SARDAR PATEL UNIVERSITY, BALAGHAT
School of Engineering and Technology
Syllabus

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

AIML401 Applied Machine Learning

Course Objectives:

1. Understand basic concepts and types of machine learning.

2. Learn regression models and model selection techniques.

3. Explore SVMs and tree-based algorithms.

4. Study neural networks and deep learning basics.

5. Apply clustering and association rule techniques.

Unit I: Basics of Machine Learning:

Introduction to Machine Learning, Examples of Machine Learning Applications, Learning Types Supervised Learning -Learning a Class from Examples, Vapnik- Chervonenkis (VC) Dimension, Probably Approximately Correct (PAC) Learning, Noise, Learning Multiple Classes, Regression, Model Selection and Generalization, Dimensions of a Supervised Machine Learning Algorithm Dimensionality Reduction- Introduction, Subset Selection, Principal Components Analysis, Factor Analysis, Multidimensional Scaling, Linear Discriminant Analysis, Isomap, Locally Linear Embedding.

Unit II: Regression:

Introduction, Linear Regression Models and Least Squares, Subset Selection, Shrinkage Methods-Ridge Regression, Lasso Regression, Least Angle Regression, Methods Using Derived Input Directions-Principal Components Regression, Partial Least Squares, A Comparison of the Selection and Shrinkage Methods , Multiple Outcome Shrinkage and Selection, More on the Lasso and Related Path Algorithms, Logistic Regression-Fitting Logistic Regression Models, Quadratic Approximations and Inference, L1 Regularized Logistic Regression.

Unit III: Support Vector Machine:

Introduction to SVM, The Support Vector Classifier, Support Vector Machines and Kernels-Computing the SVM for Classification, The SVM as a Penalization Method, Function Estimation and Reproducing Kernels, SVMs and the Curse of Dimensionality, A Path Algorithm for the SVM Classifier, Support Vector Machines for Regression, Regression and Kernels Tree Based Methods-Regression Trees, Classification Trees, Random Forests- Definition of Random Forests, Details of Random Forests- Out of Bag Samples, Variable Importance, Proximity Plots, Random Forests and Over-fitting, Analysis of Random Forests-Variance and the De-Correlation Effect, Bias, Adaptive Nearest Neighbors.

Unit IV: Neural Networks:

Introduction-Understanding the Brain, Neural Networks as a Paradigm for Parallel Processing, The

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

Perceptron, Training a Perceptron, Learning Boolean Functions, Multilayer Perceptrons, MLP as a Universal Approximator, Back-propagation Algorithm-Nonlinear Regression, Two-Class Discrimination, Multiclass Discrimination, Multiple Hidden Layers, Training Procedures-improving Convergence, Overtraining, Structuring the Network, Tuning the Network Size, Bayesian View of Learning, Dimensionality Reduction, Learning Time-Time Delay Neural Networks, Recurrent Networks, Regularization in Neural Networks, Bayesian Neural Networks.

Unit V: Clustering Techniques:

Introduction, Association Rules-Market Basket Analysis, The Apriori Algorithm, Unsupervised as Supervised Learning, Generalized Association Rules, Cluster Analysis-Proximity Matrices, Clustering Algorithms-K-mean, Gaussian Mixtures as Soft K-means Clustering, Example: Human Tumor Microarray Data, Vector Quantization, K-medoids, Hierarchical Clustering, Self-Organizing Maps, PCA-Spectral Clustering.

Reference Books:

1. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.
2. The Elements of Statistical Learning. Trevor Hastie, Robert Tibshirani and Jerome Friedman. Second Edition. 2009.
3. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
4. Pattern Recognition and Machine Learning. Christopher Bishop. Springer. 2006.
5. Understanding Machine Learning. ShaiShalev-Shwartz and Shai Ben-David. Cambridge University Press. 2017.

Course Outcome:

At the end of the course, students should be able to:

1. Apply suitable ML techniques for various problems.

2. Build and evaluate regression and classification models.

3. Use SVMs and tree methods effectively.

4. Design and train neural networks.

5. Perform clustering and discover patterns in data.

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

AIML402 Introduction to Drones

Course Objectives:

1. Understand the basic concepts, types, and components of drones.

2. Learn fundamental principles of drone flight and control systems.

3. Explore software, sensors, communication, and data processing used in drones.

4. Understand applications of drones in computing domains such as AI, IoT, and data analytics.

5. Gain awareness of drone regulations, safety, and future trends.

Unit I: Introduction to Drones
Definition and evolution of drones, UAV, UAS, and RPAS – terminology, Classification of drones: Fixed-wing, Rotary-wing, Hybrid, Civilian and commercial drones, Basic drone architecture, Overview of drone applications

Unit II: Drone Components and Hardware
Frame, motors, propellers, ESC, Flight controller – role and types, Power systems: batteries and charging, Sensors: GPS, IMU, Gyroscope, Accelerometer, Magnetometer, Cameras and payloads, Introduction to onboard computing (Raspberry Pi, Pixhawk)

Unit III: Drone Software and Control Systems
Introduction to drone operating systems, Flight control software basics, Ground Control Stations (GCS), Communication: Radio frequency, telemetry, Wi-Fi, Basics of drone navigation, Introduction to simulation tools (e.g., simple drone simulators)

Unit IV: Drone Applications in Computer Science
Drones and Internet of Things (IoT), Computer Vision applications: Object detection, Surveillance, Mapping, AI and Machine Learning in drones (introductory level), Data collection and processing using drones, Case studies: Smart cities, Agriculture, Disaster management.

Unit V: Regulations, Safety, and Future Trends
Drone safety and operational guidelines, Overview of drone regulations (DGCA / general global view), Ethical and privacy concerns, Challenges in drone technology, Future trends: Autonomous drones, Swarm drones, Drone delivery systems. Career Opportunities and Research Areas.

Reference Books:
1. John Baichtal, Building your own drones, a beginner's guide to drones, UAVS, and ROVs‖ Pearson Education, 1st Edition,2015
2. Muhammad Usman , Quadcopter modeling and control with Matlab/Simulink implementation‖ LAB University of Applied Sciences, 1st Edition, 2020

3. Austin, R. – Unmanned Aircraft Systems: UAVS Design, Development and Deployment, Wiley Publishing.
4. K.S.Fu, R.C.Gonzalez, C.G.Lee , Robotics control, sensing, vision and intelligence‖ MGH, 1st Edition, 1987
5. R.K.Mittal , I.J.Nagrath, Robotics and control Tata McGraw-Hill, 1st Edition,2005
6. Ben Rupert , ―Drones (The ultimate guide), Create Space Independent Publishing Platform, 1st Edition,2017
7. Agam Kumar Tyagi, Matlab and Simulink for engineers‖, Oxford University Press, 1st Edition,2012

Course Outcome:

At the end of the course, students should be able to:

1. Explain basic drone terminology, types, and working principles.

2. Identify and describe major drone hardware and software components.

3. Understand basic control, navigation, and communication techniques used in drones.

4. Analyze drone applications involving computer vision, AI, and data processing.

5. Demonstrate awareness of drone laws, safety practices, and ethical issues.

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

AIML403 Theory of Computation & Compiler Design

Course Objectives:

1. To provide fundamentals of formal languages and automata theory.
2. To understand models of computation and language classification.
3. To study the structure and phases of a compiler.
4. To learn lexical, syntax analysis, and syntax-directed translation.
5. To understand code generation and optimization techniques.

Unit I: Formal Languages & Finite Automata:

Language, Grammar, Automata, Review of sets and proof techniques, Deterministic Finite Automata (DFA): Definition, Representation, Language. Non- Deterministic Finite Automata (NFA) and ε-NFA, Equivalence of DFA and NFA, Elimination of ε-transitions and DFA Minimization, Applications of Finite Automata, Moore and Mealy machines, inter conversion.

Unit II: Regular & Context-Free Languages:

Regular Expressions and identities, Conversion between Regular Expressions and Finite Automata, Regular Grammars and FA Equivalence, Closure properties of Regular Languages, Pumping Lemma for regular languages (applications only), Introduction to Context-Free Grammars and Chomsky Hierarchy.

Unit III: Introduction to Compiler & Lexical Analysis:

Introduction to compiler, phases and structure, Analysis–synthesis model, symbol table, Lexical analysis: role of lexical analyzer, input buffering, Specification and recognition of tokens, LEX tool, Finite automata in lexical analysis.

Unit IV: Syntax Analysis & Syntax-Directed Translation:

Syntax analysis: CFGs in parsing. Top-down parsing: recursive descent, predictive parsing. Bottom-up parsing: Operator Precedence Parsing, LR Parsing: SLR, LALR, LR. Parser Generators.
Syntax- directed definitions and translation schemes, S-attributed and L-attributed definitions, Construction of Syntax Trees.

Unit V: Intermediate Code Generation, Code Generation & Optimization:

Intermediate code generation: Three-address code, control flow statements, Backpatching. Type checking: type systems, type equivalence, type conversion. Code generation: Basic blocks and flow graphs. Register allocation and assignment. DAG Representation of Basic blocks. Code Optimization: Peephole Optimization. Loop optimization. Dead Code Elimination. Introduction to data flow analysis.

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, 3rd edition, Pearson Education, India.
2. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.

Reference Books:

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
2. Lex &Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly.
3. Compiler Construction, Louden, Thomson.

Course Outcome:

At the end of the course, students should be able to:

1. Apply concepts of automata and formal languages.

2. Classify languages using regular and context-free grammars.

3. Explain compiler structure and lexical analysis.

4. Analyze syntax parsing and translation techniques.

5. Implement code generation and optimization methods.

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

AIML404 Computer Networking

Course Objectives:

1. To provide students with a comprehensive understanding of Computer Networking concepts, architectures and protocols.

2. To familiarize students with key networking protocols at different layers of the OSI and TCP/IP models.

3. To equip students with the skills to design, implement, and manage computer networks, including both wired and wireless technologies.

Unit I: Introduction to Computer Network:
Computer Network: Definitions, goals, components, Architecture, Classifications & Types. Layered Architecture: Protocol hierarchy, Design Issues, Interfaces and Services, Connection Oriented & Connectionless Services, Service primitives, Design issues & its functionality. ISO- OSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP.

Unit II: Data Link Layer:
Data Link Layer: Need, Services Provided, Framing , Flow Control, Error control. Data Link Layer Protocol: Elementary & Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Bit oriented protocols: SDLC, HDLC, BISYNC, LAP and LAPB.

Unit III: Mac Sub-layer:
MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted- ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA), Collision Free Protocols: Basic Bit Map, BRAP, Binary Count Down, MLMA Limited Contention Protocols: Adaptive Tree Walk, URN Protocol, High Speed LAN: Fast Ethernet, Gigabit Ethernet, FDDI, Performance Measuring Metrics. IEEE standards 802 series & their variant.

Unit IV: Network Layer:
Network Layer: Need, Services Provided , Design issues, Routing algorithms: Least Cost Routing algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing, Broadcast Routing, Multicast Routing, Congestion Control Algorithms: General Principles of Congestion control, Prevention Policies, Congestion Control in Virtual-Circuit Subnets, Congestion Control in Datagram subnets. IP Protocol, IP address, Comparative study of IPv4 & IPv6, Mobile IP.

Unit V: Transport Layer:
Design Issues, UDP: Header Format, Per-Segment Checksum, Carrying Uni-cast /Multi-cast Real-Time Traffic, TCP: Connection Management, Reliability of Data Transfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP Timer Management. Session layer: Authentication,

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

Authorization, Session layer protocol (PAP, SCP, H.245). Presentation layer: Data conversion, Character code translation, Compression, Encryption and Decryption, Presentation layer protocol (LPP, Telnet, X.25 packet Assembler/ Dis-assembler). Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).

Reference Books:
1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks" Pearson Education.
2. Dimitri Bertsekas, Robert Gallager, "Data Networks", PHI Publication, Second Edition.
3. Kaveh Pahlavan, Prashant Krishnamurthy, "Networking Fundamentals", Wiley Publication.
4. Uyless Black, "Computer Networks", PHI Publication, Second Edition.
5. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source approach", McGraw Hill.

List of Experiments:
1. Study of Different Type of LAN& Network Equipments.
2. Study and Verification of standard Network topologies i.e. Star, Bus, Ring etc.
3. LAN installations and Configurations.
4. Write a program to implement various types of error correcting techniques.
5. Write a program to implement various types of framing methods.
6. Study of Tool Command Language (TCL).
7. Study and Installation of Standard Network Simulator: N.S-2, N.S-3.OpNet, QualNet etc.
8. Study & Installation of ONE (Opportunistic Network Environment) Simulator for High Mobility Networks
9. Configure 802.11 WLAN.
10. Implement & simulate various types of routing algorithm.
11. Study & Simulation of MAC Protocols like Aloha, CSMA, CSMA/CD and CSMA/CA using Standard Network Simulators.
12. Study of Application layer protocols- DNS, HTTP, HTTPS, FTP and TelNet.

Course Outcome:

At the end of the course, students should be able to:

1. Describe fundamental networking concepts, including the OSI and TCP/IP models, and their respective layers.

2. Analyze and explain the operation of various network protocols, such as IP, TCP, UDP, HTTP, and DNS.

3. Design and implement basic network configurations using routers, switches and wireless access points.

Course: B.Tech   **Semester: IV**

Branch: Computer Science and Engineering (AI & ML)   **w.e.f. Academic Session: 2024-25**

AIML405 Data Handling and Visualization

Course Objectives:

1. To introduce the fundamentals of data science and data visualization concepts.

2. To understand data preprocessing and exploratory data analysis using R.

3. To apply statistical analysis and basic machine learning techniques for data analysis.

4. To develop interactive data visualizations and dashboards using Power BI.

5. To integrate R and Power BI for advanced data visualization and storytelling.

Unit I: Introduction to Data Science and Data Visualization:

Data Science Introduction: Concepts, lifecycle, applications. - Role of Data Visualization in Analysis and Decision Making. Basics of R Programming: Variables, data types, operators. Data Visualization Fundamentals: Principles, visualization types.

Unit II: Data Preprocessing and EDA with R:

Data Collection and Sources: Structured, unstructured, web scraping. Data Cleaning: Handling missing data, outliers. Data Transformation Techniques: Normalization, standardization, encoding. Exploratory Data Analysis (EDA): Univariate, bivariate, multivariate analysis. Advanced EDA Plotting : ggplot2 for customized visualizations, faceting, distributions.

Unit III: Advanced Data Analysis and Visualization with R:

Statistical Analysis: Descriptive stats, hypothesis testing. Data Visualization Libraries in R: ggplot2. Machine Learning Concepts: Introduction to ML, basic models in R. R Shiny: Building interactive web applications.

Unit IV: Power BI for Data Visualization and Dashboard Creation:

Introduction to Power BI: Interface, data connection, roles. Creating Basic Visualizations: Bar charts, line charts, scatter plots. Building Interactive Dashboards: Design principles, combining visualizations. Effective Data Storytelling using Power BI.

Unit V: Advanced Data Visualization and Integration:

Advanced visualization techniques in R. Integrating R with Power BI: Using R scripts and calculations. Data visualization ethics and best practices. Capstone Project: Applying skills using R and Power BI.

Text Books / Reference Books:
1. R for Data Science by Hadley Wickham and Garrett Grolemund.
2. Hands-On Data Visualization with R by Claus O. Wilke.
3. Power BI for Data Science by Ryan Sleeper .
4. Data Visualization with Power BI by Daniel Murray.
5. Data Storytelling with Power BI by Anupam Jain.

6. The Visual Display of Quantitative Information by Edward Tufte.
7. Data Visualization: A Practical Introduction by Kieran Healy.
8. The Functional Art of Data visualization by Alberto Cairo.
9. Information Visualization: Perception for Design by Colin Ware.

Course Outcome:

At the end of the course, students should be able to:

1. Explain data science concepts and the role of data visualization in decision making.

2. Perform data preprocessing and exploratory data analysis using R.

3. Apply statistical analysis and basic machine learning models in R.

4. Create interactive dashboards and visual reports using Power BI.

5. Integrate R with Power BI and apply best practices in data visualization.

Course: B.Tech   Semester: IV

Branch: Computer Science and Engineering (AI & ML)   w.e.f. Academic Session: 2024-25

AIML406P Java Lab

Course Objectives:

1. To introduce the fundamentals and syntax of Java programming.

2. To understand object-oriented programming concepts using Java.

3. To learn exception handling and multithreading for efficient program execution.

4. To explore the Java Collections Framework and its applications.

5. To understand advanced Java concepts including JDBC, networking, GUI, and Java applications in AI.

Unit I – Fundamentals of Java Programming:

Introduction to Java: History and evolution of Java, Features: platform independence, robustness, security, portability, JVM, JDK, JRE architecture. Basic Syntax: Data types, variables, operators, Control structures: if-else, switch, loops (for, while, do-while). Arrays and Strings: Single and multidimensional arrays, String class, String methods, StringBuffer and StringBuilder. Input/Output Basics: Scanner Class, Command-line Arguments.

Unit II – Object-Oriented Programming Concepts:

Classes and Objects: Defining classes, creating objects, Instance variables, methods. Constructors: Default, parameterized, copy constructors, Constructor chaining. Inheritance: Types: single, multilevel, hierarchical, Method overloading and overriding, Use of super keyword. Polymorphism: Compile-time and runtime polymorphism. Abstract Classes and Interfaces: Abstract methods, abstract classes, Interfaces and multiple, inheritance through interfaces. Packages: Creating and using packages, Access modifiers.

Unit III – Exception Handling & Multithreading:

Exception Handling: Types of exceptions: checked, unchecked, Try-catch-finally, throw, throws, User-defined exceptions. Multithreading: Thread lifecycle, Creating Threads: extending Thread class, implementing Runnable, Thread methods: sleep, join, yield, priority, Synchronization and inter-thread communication. Applications in AI: Parallel data processing, Multithreading for training ML models.

Unit IV – Java Collections Framework:

Introduction to Collections: Need for collections vs arrays, Interfaces: Collection, List, Set, Map. List Implementations: ArrayList, LinkedList, Vector. Set Implementations: HashSet, LinkedHashSet, TreeSet. Map Implementations: HashMap, LinkedHashMap, TreeMap. Generics: Type safety in collections, Generic classes and methods. Iterators: Iterator and ListIterator. Applications in AI: Managing datasets, Graph structures for search algorithms.

Unit V – Advanced Java Concepts:

Database Connectivity (JDBC): JDBC architecture, Steps to connect Java with databases, Executing Queries (insert, update, delete, select). Networking: Basics of networking in Java, Socket

programming (client-server), URL and HTTP communication. GUI Programming: Swing basics: JFrame, JButton, JTextField, JLabel, Event handling, JavaFX introduction. Java in AI: Overview of Java-based AI libraries (Weka, Deeplearning4j), Case studies: using Java for ML preprocessing and visualization.

Text Books:
1. Herbert Schildt – Java: The Complete Reference (McGraw Hill)
2. Balagurusamy – Programming with Java (McGraw Hill)

Reference Books:
1. Cay S. Horstmann – Core Java Volume I & II
2. Joshua Bloch – Effective Java
3. Dietel&Dietel – Java How to Program
4. Patrick Naughton& Herbert Schildt – The Java Handbook

Course Outcome:

At the end of the course, students should be able to:

1. Write Java programs using basic syntax, control structures, arrays, and strings.

2. Apply object-oriented concepts such as inheritance, polymorphism, and interfaces.

3. Implement exception handling and multithreaded applications.

4. Use Java Collections and generics for effective data management.

5. Develop basic applications using JDBC, networking, GUI, and Java-based AI tools.

List of Experiments:
1. Write a program to check whether a number is prime.
2. Implement Fibonacci series using recursion.
3. Create a class Student with attributes and display details using constructors.
4. Demonstrate method overloading and overriding with a Shape class.
5. Write a program to implement single and multilevel inheritance.
6. Create an interface Vehicle and implement it in Car and Bike classes.
7. Write a program to handle ArrayIndexOutOfBoundsException.
8. Implement a user-defined exception for invalid age input.
9. Create two threads: one prints even numbers, another prints odd numbers.
10. Demonstrate thread synchronization using a shared resource.
11. Write a program to store and retrieve student records using ArrayList.
12. Implement a program using HashMap to store employee IDs and names.
13. Demonstrate use of TreeSet to store sorted data.
14. Implement a program to demonstrate generics with a collection.

15. Create a program to connect Java with a database (JDBC) and insert records.
16. Write a program to fetch data from a database and display it.
17. Implement a simple chat application using sockets.
18. Create a GUI application using Swing for a calculator.
19. Write a program to demonstrate use of StringBuffer for string manipulation.
20. Implement a program to demonstrate LinkedList operations.