



Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	AM -501
Course Code	Deep Learning -I
Semester	5
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

- Linear Algebra
- Calculus (including derivatives and gradients)
- Programming Proficiency (Python preferred)

2. Course Learning Objectives

- To provide students with a foundational understanding of artificial intelligence, focusing on the principles and applications of neural networks and deep learning.
- To equip students with the practical skills necessary to build, train, and evaluate various types of neural networks, including feed-forward networks and convolutional neural networks, using TensorFlow/Keras.
- To foster a comprehensive understanding of the mathematical and computational underpinnings of deep learning, including optimization algorithms, regularization techniques, and loss functions, without requiring extensive mathematical proofs.

- To cultivate critical thinking skills related to the ethical implications of AI and machine learning, emphasizing bias, fairness, and accountability in the development and deployment of AI systems.
- To enable students to apply their knowledge through the completion of a substantial deep learning project, demonstrating proficiency in data preprocessing, model selection, training, evaluation, and hyperparameter tuning.

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	Introduction to Artificial Intelligence and Neural Networks - Introduction to AI and its subfields (focus on ML & Deep Learning) - Brief history of AI and Deep Learning (key milestones) - Biological inspiration for neural networks (simplified overview) - Introduction to Neural Networks: basic concepts	10

	(neurons, layers, weights, biases) - Representing data for neural networks	
2	Fundamentals of Feed-forward Neural Networks - Perceptron and threshold units - Multilayer perceptrons (MLPs): architecture and functionality - Activation functions (sigmoid, ReLU, softmax) - Loss functions (MSE, cross-entropy, binary cross-entropy) - Backpropagation algorithm: conceptual understanding and intuition (detailed derivation omitted)	7
3	Optimization and Regularization Techniques - Gradient Descent (batch, stochastic, mini-batch) - Optimization algorithms (SGD with Momentum, Adam - conceptual understanding) - Regularization techniques (L2 regularization, Dropout) - Overfitting and Underfitting: prevention strategies - Practical aspects of hyperparameter tuning	6
4	Convolutional Neural Networks (CNNs) - Convolution operations: intuitive explanation and examples - Architectural elements of CNNs (convolutional layers, pooling layers, fully connected layers) - Classic CNN architectures (LeNet, AlexNet - high-level overview) - Understanding receptive fields (intuitive explanation) - Applications of CNNs in image classification and object detection	6
5	Practical Deep Learning with TensorFlow/Keras - Introduction to TensorFlow/Keras: building and training simple MLPs and CNNs - Data preprocessing (normalization, standardization) - Data augmentation techniques (image rotation, flipping) - Model evaluation metrics (accuracy, precision, recall, F1-score, AUC) - Using pre-trained models	6
6	Deep Learning Projects and Ethical Considerations - Implementing a chosen project (e.g., image classification, sentiment analysis)	7

	<ul style="list-style-type: none"> - Model deployment considerations (simplified overview) - Hyperparameter tuning strategies (grid search, random search) - Introduction to transfer learning (conceptual understanding) - Ethical considerations in AI and ML (bias, fairness, accountability) 	
--	--	--

6. References

Textbooks:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
2. Chollet, F. (2017). Deep Learning with Python. Manning Publications.

Reference Books:

1. Trask, A. (2019). Grokking Deep Learning. Manning Publications
2. Aggarwal, C. C. (2018). Neural Networks and Deep Learning: A Textbook. Springer.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
Deep Learning - I.1	Define key concepts in artificial intelligence, including machine learning and deep learning, and describe the biological inspiration behind neural networks.	Define, Describe	Remember, Understand
Deep Learning - I.2	Explain the architecture and functionality of feed-forward neural networks, including different activation and loss functions, and apply the backpropagation algorithm conceptually to understand model training.	Explain, Apply	Understand, Apply
Deep Learning - I.3	Apply various optimization and regularization techniques, such as gradient descent and L2 regularization, to improve the performance of neural networks and prevent overfitting.	Apply	Apply

Deep Learning - I.4	Analyze the architecture and functionality of Convolutional Neural Networks (CNNs), including convolutional and pooling layers, and explain their applications in image processing tasks.	Analyze, Explain	Analyze, Understand
Deep Learning - I.5	Design, implement, and evaluate a deep learning model using TensorFlow/Keras, incorporating data preprocessing, augmentation, and appropriate evaluation metrics for a chosen application (e.g., image classification, sentiment analysis).	Design, Implement, Evaluate	Apply, Create
Deep Learning - I.6	Evaluate and discuss the ethical implications of AI and ML applications, considering issues of bias, fairness, and accountability in the context of a chosen project.	Evaluate, Discuss	Evaluate

8. CO-PO Mapping

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

9. CO-PSO Mapping

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



Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Computer Networks
Course Code	AM-502
Semester	5
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

- Fundamental understanding of computer architecture and operating systems
- Basic programming proficiency (e.g., Python or Java) for socket and network-related coding
- Introductory knowledge of discrete mathematics and binary number systems

2. Course Learning Objectives

- This course introduces students to fundamental concepts and applications of the subject
- Students will learn theoretical foundations and practical skills relevant to the field

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 Networking <ul style="list-style-type: none"> - Introduction to computer networks and the Internet - Network classifications: LAN, MAN, PAN, WAN - Physical media: guided (twisted-pair, coaxial, fiber-optic) and unguided (radio, microwave, infrared) - Network topologies: bus, star, ring, mesh, hybrid - Switching concepts: circuit switching vs. packet switching - OSI 7-layer model and TCP/IP 4-layer model - Key performance metrics: throughput, latency, jitter, bandwidth, delay - Multiplexing techniques (frequency, time, wavelength) - Standards bodies (IEEE, IETF) and the evolution of networking standards 	8
2	Data-Link Layer & LAN Technologies <ul style="list-style-type: none"> - Functions and services of the Data-Link layer - Frame structure and Ethernet II framing - Framing techniques and byte-stuffing - Error detection: parity, checksum, CRC - Simple error correction: Hamming code (conceptual overview) - Flow control and ARQ: Stop-and-Wait, Sliding-Window - MAC access methods: ALOHA, CSMA/CD (wired Ethernet), CSMA/CA (Wi-Fi) - Ethernet standards (10/100/1000 Mbps, full-duplex) and 802.11 WLAN 	6

	basics - Switches, VLANs (802.1Q tagging) and basic switched-LAN design - Data-Link protocols: HDLC, PPP (overview)	
3	Network Layer - Addressing & Routing - IPv4 addressing: binary notation, classful vs. classless, subnet mask, FLSM, VLSM, CIDR, supernetting - Practical subnetting exercises - IPv6 address format, types, and transition mechanisms (dual-stack, tunneling) - ARP operation and ICMP basics (echo, destination-unreachable) - NAT concepts and common use-cases - Routing fundamentals: static, default, and dynamic routing - Distance-vector routing: RIP v2, basic EIGRP concepts - Link-state routing: OSPF fundamentals and area design - Brief overview of BGP as an inter-domain protocol (high-level concepts only) - Multicast addressing basics - Introductory view of SDN control-plane (centralized vs. per-router) for future networking	8
4	Transport Layer - Services & Reliability - Transport-layer role and relationship to the Network layer - Port numbers, socket addressing, and the client/server model - UDP: connectionless service, use-cases (real-time, AI streaming) - TCP: three-way handshake, connection management - TCP flow control (sliding-window, receiver window) - TCP congestion control basics: slow start, AIMD, fast retransmit, fast recovery (CUBIC overview) - Reliable Data Transfer protocols: Stop-and-Wait ARQ, Go-Back-N, Selective Repeat - Introduction to Quality of Service (QoS) concepts: classification, policing, shaping	7
5	Application Layer & Network Programming - Application-layer architecture and service models (client-server, peer-to-peer) - Web protocols: HTTP/1.1, HTTPS (TLS basics), RESTful APIs, JSON - Real-time transport: RTP/RTCP (relevant for AI video/audio streams) - File transfer and remote access: FTP, SFTP, Telnet (legacy) and SSH - Email protocols: SMTP, POP3, IMAP - DNS: hierarchical naming, query types, caching - Socket programming fundamentals (TCP and UDP client/server examples in Python/Java) - Authentication and encryption basics for applications (TLS, JWT, OAuth2) - gRPC and protobuf for high-performance AI service communication	7
6	Advanced Networking & Security - Wireless & mobile networking essentials: 802.11 standards, roaming,	6

	basic mobility management - WAN technologies overview: Frame Relay, ATM, SONET/SDH (conceptual) - Core network security principles: confidentiality, integrity, authentication, non-repudiation - Network-layer security: IPsec (AH/ESP), VPN types (site-to-site, remote-access) - Data-centre networking basics: leaf-spine topology, high-speed Ethernet, load balancing - Software-Defined Networking (SDN) fundamentals and network virtualization (VXLAN, NVGRE) - MPLS overview (label switching, traffic engineering - high-level) - Network management & monitoring: SNMP, NetFlow, basic troubleshooting workflow - Edge computing and AI workload distribution (inference at the edge, federated learning networking considerations) - Zero-Trust networking concepts and best-practice security controls	
--	---	--

6. References

Textbooks:

1. Behrouz A. Forouzan, Data Communications and Networking with TCP/IP Protocol Suite, Sixth Edition TMH, 2022.
2. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach", 2022, 8th Edition (Paperback), Pearson, United Kingdom.

Reference Books:

1. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, Fifth Edition, Morgan Kaufmann Publishers Inc., 2012.
2. William Stallings, Data and Computer Communications, Tenth Edition, Pearson Education, 2013.
3. Nader F. Mir, Computer and Communication Networks, Second Edition, Prentice Hall, 2014.
4. Andrew S. Tanenbaum, "Computer Networks", 2013, 6th Edition, Pearson, Singapore.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
----	-----------	-------------	-----------------

AM-502.1	Identify and describe the fundamental components of computer networks, including classifications (LAN, MAN, WAN, PAN), physical media, topologies, switching concepts, and the OSI/TCP-IP reference models.	Identify	Remember
AM-502.2	Explain the functions of the Data-Link layer, covering frame structures, error-detection methods (parity, checksum, CRC), flow-control techniques, and MAC access protocols (ALOHA, CSMA/CD, CSMA/CA) with illustrative examples.	Explain	Understand
AM-502.3	Calculate IPv4 and IPv6 subnets, design addressing schemes using VLSM/CIDR, and configure both static routes and dynamic routing protocols (RIP v2, OSPF) for given network topologies.	Calculate	Apply
AM-502.4	Compare TCP and UDP transport services, analyze TCP congestion-control algorithms (slow start, AIMD, fast retransmit, fast recovery) and assess their suitability for AI/ML real-time streaming applications.	Compare	Analyze
AM-502.5	Evaluate the security and performance implications of deploying AI/ML workloads on edge and data-center networks, including the use of IPsec/VPNs, SDN, VXLAN, MPLS, and Zero-Trust controls.	Evaluate	Evaluate
AM-502.6	Design and implement a secure, high-throughput networked application for AI model inference using Python socket programming, RESTful APIs, and gRPC over TLS, and demonstrate its operation in a simulated edge-computing environment.	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
CO2	3	2	1	2	2	-	-	-	-	2	-	1
CO3	3	2	3	2	3	-	-	-	1	2	2	1
CO4	3	2	2	2	2	1	-	-	-	2	-	1
CO5	3	3	2	2	2	2	1	2	1	2	2	1
CO6	3	2	3	2	3	2	1	2	2	2	2	2

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	2	1	-
CO2	2	1	1
CO3	2	2	1

CO4	3	2	1
CO5	3	2	2
CO6	3	3	2

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Database Management Systems
Course Code	AM-503
Semester	5
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

- Fundamentals of Computer Science (including data structures, algorithms, and basic programming)
- Introductory course in relational databases and SQL (basic DDL/DML operations)
- Discrete mathematics fundamentals (sets, relations, functions, and basic logic)

2. Course Learning Objectives

- This course introduces students to fundamental concepts and applications of the subject
- Students will learn theoretical foundations and practical skills relevant to the field

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	<p>Foundations of Database Systems</p> <ul style="list-style-type: none"> - Introduction to Database Systems Objectives, motivation, evolution, advantages Typical application domains (e-commerce, AI data stores, analytics) - Database System Architecture Storage manager, query processor, transaction manager, recovery manager Three-level schema architecture (internal, conceptual, external) - Data Abstraction & Independence External, conceptual, internal levels Instances vs. schemas, logical vs. physical data independence - Users, Administrators & Roles Types of users, DBA responsibilities, role-based security Data-governance basics (privacy, audit trails) - Database Languages Overview DDL, DML, DCL, TCL fundamentals Query-by-example (QBE) basics 	5
2	<p>Data Modeling and Design Fundamentals</p> <ul style="list-style-type: none"> - Data Modeling Overview Why modeling matters, building blocks, business rules, abstraction levels - Survey of Data Models Relational, network, hierarchical, object-oriented, NoSQL (key-value, document, column, graph) - high-level comparison - Entity-Relationship (ER) & Enhanced ER (EER) Modeling Entities, attributes, weak entities, relationships, cardinalities, constraints ER diagram creation, common pitfalls, ER-to- 	7

	<ul style="list-style-type: none"> relational mapping rules - Introduction to UML for Database Design Class diagrams as logical models, mapping to relational schema - Conceptual Design Process Requirements capture, high-level ER design (example: university information system) - Functional Dependencies & Normalization Basics Definition, Armstrong's axioms, closure, inference 1NF, 2NF, 3NF, BCNF, lossless-join & dependency-preservation criteria - Schema Evolution & Versioning (brief) Managing changes without disrupting applications 	
3	<p>Relational Theory and Query Foundations</p> <ul style="list-style-type: none"> - Relational Model Fundamentals Relations, tuples, attributes, domains, keys (primary, candidate, foreign) Integrity rules (entity, referential) - Relational Algebra Selection, projection, union, intersection, difference, rename Joins (inner, outer, natural, theta), division, grouping/ungrouping - Relational Calculus Tuple relational calculus, domain relational calculus, equivalence with algebra - Basic SQL DDL (CREATE, ALTER, DROP) and DML (SELECT, INSERT, UPDATE, DELETE) Simple queries, set operators (UNION, INTERSECT, EXCEPT), aggregate functions (COUNT, SUM, AVG, MIN, MAX) GROUP BY, HAVING, basic joins, simple nested subqueries - SQL for AI-Ready Data Storing and retrieving semi-structured data (JSON, ARRAY) - syntax overview 	8
4	<p>Advanced SQL, Constraints, Views & Stored Objects</p> <ul style="list-style-type: none"> - Advanced SQL Features Correlated subqueries, common table expressions (CTE), window functions (ROW_NUMBER, RANK) - practical use cases - Integrity Constraints in SQL PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT Declarative enforcement of business rules - Views Creation, updatable views, materialized views, security considerations, view-update anomalies - Triggers & Stored Procedures/Functions Definition, typical use-cases (audit, data 	7

	<p>validation, ETL hooks), execution flow, examples in PostgreSQL/MySQL</p> <ul style="list-style-type: none"> - Transaction Control Statements COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION - Security & Role Management Users, roles, privileges, revocation, best-practice checklist - Comparison: Tables vs. Views When to choose each, performance impact 	
5	<p>Storage Structures, Indexing, Query Processing & Optimization</p> <ul style="list-style-type: none"> - Storage Structures & File Organization Record layout, heap files, sorted files, simple hash files - Indexing Techniques Primary & secondary indexes, clustered indexes, B-tree & B+-tree, hash indexes, bitmap indexes (read-only analytics) - Query Processing Fundamentals Operator evaluation order, selection algorithms, join algorithms (nested-loop, sort-merge, hash join), basic external sorting - Cost Estimation (conceptual) Estimating I/O cost for selection, projection, join, and aggregate operations - qualitative approach - Query Optimization Strategies Heuristic rewrite rules, basic cost-based optimization, equivalence rules, System-R style optimizer concepts, view materialization decisions - Physical Design for AI Workloads Choosing file organization & indexes for large-scale feature tables, time-series data, and read-heavy analytics 	8
6	<p>Transaction Management, Recovery & Emerging Database Technologies</p> <ul style="list-style-type: none"> - Transaction Management Transaction states, ACID properties, atomicity, durability, isolation, recoverability - Concurrency Control Serializability (conflict & view), lock-based protocols (2PL, multiple-granularity), deadlock detection/avoidance, timestamp ordering, optimistic concurrency, multiversion concurrency control, isolation levels (READ COMMITTED, REPEATABLE READ, SNAPSHOT) - Recovery Techniques Log-based recovery, write-ahead logging, checkpointing, shadow paging, buffer management, handling system and media failures - Emerging & NoSQL Technologies (practical focus) Column-store, key-value, document, graph 	7

	<p>databases - when to use them for AI pipelines</p> <p>Distributed databases basics, data-partitioning, replication</p> <p>Data-warehouse concepts relevant to AI model training (star schema, OLAP cubes)</p> <p>- Laboratory Integration & Project Workflow</p> <p>End-to-end project: schema design, constraints, triggers, JDBC/ODBC access, role administration, loading AI training data, performance testing</p>	
--	--	--

6. References

Textbooks:

1. Data base System Concepts, Silberschatz, Korth, McGraw hill, Sixth Edition.(All UNITS except III th)
2. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition.

Reference Books:

1. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
2. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson, Eight Edition for UNIT III.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-503.1	Recall and list the core components of a database management system, the three-level schema architecture, and the primary database language categories (DDL, DML, DCL, TCL).	Recall	Remember
AM-503.2	Explain the purpose of conceptual data modeling and construct a correct ER/EER diagram for a given case study, then translate it into a normalized relational schema up to BCNF.	Explain	Understand
AM-503.3	Write and execute SQL statements--including SELECT with joins, subqueries, CTEs, window functions, and JSON	Write	Apply

	operators--to retrieve and manipulate AI-ready data as specified in problem scenarios.		
AM-503.4	Analyze a set of query workloads and, based on storage structures and indexing options, recommend and justify an optimal physical design (file organization, index type) to minimize I/O cost.	Analyze	Analyze
AM-503.5	Evaluate different concurrency-control and recovery techniques, and select appropriate isolation levels and logging strategies to ensure ACID compliance for concurrent AI model-training data loads.	Evaluate	Evaluate
AM-503.6	Design and implement a complete database solution for an AI/ML pipeline, encompassing schema design, integrity constraints, triggers, role-based security, data-ingestion scripts, and performance benchmarking, and produce a documented report of the results.	Design	Create

8. CO-PO Mapping

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

9. CO-PSO Mapping

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



Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Internet of Things
Course Code	AM-504
Semester	5
Course Category	Open Elective Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

- Proficiency in Java programming (core syntax, OOP concepts, and basic libraries)
- Fundamentals of computer networking (TCP/IP stack, OSI model, basic routing and wireless protocols)
- Basic understanding of embedded systems and Linux command-line environment (microcontroller basics, sensors/actuators, and shell usage)

2. Course Learning Objectives

- This course introduces students to fundamental concepts and applications of the subject
- Students will learn theoretical foundations and practical skills relevant to the field

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	<p>Foundations of IoT and Web Technologies</p> <ul style="list-style-type: none"> - IoT Overview <ul style="list-style-type: none"> * Evolution, key concepts, characteristics, typical use-cases (smart home, smart city, health-care, Industry 4.0) * Reference models (ITU, IETF, IoT-WF) and functional stack (sensors, actuators, gateways, cloud) - Core Web Foundations <ul style="list-style-type: none"> * Internet basics, client-server model, DNS, HTTP/HTTPS * HTML5 & CSS3 fundamentals, responsive design * JavaScript essentials for dynamic pages and AJAX - Java Foundations for IoT Services <ul style="list-style-type: none"> * Core Java syntax, OOP principles, collections, 	6

	<p>exception handling, basic multithreading</p> <ul style="list-style-type: none"> * Introduction to Java EE: Servlets, JSP, JavaBeans, JDBC, RESTful APIs (JAX-RS) <p>- Development Environment & Tooling</p> <ul style="list-style-type: none"> * IDEs (Eclipse/IntelliJ), build tools (Maven/Gradle), version control (Git) * Debugging and unit testing basics 	
2	<p>IoT Communication Layers & Protocols</p> <ul style="list-style-type: none"> - Physical & Link-Layer Technologies <ul style="list-style-type: none"> * IEEE 802.11 (Wi-Fi), 802.11ah (Wi-Fi HaLow), 802.15.4, ZigBee, Thread, BLE, LoRaWAN, NB-IoT * Trade-offs: range, power, topology, hardware constraints - Network Layer for IoT <ul style="list-style-type: none"> * IPv4 vs IPv6, 6LoWPAN, RPL (routing for low-power networks) * Addressing, DHCP, ND, basic ICMP concepts - Transport & Session Layer <ul style="list-style-type: none"> * TCP, UDP, CoAP (lightweight REST), MQTT (publish/subscribe), HTTP/2 * Security extensions: TLS, DTLS - Inter-operability & Service Standards <ul style="list-style-type: none"> * oneM2M, ETSI M2M, OMA Lwm2m - purpose and high-level architecture - Security Basics in Protocols <ul style="list-style-type: none"> * Link-layer security (IEEE 802.15.4 MAC), network-layer security (IPsec, DTLS), authentication models (pre-shared keys, certificates) 	8
3	<p>IoT Devices, Platforms and Edge Computing</p> <ul style="list-style-type: none"> - Edge Devices & Embedded Platforms <ul style="list-style-type: none"> * Sensors, actuators, RFID basics * Microcontrollers vs. System-on-Chip (Arduino, ESP32, Raspberry Pi, Nordic, STM32) - Fog, Edge & Cloud Architecture <ul style="list-style-type: none"> * Roles of edge, fog and cloud, latency vs. bandwidth considerations * Data-flow patterns (filter-aggregate-store) and placement decisions - IoT Platform Landscape <ul style="list-style-type: none"> * Open-source platforms (ThingsBoard, Node-RED, Eclipse Kura) - core services overview * Proprietary offerings (AWS IoT, Azure IoT Hub, Google Cloud IoT) - comparison of device management, rule engine, dashboards - Service Models for IoT <ul style="list-style-type: none"> * IaaS, PaaS, SaaS, DaaS - what they mean for developers * API-first design, MQTT broker as a service, 	7

	<p>REST endpoints</p> <ul style="list-style-type: none"> - Embedded Software Development Lifecycle * Requirements, design, coding, testing, OTA firmware updates, real-time constraints (no deep RTOS theory) 	
4	<p>Data Management and Analytics for IoT</p> <ul style="list-style-type: none"> - IoT Data Characteristics * Structured vs. semi-structured streams, time-series nature, metadata & tagging - Data Ingestion & Storage * Edge buffering strategies, time-series databases (InfluxDB, TimescaleDB), cloud storage buckets * Simple ETL pipelines for streaming data (Kafka -> Processor -> DB) - Foundations of IoT Analytics * Descriptive analytics (dashboards, alerts) * Introductory predictive analytics using lightweight models (rule-based, decision trees) that run on edge devices * Real-time stream processing concepts (Apache Kafka, Flink, Spark-Streaming - high-level view only) - Cloud-Native Analytics Services * Serverless functions (AWS Lambda, Azure Functions) for data transformation * Containerization basics (Docker) for deploying analytics micro-services * Case-study snapshots: smart-city traffic flow, predictive maintenance in manufacturing 	6
5	<p>IoT System Design, Architecture & Application Development</p> <ul style="list-style-type: none"> - Reference Architecture Views * Functional, information, deployment, and operational viewpoints (based on ISO/IEC/IEEE 42010) * Mapping constraints: power, latency, scalability, cost - System-Level Design Process * Requirement capture, use-case modeling, block diagram creation, capability mapping * Selecting communication patterns (publish/subscribe vs. request/response) and data models (JSON, CBOR) - Application Development Stack * Building RESTful services with Java (JAX-RS), integrating MQTT clients * Database access (JDBC, connection pooling, transaction basics) * Service discovery (Consul/Eureka) and 	7

	<p>orchestration basics (Docker-Compose, Kubernetes-lite)</p> <ul style="list-style-type: none"> - Prototyping, Testing & Performance <ul style="list-style-type: none"> * Simulation tools (Cooja, NS-3 - overview only), hardware-in-the-loop testing * Basic performance metrics (throughput, latency, power consumption) and benchmarking approaches 	
6	<p>IoT Security, Privacy, and Emerging Applications</p> <ul style="list-style-type: none"> - Security Foundations & Threat Landscape <ul style="list-style-type: none"> * Confidentiality, integrity, availability, privacy principles for IoT * Common attacker models (physical, network, cloud) and vulnerability assessment checklist - Device-Level Security Mechanisms <ul style="list-style-type: none"> * Secure boot, firmware signing, OTA update integrity * Key provisioning, lightweight PKI, DTLS/TLS session management - Emerging Security Approaches <ul style="list-style-type: none"> * Blockchain basics for device identity and immutable logs (conceptual only) * Zero-Trust networking principles applied to IoT segments - AI-Enabled Edge Applications <ul style="list-style-type: none"> * Edge AI inference basics (TensorFlow Lite, ONNX Runtime) - model deployment without deep math * Example use-cases: anomaly detection in sensor streams, smart-camera object detection - Compliance & Industry Practices <ul style="list-style-type: none"> * Overview of GDPR, IEC 62443, NIST 800-53 relevance to IoT deployments * Best-practice checklist for secure IoT system design 	8

6. References

Textbooks:

1. Olivier Hersent, "The Internet of Things Key Applications and Protocols", Wiley, 2012.
2. Sudip Misra, "Introduction to IoT", Cambridge University Press; First edition, 2021.

Reference Books:

1. Arshdeep Bahga, Vijay Madiseti, "Internet of Things: A Hands-on Approach", Universities Press, 2015.

2. Raj Kamal, "Internet of Things: Architecture and Design Principles", McGraw Hill Education private limited, 2017.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-504.1	Recall and enumerate the fundamental IoT concepts, reference models (ITU, IETF, IoT-WF), and core web technologies (HTML5, CSS3, JavaScript, HTTP/HTTPS) that form the basis of modern IoT systems.	Recall	Remember
AM-504.2	Explain the characteristics, trade-offs, and typical use-cases of the major IoT communication layers and protocols (e.g., Wi-Fi, BLE, LoRaWAN, 6LoWPAN, CoAP, MQTT, HTTP/2) and map them to latency, power, and scalability requirements.	Explain	Understand
AM-504.3	Develop a functional IoT service by implementing Java EE RESTful APIs, integrating an MQTT client, and deploying the solution with Maven/Gradle and Git version control.	Develop	Apply
AM-504.4	Analyze a specified IoT data-stream scenario to select suitable ingestion, storage, and lightweight analytics techniques, and construct a prototype ETL pipeline using Apache Kafka and a time-series database (InfluxDB or TimescaleDB).	Analyze	Analyze
AM-504.5	Evaluate common IoT security threats and countermeasures, and design a comprehensive security architecture that incorporates secure boot, firmware signing, DTLS/TLS, and zero-trust networking principles.	Evaluate	Evaluate
AM-504.6	Create an edge-AI application that deploys a TensorFlow Lite model for real-time anomaly detection on	Create	Create

	sensor data, integrates with the previously built IoT service, and measures performance against defined latency and accuracy criteria.		
--	--	--	--

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	2	1	2
CO2	3	3	2	2	2	1	1	1	1	2	1	2
CO3	3	2	3	2	3	1	1	1	2	2	2	2
CO4	3	3	3	3	3	2	2	1	2	2	2	3
CO5	3	3	3	3	2	3	2	3	2	2	2	2
CO6	3	2	3	2	3	2	1	2	2	2	2	3

9. CO-PSO Mapping

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



Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Embedded AI
Course Code	AM-505
Semester	5
Course Category	Open Elective Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

- Fundamentals of Digital Logic and Computer Organization (combinational/sequential logic, basic processor architecture)
- Proficiency in C programming (including pointers) and introductory assembly language concepts
- Understanding of basic data structures and algorithm analysis (arrays, linked lists, recursion, Big-O notation)

2. Course Learning Objectives

- This course introduces students to fundamental concepts and applications of the subject
- Students will learn theoretical foundations and practical skills relevant to the field

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	<p>Fundamentals of Embedded Systems & Computing</p> <ul style="list-style-type: none"> - Introduction to embedded systems and typical application domains - Design challenges, key metrics (performance, power, cost, reliability) - Embedded systems design flow and basic UML modelling (use-case, activity, class, component, sequence diagrams) - Processor families overview (Intel x86, ARM Cortex-M/A/R) and selection criteria - ARM Cortex-M architecture basics: pipeline, programmer's model, registers, memory map, interrupts & exceptions - Keil (or equivalent) development environment: 	5

	<ul style="list-style-type: none"> project set-up, build, debug cycle - Introductory assembly language programming for Cortex-M - Fundamentals of digital logic: combinational/sequential logic, bistable devices, state-machine concepts (Moore, Mealy, ASM charts) - VHDL/Verilog basics - data types, concurrent vs. sequential statements, behavioral modelling - Overview of popular microcontroller platforms (Arduino, ESP32, Raspberry Pi 4, Jetson Nano) and their resource constraints - Basic C programming concepts for embedded software (variables, control structures, functions, pointers) 	
2	<p>Digital Design, FPGA Technology & CAD Flow</p> <ul style="list-style-type: none"> - Synchronous sequential circuit design and algorithmic state machines (ASM) - Data-path and control-unit synthesis from specifications - Programmable logic families: PAL, PLD, CPLD, FPGA - when to use each - FPGA implementation flow: RTL coding, synthesis, technology mapping, placement & routing - Physical design concepts for FPGAs (partitioning, placement, routing, design-rule checks) - Logic-level optimisation techniques (two-level, multi-level, sequential optimisation) - Introduction to high-level synthesis and low-power design considerations for FPGAs - Hands-on labs using industry-grade tools (e.g., Synopsys or Cadence): 4-bit adder, shift register, sequence detector, simple datapath - Brief overview of FPGA-based soft-core processors (e.g., MicroBlaze, Nios II) for embedded AI prototypes 	8
3	<p>Data Structures, Algorithms & Computational Foundations</p> <ul style="list-style-type: none"> - Static vs. dynamic memory allocation, recursion and its practical uses - Complexity analysis fundamentals and Big-O notation - Searching techniques (linear, binary, recursive) and sorting algorithms (quick, merge, heap) - Linear data structures: arrays, stacks, queues, singly/doubly linked lists - implementation and use-cases - Tree structures: binary search trees, AVL trees, B-trees - insertion, deletion, traversal - Basic graph concepts and a single algorithm focus: 	6

	<p>shortest-path (Dijkstra) for embedded routing problems</p> <ul style="list-style-type: none"> - Number systems and computer arithmetic basics (binary, two's complement, floating-point representation) - Introductory machine-learning concepts relevant to embedded AI (feature vectors, training vs. inference, evaluation metrics) - Coding best practices for embedded software (modular design, error handling, code review) 	
4	<p>Embedded AI Foundations & Lightweight Machine Learning</p> <ul style="list-style-type: none"> - Edge vs. cloud AI paradigms and why edge inference matters - Resource constraints on edge devices (memory, latency, power budget) - Hardware platforms for edge AI: Arduino Nano 33 BLE Sense, ESP32, Raspberry Pi, Jetson Nano, Coral TPU, PYNQ, BeagleBone AI - Lightweight ML models suitable for microcontrollers: linear/logistic regression, decision trees, tiny fully-connected nets, shallow CNNs - Model-compression basics: pruning, quantisation, knowledge distillation - concepts, not heavy math - Data acquisition and preprocessing for sensor-driven AI (sampling, filtering, feature extraction) - Introduction to TensorFlow Lite, TensorFlow Lite Micro, ONNX Runtime, and the Edge Impulse ecosystem - End-to-end workflow: collect data on host, train model, convert to on-device format, validate accuracy 	7
5	<p>Model Optimisation, Deployment & Real-Time Embedded AI Applications</p> <ul style="list-style-type: none"> - Quantisation strategies (dynamic, static, quantisation-aware training) and their impact on accuracy - Pruning, weight sharing and knowledge distillation for model size reduction - Benchmarking inference: measuring latency, memory footprint, and power consumption on target hardware - Deployment on microcontrollers (Arduino Nano 33 BLE Sense, ESP32): sensor interfacing, interrupt-driven AI, low-power modes - Deployment on single-board computers (Raspberry Pi, Jetson Nano/Xavier): Linux-based inference, use of hardware accelerators (Coral TPU, GPU) - Real-time AI application examples: object 	8

	<p>detection, keyword spotting, gesture control, human-activity recognition, fault detection, predictive maintenance</p> <ul style="list-style-type: none"> - Performance profiling tools and optimisation loop (profiling -> bottleneck analysis -> optimisation) - Basics of real-time operating system (RTOS) scheduling for AI tasks and brief security considerations for edge AI (secure boot, model integrity) 	
6	<p>Laboratory Integration, Projects & Specialized Topics</p> <ul style="list-style-type: none"> - Hardware Systems Lab: Boolean algebra, ROBDD, graph-based optimisation, scheduling and binding techniques - Software Systems Lab: assembly programming, device-driver development, object-oriented interfacing, debugging (JTAG, ISP) - System Design Lab: design and verification of a full-adder, 4-bit GCD computer, Booth multiplier, 4-point FFT/IFFT, CORDIC, floating-point unit, BIST generation - Mini-project: design, implement and demonstrate a complete embedded AI system (sensor -> model -> real-time output) using an approved platform - Elective / specialised modules (selected per programme): VLSI testing, formal verification, MEMS/NEMS, advanced computer architecture, IoT & IIoT protocols, computer vision basics, probabilistic machine learning, parallel & distributed systems - Software engineering practices for embedded projects: version control (Git), unit testing, continuous integration, documentation, ethical considerations - Project & dissertation guidelines, evaluation criteria, and presentation skills 	8

6. References

Textbooks:

1. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers - Pete Warden & Daniel Situnayake(O'Reilly)
2. Embedded Artificial Intelligence: Principles, Platforms and Practices -- Bin Li; Springer Singapore, 2024.

Reference Books:

1. Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Software Optimizations and Hardware/Software Codesign, Springer.
2. Embedded Machine Learning with Microcontrollers: Applications on Arduino Boards -- Cem Ünsalan, Berkan Höke, Eren Atmaca; Springer Cham.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-505.1	Recall and list the key characteristics of embedded system architectures, processor families (e.g., ARM Cortex-M), digital-logic components, and the basic C and assembly syntax required for microcontroller programming.	Recall	Remember
AM-505.2	Explain the end-to-end design flow for embedded hardware and software, including UML use-case/activity diagrams, FPGA synthesis steps, and the edge-AI inference pipeline, describing the purpose of each stage.	Explain	Understand
AM-505.3	Implement a functional prototype by writing C code for sensor acquisition on an Arduino Nano 33 BLE Sense, developing a VHDL module for a 4-bit adder, and deploying a TensorFlow Lite Micro keyword-spotting model that achieves ≤ 80 ms inference latency on the target device.	Implement	Apply
AM-505.4	Analyze the impact of processor selection, memory hierarchy, and model-compression techniques (quantisation, pruning, knowledge distillation) on latency, power consumption, and classification accuracy for a specified edge-AI task.	Analyze	Analyze
AM-505.5	Evaluate alternative optimisation strategies for a deployed AI model on an ESP32 by using profiling tools to measure latency, memory footprint, and energy use, and recommend modifications that improve inference speed by at least 30% while preserving ≥ 90 % of the baseline	Evaluate	Evaluate

	accuracy.		
AM-505.6	Create and demonstrate a complete real-time embedded AI application (e.g., a gesture-controlled robot) that integrates hardware interfacing, RTOS task scheduling, optimized ML inference, and comprehensive documentation, delivering a working system that meets predefined functional and performance criteria.	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	1	1	1	1	1	1	2
CO2	3	2	3	2	3	1	1	1	2	2	1	2
CO3	3	2	3	2	3	1	1	1	2	2	2	2
CO4	2	3	2	3	2	1	2	1	1	1	1	2
CO5	2	3	3	3	3	1	2	1	2	2	2	2
CO6	3	3	3	3	3	2	2	2	3	3	3	3

9. CO-PSO Mapping

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



Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(AIML)

Field	Details
Course Name	Bio-informatics and Computational Biology
Course Code	AM-506
Semester	5
Course Category	Open Elective Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

- Fundamental concepts in molecular and cellular biology (DNA/RNA/protein basics, cell structure, and metabolism)
- Proficiency in Python programming and basic command-line data handling (including file formats like FASTA/FASTQ and version control with Git)
- Introductory statistics and probability plus basic linear algebra (understanding of distributions, hypothesis testing, and matrix operations)

2. Course Learning Objectives

- This course introduces students to fundamental concepts and applications of the subject
- Students will learn theoretical foundations and practical skills relevant to the field

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

5. Course Modules

Module	Topics	Hours
1	<p>Fundamentals of Life Sciences and Molecular Biology</p> <ul style="list-style-type: none"> - Scope of life sciences and major branches - Levels of biological organization (molecule -> ecosystem) - Core concepts of biochemistry relevant to bioinformatics (proteins, nucleic acids, lipids, carbohydrates) - Cell structure: prokaryotic vs. eukaryotic, organelles, membrane systems - Cell division basics: mitosis and meiosis - Energy metabolism fundamentals (thermodynamics, glycolysis, TCA cycle, respiration, photosynthesis) - Central Dogma in detail: DNA replication, transcription, translation - Genetic code features (codon usage, wobble, GC content) - Overview of RNA processing (splicing, editing, antisense strands) - Introduction to programming for biologists (Python basics, handling FASTA/FASTQ files) - Data-handling best practices and ethical considerations (FAIR principles, privacy of genomic data) 	5

2	<p>Introduction to Bioinformatics and Biological Databases</p> <ul style="list-style-type: none"> - Definition and scope of bioinformatics; role in modern biotechnology - Primary nucleotide databases (EMBL-EBI, GenBank, DDBJ) and accession concepts - Protein databases (UniProtKB/Swiss-Prot, TrEMBL, PIR) - Specialized resources: Pfam, PROSITE, InterPro (motif & domain), PDB, SCOP, CATH (structure) - Pathway and reaction databases (KEGG, Reactome, BioCyc) - Entrez, NCBI utilities and programmatic access (E-utilities, REST APIs) - Genome browsers (UCSC, Ensembl, JBrowse) - navigation and custom tracks - Practical session: retrieving sequences, annotations, and metadata from GenBank & UniProt - Version control and reproducible pipelines (Git, GitHub, workflow managers - Snakemake/Nextflow basics) - Cloud-based resources for bioinformatics (Google Cloud, AWS, Galaxy) and data-sharing standards 	6
3	<p>Sequence Alignment, Search Tools, and Motif Discovery</p> <ul style="list-style-type: none"> - Principles of pairwise alignment; dot-matrix visualization - Dynamic-programming algorithms: global (Needleman-Wunsch) and local (Smith-Waterman) - conceptual overview - Scoring schemes: substitution matrices (PAM, BLOSUM) and gap penalties - Interpreting alignment significance (e-value, bit score) without heavy statistics - BLAST family of tools (blastn, blastp, blastx, tblastn, tblastx) - practical use cases - Multiple sequence alignment methods (Clustal Omega, MAFFT, MUSCLE) and alignment quality assessment - Visualization of alignments and sequence logos (Jalview, WebLogo) - Motif-finding concepts: consensus patterns, MEME suite, and simple probabilistic models - Introductory AI/ML for pattern detection - k-mer based classifiers and basic feature extraction - Building and querying profile HMMs at a conceptual level (no deep math) 	7
4	<p>Phylogenetics, Evolutionary Analysis, and Gene Prediction</p>	8

	<ul style="list-style-type: none"> - Evolutionary concepts: homology, orthology, paralogy, molecular clocks (qualitative) - Substitution models overview (JC, K2P) - when to choose a model - Tree-building methods: distance-based (NJ, UPGMA) and maximum-likelihood basics (IQ-TREE, FastTree) - hands-on sessions - Interpreting phylogenetic trees (rooting, bootstrap support) and common pitfalls - Gene prediction strategies: ab-initio vs. evidence-driven (Augustus, GeneMark) - practical demo - Genome annotation workflow: structural annotation, functional annotation, GFF3 format - Use of annotation pipelines (NCBI's PGAP, MAKER) and best-practice guidelines - Comparative genomics snapshots: gene family analysis, synteny browsers - Brief discussion of AI-enhanced annotation tools (deep-learning gene predictors) - conceptual overview 	
5	<p>Genomics, Transcriptomics, Metagenomics & Omics Data Analysis</p> <ul style="list-style-type: none"> - Whole-genome sequencing pipelines: quality control, de-novo assembly (SPAdes, Flye) and assembly assessment (QUAST) - Annotation of assembled genomes (Prokka, RAST) - practical exercise - RNA-Seq workflow: library design, read QC, alignment (STAR/HISAT2), quantification (featureCounts), differential expression (DESeq2/edgeR) - Introduction to single-cell RNA-Seq concepts (brief, industry relevance) - Metagenomics basics: shotgun sequencing, taxonomic profiling (Kraken2, MetaPhlAn), assembly of metagenome-assembled genomes (MAGs) - Variant discovery: SNP and indel calling (GATK best practices), CNV detection basics - Genome-wide association studies (GWAS) - study design and interpretation without heavy statistics - Data integration across omics layers; introduction to multi-omics clustering and dimensionality reduction (PCA, t-SNE, UMAP) using Python/R - AI/ML applications in omics: supervised classification for disease prediction, unsupervised clustering for patient stratification - Visualization of omics results (IGV, Circos, Plotly, Shiny dashboards) 	8

6	Structural Bioinformatics, Proteomics, and Systems-Biology Applications	8
	<ul style="list-style-type: none"> - Relationship between protein sequence, structure, and function - case studies - Protein structure prediction approaches: comparative modeling, threading, and recent deep-learning methods (AlphaFold overview) - Model quality assessment (Ramachandran plot, MolProbity) and structure validation - Functional annotation from structure: active-site prediction, ligand-binding site identification - RNA secondary-structure prediction tools (Mfold, RNAstructure) and comparative analysis - Mass-spectrometry basics for proteomics: peptide identification workflows (Mascot, MaxQuant) - Protein-protein interaction networks: data sources (STRING, BioGRID) and network visualization (Cytoscape) - Pathway databases and enrichment analysis (KEGG, Reactome, GSEA) - practical session - Systems-biology modeling: deterministic (ODE) vs. stochastic approaches, introduction to SBML and simulation tools (COPASI) - AI-driven systems biology: network inference using machine learning, predictive modeling of metabolic pathways 	

6. References

Textbooks:

1. Baxevanis, Andreas D. et. Al., Bioinformatics, (4th ed.), John Wiley & Sons, 2020.

Reference Books:

1. Mount, David W. "Bioinformatics: sequence and genome analysis." (2004).
2. Lesk, Arthur. Introduction to Bioinformatics. United Kingdom: OUP Oxford, 2014.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-506.1	Recall the fundamental concepts of life sciences and molecular	Recall	Remember

	biology, including the levels of biological organization, central-dogma processes, cell structure, energy-metabolism pathways, and FAIR data-handling principles.		
AM-506.2	Explain the purpose and structure of major biological databases and bioinformatics tools, and retrieve specific nucleotide or protein sequences and associated metadata using command-line utilities and programmatic APIs.	Explain	Understand
AM-506.3	Apply pairwise and multiple sequence alignment methods and motif-discovery tools to provided datasets using Python scripts and command-line software, and assess alignment quality with appropriate scores and visualizations.	Apply	Apply
AM-506.4	Analyze phylogenetic trees and gene-prediction outputs by selecting suitable substitution models, constructing trees with distance-based and maximum-likelihood methods, and interpreting bootstrap support and annotation results.	Analyze	Analyze
AM-506.5	Evaluate the performance of AI/ML approaches for multi-omics data analysis--such as k-mer classifiers, dimensionality-reduction clustering (PCA, t-SNE, UMAP), and disease-prediction models--using appropriate validation metrics.	Evaluate	Evaluate
AM-506.6	Design and implement a reproducible end-to-end bioinformatics pipeline that integrates AI/ML components (e.g., deep-learning protein-structure prediction, network inference) to solve a defined biological research problem, and communicate the results through interactive visualizations and a written report.	Design	Create

8. CO-PO Mapping

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

9. CO-PSO Mapping

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