604.1	Define key concepts of convex sets, convex functions, and optimality conditions, and recall how linear-algebra and multivariate-calculus tools are used to formulate continuous optimization problems.	Define	Understand
AM-604.2	Implement gradient descent, projected gradient descent, and Adam from scratch in Python to minimize convex loss functions, and demonstrate convergence to within 1 % of the known optimum on benchmark problems.	Implement	Apply
AM-604.3	Compare first-order and second-order optimization algorithms by analysing their convergence rates, computational complexity, and robustness to ill-conditioning, and justify the selection of the most suitable method for a given problem scenario.	Compare	Analyze
AM-604.4	Evaluate the performance of classical machine-learning models (linear regression, ridge regression, logistic regression, SVM, Lasso) trained with appropriate optimization techniques, using metrics such as RMSE, classification accuracy, and sparsity, and identify the model that satisfies predefined performance thresholds.	Evaluate	Evaluate
AM-604.5	Design a large-scale deep-learning training pipeline that incorporates advanced optimizers (e.g., AdamW), learning-rate schedules, mixed-precision arithmetic, and data-parallel distribution, and achieve at least a 2 % improvement in validation accuracy over a baseline on a standard image-classification dataset.	Design	Create
AM-604.6	Synthesize an end-to-end AI optimization project--including data preprocessing, model selection, hyper-parameter tuning,	Synthesize	Create

	reproducibility practices, and ethical impact assessment--and communicate the results through a written report and oral presentation that meet the course grading rubric.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	1	1	-	-	-	-	-	-	-
CO2	2	2	-	2	3	-	-	-	1	1	-	-
CO3	2	3	2	3	2	-	-	-	-	-	-	-
CO4	2	2	2	3	3	-	-	-	1	1	-	-
CO5	2	-	3	2	3	1	-	-	1	1	2	-
CO6	2	2	2	3	2	2	-	3	3	3	2	1

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	2	3	1
CO3	3	2	1
CO4	2	3	1
CO5	2	3	1
CO6	3	3	3

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Blockchain Technology
Course Code	AM-605
Semester	6
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

• Data Structures and Algorithms (including complexity analysis)

• Fundamentals of Computer Networks and Distributed Systems

• Basic Cryptography and Number Theory

2. Course Learning Objectives

• Enable students to comprehend and articulate the fundamental principles of distributed systems and cryptography that underpin modern blockchain technologies.

• Guide learners in analyzing and comparing a variety of consensus mechanisms and blockchain architectures, assessing their security, scalability, and suitability for different application domains.

• Develop students' ability to design, implement, and critically evaluate smart contracts and decentralized applications, emphasizing correctness, security best-practices, and appropriate platform selection.

• Equip learners with the knowledge to identify, assess, and propose blockchain-based solutions for real-world problems across diverse sectors such as finance, supply chain, public services, and healthcare.

• Foster an appreciation of emerging trends, research challenges, and the broader ethical, regulatory, and societal implications of decentralized ledger technologies.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full marks**	**Assignment Full marks**	**Attendance Full marks**	**Total Marks**
CIA-1	25	10	5	40
CIA-2	25	10	5	40
End Semester Examination (ESE)	_	_	_	60
Total				100

5. Course Modules

Module	Topics	Hours
1	**Foundations of Distributed Systems & Cryptography** - Why distributed record-keeping matters for	8

	modern applications - Core distributed-system concepts: nodes, messages, consistency, CAP theorem - Fault models and adversary assumptions (crash, Byzantine) - Atomic broadcast and the Byzantine Generals problem - Public-key cryptography fundamentals (key pairs, encryption, decryption) - Secure hash functions, collision resistance and hash pointers - Digital signatures and basic signature schemes - Simple symmetric encryption and the concept of a random oracle - The double-spend problem and its relevance to decentralized ledgers - Core consensus properties: safety, liveness, and fairness	
2	**Blockchain Architecture & Bitcoin Core** - Blockchain as a layered protocol stack (network, consensus, data, application) - Block structure, hash chaining and Merkle trees - Bitcoin's UTXO model and transaction lifecycle - Wallets, addresses and basic key-management practices - Mining basics and Proof-of-Work as a probabilistic puzzle - Nakamoto consensus on permission-less networks - Incentive mechanisms: block reward and transaction fees - Forks, hard vs. soft forks and how network upgrades happen - Formal notions of consistency, liveness and fairness in PoW systems	7
3	**Consensus Algorithms & Advanced Models** - Recap of Proof-of-Work and its scalability challenges - Proof-of-Stake fundamentals and common variants (DPoS, PoET, deposit-based) - Hybrid consensus designs (e.g., PoW + PoS) - Practical Byzantine Fault Tolerance (PBFT) and federated Byzantine consensus - Permissioned vs. permissionless consensus design goals - Scalability techniques: sharding basics and Algorand-style cryptographic sortition - Common security threats: 51 % attacks, selfish mining, Sybil attacks	7
4	**Smart Contracts, Ethereum & Permissioned**	6

	Platforms - What smart contracts are and why correctness matters - Ethereum architecture: accounts, EVM, gas model - Solidity language basics and simple contract patterns - Development workflow with Truffle (compile, test, deploy) - Testing and lightweight formal verification of contracts - Typical smart-contract attacks and mitigation strategies - Bitcoin script vs. Ethereum's Turing-complete contracts - Introduction to permissioned blockchains: Hyperledger Fabric components - Chaincode design principles for Fabric	
5	**Blockchain Applications & Use-Cases** - Supply-chain provenance and traceability - Financial services: settlement, KYC, tokenised assets, invoice financing - Government & public sector: digital identity, land-registry, welfare distribution - Health-record management and self-sovereign identity - Privacy-preserving solutions: overview of Zcash, basic zero-knowledge proofs - Decentralised AI data markets and model provenance - Beyond cryptocurrency: cybersecurity, tamper-evident logging, contract enforcement	7
6	**Emerging Trends, Limitations & Research Directions** - Evolution from Blockchain 1.0 -> 2.0 -> 3.0 - Technical limitations and common myths (throughput, energy, decentralisation) - Introductory zero-knowledge proof systems (SNARKs, zk-STARKs) and blockchain integration - Privacy vs. pseudo-anonymity debate - Emerging consensus research (Algorand, sharding, hybrid models) in plain language - Interoperability standards (Polkadot, Cosmos) and cross-chain communication - Open research problems: scalability, governance, AI-driven smart contracts - Ethical, regulatory and societal implications of decentralized ledgers	7

6. References

Textbooks:

1. Mark Gates, "Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money", Wise Fox Publishing and Mark Gates 2017

2. Kirankalyan Kulkarni, Essentials of Bitcoin and Blockchain, Packt Publishing.

3. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.

4. Draft version of "S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, 'Blockchain Technology: Cryptocurrency and Applications', Oxford University Press, 2019.

Reference Books:

1. Josh Thompson, 'Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming', Create Space Independent Publishing Platform, 2017.

2. Blockchain: Blueprint for a New Economy by Melanie Swan, Shroff Publisher O'Reilly Publisher Media; 1st edition (2015).

3. Andreas Antonopoulos, "Mastering Bitcoin: Unlocking Digital Crypto currencies", O'Reilly Media, Inc. 2014.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-605.1	List and define the core concepts of distributed systems and cryptographic primitives, including nodes, messages, the CAP theorem, fault models, hash functions, and digital signatures.	List	Remember
AM-605.2	Explain the layered architecture of blockchain systems and describe Bitcoin's UTXO model, block structure, and Proof-of-Work consensus mechanism.	Explain	Understand
AM-605.3	Apply selected consensus algorithms (e.g., PoW, PoS, PBFT, sharding) to evaluate their safety, liveness, and scalability properties for given application scenarios.	Apply	Apply

AM-605.4	Analyze common smart-contract vulnerabilities and conduct unit testing and lightweight formal verification of Solidity contracts using Truffle and related tools.	Analyze	Analyze
AM-605.5	Evaluate the suitability of blockchain solutions for AI/ML use-cases such as data provenance, decentralized model marketplaces, and privacy-preserving computation, comparing approaches like Zcash-style zero-knowledge proofs.	Evaluate	Evaluate
AM-605.6	Design and prototype a decentralized AI service that integrates on-chain provenance, incentive mechanisms, and appropriate consensus and privacy techniques, and demonstrate its functionality through a working implementation.	Design	Create

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	1	1	1	1	2
CO2	3	2	2	1	2	1	1	1	1	2	1	2
CO3	2	2	3	2	3	1	1	1	2	2	2	2
CO4	2	2	2	3	3	2	1	2	2	2	2	2
CO5	2	3	2	2	2	2	2	2	1	2	2	3
CO6	2	2	3	2	3	2	2	2	3	3	3	3

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	2	1	1
CO2	2	1	1
CO3	2	1	1
CO4	2	1	2
CO5	3	2	2
CO6	3	3	3

DR. B.C. ROY SOCIETY
ENGINEERING
AN AUTONOMOUS INSTITUTE

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	AR-VR
Course Code	AM-606
Semester	6
Course Category	Open Elective Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

- Fundamental programming skills (e.g., C#/C++ or Python) and object-oriented concepts

- Linear algebra and geometry basics (vectors, matrices, coordinate transformations)

- Introductory computer graphics / real-time rendering concepts and basic human-computer interaction

2. Course Learning Objectives

- Provide students with a comprehensive understanding of the theoretical foundations, historical evolution, and interdisciplinary contexts of immersive technologies (VR, AR, and XR).

- Equip learners with the knowledge of human perceptual mechanisms, hardware architectures, and system constraints that influence the design of comfortable, ethical, and effective immersive experiences.

- Develop students' ability to apply core graphics, geometric modeling, and interaction design principles to create performant, user-centered XR applications across multiple platforms.

- Enable learners to integrate modern development toolkits, computer-vision techniques, and AI-driven content generation into end-to-end XR project workflows, from prototyping to deployment.

- Foster critical thinking about the societal, ethical, and accessibility implications of immersive media, preparing students to evaluate and improve XR solutions through rigorous usability testing and user-experience research.

3. Teaching Methodology

- Lectures and Presentations

- Interactive Discussions and Case Studies

- Lab Sessions

- Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Foundations of Immersive Technologies** - Definition and Overview of Virtual Reality (VR) - Definition and Overview of Augmented Reality (AR) - Historical Development and Key Milestones of VR/AR - Taxonomy of Immersive Systems and Types (immersive, semi-immersive, non-immersive) - Comparative Analysis: AR vs. VR - Core Benefits and Limitations of Immersive Media - Major Application Domains (Education, Training, Gaming, Simulation, Mobile/Wearable)	7

	- Emerging AI-driven Content Generation and Adaptive Experiences - Ethical, Privacy, and Societal Considerations in XR	
2	**Human Perception & Sensory Foundations** - Anatomy of the Human Visual System and Eye Movements - Light, Optics, and Color Theory for Display Design - Depth Cues and Perceptual Mechanisms for Spatial Awareness - Auditory Perception and Fundamentals of Sound for Immersive Audio - Basic Haptic (Somatic) Perception and Tactile Feedback - Motion Sickness, Comfort, and Fatigue Mitigation Strategies - Cross-modal Perception: Integrating Visual, Auditory, and Haptic Cues	8
3	**Computer Graphics & Geometric Modeling for VR/AR** - Fundamentals of Computer Graphics for Real-Time XR - Virtual World Space, Coordinate Systems, and Observer Positioning - Perspective Projection and Viewing Transformations - Hierarchical Transformations and Chaining (position & orientation) - Geometric Modeling Techniques and Asset Creation for XR - Introduction to Physical Simulation (collision, basic dynamics) - Performance-Oriented Graphics: LOD, Culling, and Basic Shader Concepts - Conceptual Overview of Quaternions for Smooth Rotation (no heavy math)	7
4	**Hardware and System Architecture** - Overview of VR and AR System Architectures - Sensor Hardware: Tracking, Inertial, Depth, and Controllers - Head-Mounted Displays and Optical Design (visual and AR see-through) - Acoustic Hardware and Spatial Audio Rendering - Input/Output Devices: Visual, Aural, and Haptic Interfaces - Computing Platforms: Standalone, PC-Based, Mobile, Cloud/Edge Streaming	6

	- System Requirements: Latency, Bandwidth, Fidelity, Power & Thermal Constraints	
5	**Interaction Design & Human Factors** - Human Factors in Immersive Environments (vision, hearing, haptics) - Core Interaction Techniques: Selection, Manipulation, and Navigation - Locomotion Methods and Motion Sickness Mitigation - 3D User Interface Design Principles for XR - Social Interaction Models and Multi-User Collaboration - Audio Rendering and Auditory Interaction Strategies - AI-Driven Adaptive Interfaces and Accessibility Considerations - Usability Evaluation Methods for XR (heuristics, user testing)	7
6	**Development Platforms, Toolkits, and Applications** - Overview of XR Development Platforms (Unity, Unreal) and Workflow - VR Toolkits and Scene Description (modern alternatives to VRML) - AR SDKs: ARCore, ARKit, Vuforia and Cross-Platform Solutions - Computer Vision Basics for AR: Marker-Based and Markerless Tracking - SLAM Fundamentals and Real-Time Pose Estimation - Visualization Techniques and UI Integration for XR - Incorporating AI: Generative Content, ML Agents, and Real-Time Analytics - Case Studies: Education, Training, Gaming, Simulation, Mobile/Wearable - Project Planning, Testing, Deployment, and Future Trends in XR	7

6. References

Textbooks:

1. Paul Mealy, John Wiley & Sons - "Virtual & Augmented Reality For Dummies

2. Chiara Gancia et al. (ed.) - "Augmented Reality and Virtual Reality: Empowering Human, Place and Business" (Springer)

Reference Books:

1. John Wiley & Sons - "Virtual Reality and Augmented Reality: Myths and Realities

2. Virtual and Augmented Reality: Concepts, Methodologies, Tools, and Applications

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-606.1	Describe the fundamental concepts, historical milestones, taxonomy, benefits, limitations, and ethical considerations of virtual reality (VR) and augmented reality (AR) technologies.	Describe	Remember
AM-606.2	Explain how human visual, auditory, and haptic perception principles influence design decisions to mitigate motion sickness and fatigue in immersive environments.	Explain	Understand
AM-606.3	Apply real-time graphics techniques--including coordinate systems, perspective projection, hierarchical transformations, LOD, culling, and basic shader concepts--to create optimized 3D assets suitable for XR applications.	Apply	Apply
AM-606.4	Analyze the specifications of various XR hardware components and system architectures (sensors, HMDs, computing platforms) to recommend the most appropriate configuration for a given application scenario.	Analyze	Analyze
AM-606.5	Evaluate interaction-design strategies, including AI-driven adaptive interfaces and accessibility features, using usability heuristics and ethical guidelines to determine their effectiveness in multi-user XR experiences.	Evaluate	Evaluate

AM-606.6	Create a functional XR prototype using a modern development platform (Unity or Unreal) that integrates AI/ML components for generative content, real-time analytics, and adaptive user experiences, and demonstrate its performance on selected hardware.	Create	Create

8. CO-PO Mapping

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	1	1	1	1	2	1	2	1	2	1	2
CO2	2	2	3	2	1	1	1	1	1	1	1	1
CO3	3	2	3	2	3	1	1	1	2	2	2	2
CO4	2	3	3	3	2	2	2	1	2	2	2	2
CO5	2	2	3	3	3	3	2	3	2	3	2	2
CO6	3	2	3	3	3	2	2	2	3	3	3	3

9. CO-PSO Mapping

CO	**PSO1**	**PSO2**	**PSO3**
CO1	1	1	2
CO2	1	1	1
CO3	1	1	1
CO4	1	1	1
CO5	3	3	3
CO6	3	3	2

DR. B.C. ROY SOCIETY
ENGINEERING
AN AUTONOMOUS INSTITUTE

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Cognitive Computing
Course Code	AM-607
Semester	6
Course Category	Open Elective Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

• Proficiency in a programming language (e.g., Python) and basic software engineering concepts

• Fundamental understanding of data structures, algorithms, and discrete mathematics (including logic and set theory)

• Introductory knowledge of artificial intelligence/machine learning concepts and probability/statistics

2. Course Learning Objectives

• Enable students to articulate the fundamental concepts, historical evolution, and societal implications of cognitive computing, distinguishing it from traditional computing and AI paradigms.

• Guide students in designing and evaluating modular, scalable cognitive system architectures, applying appropriate design patterns and integration strategies with AI services.

• Develop students' ability to model, represent, and reason over knowledge using logical formalisms, ontologies, and knowledge graphs, and to select suitable reasoning techniques for given problems.

• Equip students with a conceptual toolkit for incorporating learning paradigms--supervised, unsupervised, reinforcement, and incremental--into cognitive pipelines, emphasizing data preparation, model evaluation, and explainability.

• Prepare students to prototype, assess, and responsibly deploy end-to-end cognitive applications (e.g., agents, decision-support tools, multimodal interfaces) using contemporary platforms and to address ethical, bias, and privacy considerations throughout the development lifecycle.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Foundations of Cognitive Computing** - Overview & Introduction to Cognitive Computing - Cognitive Computing vs. Traditional Computing - Cognitive Computing vs. AI - Core Characteristics & Design Principles - Cognitive Psychology Foundations (perception, memory, reasoning) - Reference Architecture (input-processing-output loop)	6

	- Historical Milestones & Emerging Trends - Ethical, Social & Privacy Considerations - Representative Applications & Use-Cases	
2	**Cognitive Architectures and System Design** - Overview of Cognitive Computing Architectures - Key Architectural Styles (layered, service-oriented, hybrid) - Canonical Cognitive Architectures (e.g., SOAR, ACT-R, LIDA) - concepts only - System Design Process: requirements -> modeling -> prototyping -> evaluation - Modularity, Scalability & Integration with AI Services - Design Patterns for Knowledge-Intensive Systems - Evaluation Criteria (performance, adaptability, explainability)	7
3	**Knowledge Representation & Reasoning** - Fundamentals of Knowledge Representation in Cognitive Systems - Logical Formulations (propositional & first-order basics) - Semantic Networks & Conceptual Graphs - Ontologies & Knowledge Graphs (construction & maintenance) - Rule-Based and Frame-Based Representations - Knowledge Acquisition & Engineering Techniques - Reasoning Methods: deduction, forward/backward chaining, constraint satisfaction - Querying with SPARQL / Simple Retrieval - Symbolic vs. Sub-symbolic Overview (high-level only, no heavy math)	8
4	**Learning Paradigms for Cognitive Systems** - Learning in Cognitive Systems: overview - Supervised Learning - concepts & typical workflows - Unsupervised Learning - clustering & pattern discovery - Reinforcement Learning - agents & reward loops (conceptual) - Incremental & Online Learning for continual adaptation - Feature Engineering & Data Pre-processing for cognitive pipelines - Model Evaluation & Validation (accuracy, robustness, fairness) - Personalization & Adaptive User Modeling	7

	- Brief intro to Transfer Learning & Explainability (non-mathematical)	
5	**Perception, Interaction & Human-Computer Interfaces** - Perception in Cognitive Systems: role & challenges - Basics of Computer Vision for cognition (image preprocessing, object detection concepts) - Speech Processing Foundations (recognition, synthesis basics) - Multimodal Interaction (fusion of vision, speech, touch) - Natural Language Understanding & Processing (intent detection, entity extraction) - Dialog System Architecture & Conversational Flow - Action & Motor Primitives for embodied cognition - Human-Computer Interaction Principles for cognitive applications - Usability Testing, Accessibility, and UX Evaluation	7
6	**Agents, Robotics, Decision Support, Tools & Applications** - Cognitive Agents & Multi-Agent System Concepts - Reinforcement-Based Learning for Agent Behaviour (high-level) - Introductory Robotics for Cognitive Systems (sensing, actuation, control loops) - Decision Support & Planning Foundations (rule-based, heuristic approaches) - Intelligent Assistants & Analytics Applications - Domain-Specific Case Studies (Healthcare, Finance, Education) - Overview of Major Cognitive Computing Platforms & Tools (e.g., IBM Watson, Azure Cognitive Services, Google Vertex AI) - APIs, SDKs, and Service Integration Patterns - Project/Practicum Workflow: problem definition -> prototyping -> evaluation -> deployment - Ethical Deployment, Bias Mitigation, and Responsible AI Practices	7

6. References

Textbooks:

1. Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles - "Cognitive Computing and Big Data Analytics" (Wiley)

2. Kumar Raghavan (ed.) - "Cognitive Computing: Theory and Applications, Handbook of Statistics, Volume 35" (Elsevier)

Reference Books:

1. Cognitive Computing Systems: Applications and Technological Advancements

2. Advanced Cognitive Computing: Tools and Techniques for Smart Application Development

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-607.1	Define the core concepts, characteristics, and ethical considerations of cognitive computing and distinguish it from traditional computing and artificial intelligence.	Define	Remember
AM-607.2	Explain the main cognitive-computing architectural styles (layered, service-oriented, hybrid) and the principles of canonical architectures such as SOAR, ACT-R, and LIDA, highlighting their modularity, scalability, and integration with AI services.	Explain	Understand
AM-607.3	Construct a simple ontology or knowledge graph for a chosen domain, represent knowledge using semantic networks or frames, and formulate and execute SPARQL queries to retrieve information.	Construct	Apply
AM-607.4	Compare supervised, unsupervised, and reinforcement learning paradigms for cognitive systems, and design an incremental learning pipeline that selects appropriate techniques, performs feature engineering, and evaluates model performance using accuracy, robustness, and fairness metrics.	Compare	Analyze

AM-607.5	Assess a multimodal conversational interface by integrating vision and speech perception modules, conducting usability testing with target users, and evaluating the system's accessibility, user experience, and compliance with HCI principles.	Assess	Evaluate
AM-607.6	Develop a complete cognitive application (e.g., a decision-support agent) using a cloud cognitive platform such as IBM Watson, Azure Cognitive Services, or Google Vertex AI, demonstrating end-to-end prototyping, deployment, explainability, bias mitigation, and ethical compliance.	Develop	Create

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	1	1	2	-	3	-	1	-	1
CO2	3	2	3	2	2	1	-	1	-	2	1	1
CO3	3	2	3	2	3	1	1	1	1	2	1	2
CO4	3	3	3	3	3	2	1	3	1	2	2	2
CO5	3	2	3	3	3	3	1	2	2	2	2	1
CO6	3	2	3	2	3	3	1	3	2	2	3	3

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	3
CO2	3	2	1
CO3	2	3	1
CO4	3	2	2
CO5	2	2	3
CO6	3	3	3