CURRICULUM STRUCTURE OF T. Y. B. TECH

Effective from A. Y. 2025-26

Semester V

Sr. No.	Course Code	Course Name	Teaching Scheme			Credits
			L	T	P	
1	PCC-16	Software Engineering	3	0	2	4
2	PCC-09	Database Management Systems	3	0	2	4
3	PCC-10	Operating Systems	3	0	2	4
4	PEC-01	DE-1	2	0	2	3
5	OE-03	Fundamentals of Machine Learning	2	0	0	2
6	OJT-01	Internship	--	--	2	1
		Project Stage 1	--	--	4	2
7	MD M-02	MDM – Quantum Computing / AI / CS/ Database	3	0	0	4
		Total Academic Credits				**24**

Semester VI

Sr. No.	Course Code	Course Name	Teaching Scheme			Credits
			L	T	P	
1	PCC-11	Computer Networks	3	0	2	4
2	PCC-08	Design and Analysis of Algorithms	3	0	2	4
3	PCC-13	Artificial Intelligence	3	0	0	3
4	PES-2	DE-2	3	0	2	4
5	RM	Research Methodology	2	0	0	2
6	VSEC-02	Project Stage 2	0	0	4	2
7	MD M-03	MDM – Quantum Computing / AI / CS/ Database	3	0	0	4
		Total Academic Credits				**23**

(PCC-16) Software Engineering

Teaching Scheme:

Lectures : 3 Hrs/week

Examination Scheme:

Mid Semester Exam: 30 marks

End Semester Exam: 60 Marks

Quiz/Assignment/Faculty Internal Assessment:10 Marks

Course Outcomes

Students will be able to:

1. Describe fundamental concepts of system development lifecycle through SDLC.
2. Design user interface prototypes for real world scenarios using appropriate methods of analysis and design.
3. Devise procedure to assure the quality and maintainability of the product before and after deployment.
4. Develop skill to transfer acquired knowledge across a wide range of industrial and commercial domains and have a basis for further studies in software engineering or in computing related industries.
5. Analyse real world scenario and apply tools and techniques to produce application software solutions from informal and semiformal problem specifications.
6. Develop an ability to work in a team by communicating computing ideas effectively in writing a technical report.

Contents

Unit 1 - Software Development Process: [8 Hrs]

Software Engineering basics, Software Crisis and Myths, Software Quality Attributes, Software Process and development, Software life cycle and Models, Analysis and comparison of various models, Iterative Enhancement Models, agile process.

Unit 2 - Requirement Engineering: [6 Hrs]

Requirements Engineering, requirement engineering process, SRS Document, IEEE Standards for SRS, Introduction to Analysis model.

Self Study: Case Study of SRS as per IEEE Standards**.**

Unit 3 - System Architecture and Design Overview: [8 Hrs]

Architecture 4+1 view, architecture styles, Design process, Data Flow Diagrams, Entity Relationship Diagrams, Refactoring of designs, quality concepts – modularity, cohesion coupling, Analysis model to Design Model transformation, Standardisation using UML.

Self Study: Entity Relationship Diagrams

Unit 4 – Project Management & Software Metrics: [6 Hrs]
Project Management Concepts - Planning, Scope, Project Scheduling, Work break-down structure, Milestones, Deliverables, Introduction to Software Metrics-Size-oriented metrics and function point metrics. Effort and cost estimation techniques - LOC-based and Function-point based measures - The COCOMO model.
Self Study: Risk Management

Unit 5 - Testing and Maintenance: [8 Hrs]
Validation and Verification activities, Testing Principles and strategies, Testing levels- Test Drivers and Test Stubs, Testing types- White Box & Black Box Testing, Software Maintenance, Types of Maintenance, Overview of Software Configuration Management.
Self Study: Software Configuration Management Tools

Unit 6 - Quality Assurance: [6 Hrs]
Quality Assurance, Quality Standards Models Overview - ISO, Process improvement Models: CMM & CMMI. Quality Control and Quality Assurance
Self Study: Quality Standards Models - TQM, Six-Sigma

TEXT BOOKS

- R. S. Pressman, "Software Engineering: A Practitioners Approach", McGraw Hill, 7th edition, 2010
- Rajib Mall, "Fundamentals of Software Engineering", PHI Publication, 3rd edition, 2009
- "G. Booch, J. Rumbaugh and I. Jacobson. The Unified Modeling Language User Guide, Addison Wesley, 1999.

Reference Books

- "Shari Pfleeger, "Software Engineering", 2nd Edition. Pearson"s Education, 2001. · "Ian Sommerville, "Software Engineering", 6th Edition, Addison-Wesley, 2000
- "Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa publication house
- "Fred Brooks, "Mythical Manmonths", www.cs.drexel.edu/~yfcai/CS4517/.
-

(PCC-16) Software Engineering (Lab)

Examination Scheme: **LAB**
Project Report: 10 Marks
ESE Viva: 30 Marks
Cont. Evaluation: 60 Marks (6 assignments-10 marks each)

A full-fledged working system in the form of mini project will be implemented following the task list given below. Students in groups of two will be working on mini

project. After consultation with the course instructor and finalisation of the topic following deliverables are expected under mini project.
Task List for the same is as
follows:

1. Carry out state of art survey, selecting appropriate domain, problem identification statement formulation based on research problems or real-world problems, industry-based problem etc.

2. Develop workflow graph and carry project estimation, calculation of efforts, project planning (schedule) using automated tools.

3. Gather requirements and Write the Software Requirement specification (SRS, IEEE specs) document for the project.

4. Design System architecture using different ERD, DFD, UML diagrams for the proposed system. (Use different open-source tools for design.)

5. Coding/Implementation using suitable Programming Languages/ Technology.

6. Develop Test cases. Propose solution for wrong results in test cases by focusing on regression testing.

7. Write the constraints, advantages and disadvantages of your project over existing system. Write the future scope of your project. Develop help manual for maintenance and usability.

Students will be required to submit a technical report written using LaTEX. The technical report will include description of the project/problem, design of the software, description of problems solved and solution design, result analysis of test cases and conclusions. Students will also be required to demonstrate and present their work in a viva-voce.

(PCC-09) Database Management Systems

Examination Scheme:
MSE: 30 Marks
ESE: 50 Marks
Mode of Exams: Offline/Online

Course Outcomes:
Students will be able to:
1. Identify and describe various components of DBMS.
2. Construct Entity-Relationship Model for given applications and demonstrate it as a relational model.

3. Design optimal (SQL) query statement.
4. Apply normalization to database design.
5. Improve efficiency in data retrieval using indexing and solve analytical problems on serializability, concurrency and recovery.

Contents

1. Unit 1: Introduction: Basic Concepts; Database system application, purpose of database systems, view of data, database languages; Database architecture: components of DBMS and overall structure of DBMS; Various types of databases. **[4 Hrs]**

2. Unit 2: E-R and Relational Model: Database design; E-R model: modeling, entity, attributes, relationships, constraints, components of E-R model; Relational model: basic concepts, attributes and domains, concept of integrity and referential constraints, schema diagram. **Self-Study : Network Model** **[8 Hrs]**

3. Unit 3: Relational Algebra and SQL: Relational algebra**:** fundamental relational algebra operations, additional relational algebra operations, extended relational algebra operations, null values, modification to database; SQL: basic structure and operations, aggregate functions, nested subqueries, complex queries. **Self-Study: Views** **[6 Hrs]**

4. Unit 4: Relational Database Design: Basic concept of normalization; Decomposition using functional dependencies (1 to BCNF and 3NF). **Self-Study: Basics of Multivalued dependencies.** **[6 Hrs]**

5. Unit 5: Indexing and Hashing: Basic concepts, Ordered Indices; Indices: concepts, B+ trees; Static and dynamic hashing. **Self-Study: B tree index file, Bit Map indices. [6 Hrs]**

Unit 6: Transactions and Concurrency control: Transaction: basic concepts, states, concurrent execution, serializability, recoverability, isolation; Concurrency control: timestamps and locking protocols, validation-based protocols, deadlock handling; Recovery. **Self-Study: M**ultiple granularity protocols, ARIES algorithm **[10 Hrs]**

Text Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database system concepts”, Fifth Edition, McGraw Hill International Edition, ISBN 978-0073523323.
- Raghu Ramkrishnan, Johannes Gehrke, “Database Management Systems”, Second Edition, McGraw Hill International Editions, ISBN 978-0072465631.

Reference Books:

- Rob Coronel, “Database systems : Design implementation and management”, Forth Edition, Thomson Learning Press, ISBN 978-1418835934.
- Ramez Elmasri and Shamkant B. Navathe, “Fundamental Database Systems”, Third Edition, Pearson Education, 2003, ISBN 978-0321204486.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Second Edition, Pearson, 2008, ISBN 978-0131873254.

(PCC-09) Database Management Systems Laboratory

Teaching Scheme: Lab	**Examination Scheme**: **LAB**
2 hours per week	Cont. Evaluation: 50 Marks
	External Oral: 50 marks

Course Outcomes:
Students will be able to:
1. Implement optimal (SQL) query for given statement.
2. Analyze requirements and Design E-R data model by applying different forms of constraints on data for the given application/problem in hand.
3. Apply Normalizations techniques to database.
4. Write high level program to connect database through appropriate API
5. Understand and use three tier architecture model.

List of Assignments:

Assignment 1: Basic SQL
Statement: Write the simple SQL Queries on the given schema.
Assignment 2a: SQL: Aggregates
Statement: Write the SQL queries using aggregates , grouping and ordering statements for given statements on given schema.
Assignment 2b: Nested Subqueries and SQL Updates
Statement: Write the SQL queries for given schema using Nested Subqueries and SQL Updates
Assignment 3: SQL DDL and updates
Statement: Write the DDL and DML statements for the given statements.
Assignment 4: Schema creation and constraints
Statement: Create the schema and constraints on the given relations using given statements.
Assignment 5 : Implement a small database application using appropriate front End and Back end. Select any real time problem for database implementation. (Draw the ER diagram for the selected problem in hand. Normalize the database up to the appropriate normal form.)
This list is a guideline. The instructor is expected to improve it continuously.

(PCC-10) Operating Systems

Teaching Scheme: Lab	**Examination Scheme**: **LAB**
3 hours per week	TA: 20 Marks
	MSE: 30 Marks
	ESE: 50 Marks
	Mode of Exams: Open Book, Offline/Online

Course Outcomes

Students will be able to:

1. Write programs to deal with processes, files, and hardware resources using appropriate system calls.
2. Illustrate the design issues, solutions and complexity of operating system by compiling, modifying an OS kernel, tracing the sequence of activities on processor, data structures of a file system, race conditions, locking mechanisms and storage techniques.
3. Correlate the computer architecture features with operating system design issues.
4. Make design choices for an operating system with given constraints
5. Discuss suitable changes to existing implementations of operating systems for new features.

Course contents

Introduction and Operating Systems structures: Operating system components, O.S. Services, System Calls, Virtual Machines, Special purpose operating systems, Open-source operating systems, Boot Procedure. Source code of boot loader.; System Services; Linkers and Loaders; Operating system structures; Calling convention and Executable file formats;
[6 Hrs]

Processes and CPU Scheduling: Interrupts and Interrupt handling, Process concept, interleaved I/O and CPU burst; Process states; Co-operating processes, Thread, Thread libraries, Multithreaded programming, Scheduling, Scheduling criterion, Scheduling algorithms, Multi processor scheduling, Real time scheduling, Source code of interrupt handling and scheduler and it's dependence on hardware features.
[6 Hrs]

Memory management: O.S. and hardware interaction, Swapping, Continuous memory management, paging, Segmentation, Virtual Memory Management, Demand Paging, Copy-on-write, Page replacement algorithms, Allocation of frames, Thrashing, Kernel memory management, SVR4 architecture, Unified buffer cache. Case study of x86 architecture and implementation of segmentation and paging in an OS kernel, code of fork(), exec(), sbrk(), etc.

[10 Hrs]

File Management and Storage Structures: File Organization, Concept of files and directories, System calls for file systems, Space allocation issues, Free space management, Data structures like inode and super block, Virtual file system and related object oriented concepts, Disk layout, Formatting, Recovery, NFS, Efficiency and performance, Distributed file systems, Disk Structure, Disk Scheduling, RAID. Case study of a file system like ext2.
[6 Hrs]

Process Synchronization: Critical section problem, Hardware support for mutual exclusion, Semaphores, Deadlock-principle, Deadlock detection, prevention and avoidance, Classical problems in concurrent programming: Producer-consumer, Reader-writer with and without bounded buffer. Design of locking primitives like spinlock, semaphore, read-write locks, recursive locks, etc. Source code of spin locks, semaphores, read-write locks.
[8 Hrs]

Inter process Communication: Pipes, Shared memory mechanism, Streams, Asynchronous communication, Signals. A shell program using system calls and pipes.
[4 Hrs]

Self Study topics:
Design aspects of Android operating system, File systems like ext3 and ext4; pthread implementation of threads; SVR4 memory management architecture; Network file systems; Synchronization problems in kernels like Solaris; IPC mechanisms in different kernels like Linux, MAC, etc.
[14 hrs]

Text Books

- Abranhan Silberschatz, Peter B Galvin, Greg Gagne; Operating System Concepts, Wiley India Students Edition, 9th Edition, ISBN: 978-81-265-2051-0
- Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau(University of Wisconsin-Madison), https://pages.cs.wisc.edu/~remzi/OSTEP/
- xv6, a simple, Unix-like teaching operating system, Russ Cox, Frans Kaashoek, Robert Morris; E-book

Reference Books

- Milan Milenkovic; Operating Systems; Tata McGraw Hill; Second Edition. ISBN: 0-07-044700-4
- Andrew S. Tanenbaum; Modern Operating Systems; Prentice Hall of India Publication; 3rd Edition. ISBN: 978-81-203-3904-0
- Maurice J. Bach; The Design of the Unix Opearating System; Prentice Hall of India; ISBN: 978-81-203-0516-8
- Uresh Vahalia; Unix Internals, The New Frontiers; Prentice Hall; ISBN: 0-13-101908-2

(PCC-10) Operating Systems Laboratory

Teaching Scheme: Lab
2 hours per week

Examination Scheme: **LAB**
Assignments: 40 Marks
Course Project: 30 marks
End Semester Exam: 30 marks

Course Outcomes

Students will be able to

1. Implement different features in code of an existing kernel.
2. Demonstrate ability to use existing system programs to inspecting operating system features.
3. Demonstrate the ability to implement system programs.

4. Demonstrate ability to create and solve race conditions using synchronization primitives.
5. Write programs using system calls and IPC.

Suggested List of Assignments

1. Install two operating systems in dual boot mode using virtual machines. One OS will run a GNU/Linux of your choice on it. The other virtual machine will run any non-Linux operating system.
2. Write a minimal version of a shell. The shell should be able to a) execute a program without the complete path name b) handle pipes c) handle redirection d) handle signals
3. Use debugfs tool to locate a file which was recently deleted on an ext2 file system.
4. Write a program, on the lines of debugfs, to browse an ext2 file system and given the complete name name of a file, print it's inode.
5. Download linux kernel source code, compile it and reboot your system with the newly compiled kernel. Add a dummy system call to the Linux kernel. Write a conformance test to test your system call.
6. Implement a list type. Write a code using pthreads for concurrent insertions to the list and demonstrate the problem of race. Then rewrite the program to show how race conditions can be solved by using proper synchronization primitives.
7. Write a program which results in a guaranteed deadlock among it's threads. Kill the threads using operating system's commands to solve the deadlock.
8. Write a program using pthreads to demonstrate the producer consumer problem. Implement appropriate synchronization. Show the different results with and without synchronization.
9. Write a program which acts as a chat application between two users on the same computer, using shared memory.
10. Implement the lseek() system call in xv6 kernel.
11. Rewrite the kernel memory manager in xv6 kernel, that is rewrite the kmalloc() and kfree() using another data structure.
12. **Suggested list of course projects:** Implement mkfs, fsck for ext2 file system; Implement a multithreading library using 1-1, many-many, many-one implementation on Linux; Any of the xv6 based projects: implement demand paging, implement ext2 filesystem, Implement signals, Implement shared memory, Implement Kernel threads, Implement mmap(), etc.

Suggested list of demonstrations by instructor.

1. Trace and explain completely the output of strace on running a "Hello World" C program.
2. Write a program to demonstrate the usage of signals – show how processes can wait for each other, kill each other, stop and continue each other.
3. Demonstrate the changing memory map of a process, by using the contents of the /proc file system, and creative use of malloc() function in the code of the process.
4. Demonstrate that not unmounting a file system results in loss of data. Recover possible data using fsck and restore file system consistency. Analyze the recovered data.

This list is a guideline. The instructor is expected to improve it continuously.

(OE-03) Fundamentals of Machine Learning

Teaching Scheme: Lab
2 hours per week

Examination Scheme: **LAB**
TA: 20 Marks
MSE: 30 Marks
ESE: 50 Marks

Course Outcomes

Students will be able to:

1.Understand the basic concepts, tools and techniques of Machine Learning.

2. Apply various data pre-processing techniques to simplify and speed up machine learning algorithms.

3. Analyze different ML models.

4. Implement ML algorithms to solve real life problems.

5. Evaluate the different ML models

Prerequisites: Relevant applied math and statistics: probability theory, probability distribution, Conditional probability, Bayesian probabilities, Data visualization

Course Contents

Unit 1: Introduction to Machine Learning: Introduction to Machine Learning, Comparison of Machine learning with traditional programming, ML vs AI vs Data Science.Types of learning: Supervised, Unsupervised, and semi-supervised, reinforcement learningtechniques, Important Elements of Machine Learning- Data formats.

[4 Hrs]

Unit 2: Data Preprocessing: Libraries: Pandas, NumPy, Scikit-Learn, Understanding Data: Dealing with missing values, Data formatting, Implementing Data Cleaning: Preprocessing of data: Transformation, Normalization and Scaling, Standardization, Importing and exporting data in Python , Data Splitting: Train-Test Split, Cross-Validation

[5 Hrs]

Unit 3: Supervised Learning : Regression Types of Regression Techniques: Linear regression, Multiple regression, Polynomial regression, Lasso regression, Ridge regression, Evaluation Metrics for Regression Models: MAE, MSE, RMSE, R square **[4 Hrs]**

Unit 4: Supervised Learning : Classification

Classification: Logistic Regression, K-nearest neighbour, Support vector machine., Random Forest, Binary-vs-Multiclass Classification, Evaluation Metrics and Score: Accuracy, Precision, Recall, Fscore **[5 Hrs]**

Unit 5: Unsupervised Learning: Introduction to Unsupervised Learning**,** Definition and Scope of Unsupervised Learning, Types of Unsupervised Learning: Clustering, Dimensionality Reduction, Association Rule Mining, Evaluation metrics and score: elbow method, extrinsic and intrinsic methods **[4 Hrs]**

Unit 6: Model Evaluation and Tuning: Bias-Variance Trade off, Overfitting and Underfitting, Cross-Validation Techniques, Hyperparameter Tuning: Grid Search, Random Search, Model Evaluation Metrics, Visualizing ROC, AUC, and Precision-Recall Curves using Matplotlib and Seaborn.

Unit 7 : Case Studies and Applications: Evaluation and Tuning of Regression Models (Business Forecasting),Evaluation and Tuning of Classification Models (Healthcare Diagnostics),Comparing Models using Evaluation Metric. **[6 Hrs]**

[Self Study Unit]

Introduction To Neural Networks: Artificial Neural Networks: Single Layer Neural Network, Multilayer Perceptron, Back Propagation Learning, Functional Link Artificial Neural Network, and Radial Basis Function Network, Activation functions, Introduction to Recurrent Neural Networks and Convolutional Neural Networks.

[10 Hrs]

Text Books

1. Bishop, Christopher M., and Nasser M. Nasrabadi, “Pattern recognition and machine learning”,Vol. 4.No. 4. New York: springer, 2006.
2. Ethem Alpaydin, “ Introduction to Machine Learning”, PHI 2nd Edition-2013

Reference Books

1. Tom Mitchell, “ Machine learning”, McGraw-Hill series in Computer Science, 1997
2. Shalev-Shwartz, Shai, and Shai Ben-David, “Understanding machine learning: From theory toalgorithms”, Cambridge university press, 2014.
3. Jiawei Han, Micheline Kamber, and Jian Pie, “Data Mining: Concepts and Techniques”, Elsevier Publishers Third Edition, ISBN: 9780123814791, 9780123814807
4. Hastie, Trevor, et al., “The elements of statistical learning: data mining, inference, and prediction”, Vol. 2. New York: springer, 2009.
5. McKinney, “Python for Data Analysis “,O' Reilly media, ISBN : 978-1-449-31979-3

5.(PEC-01) DE1- Remote Sensing

Teaching Scheme: Lab
2 hours per week

Examination Scheme: **LAB**
TA: 20 Marks
MSE: 30 Marks
ESE: 50 Marks

Course Outcomes

By the end of this course, students will be able to:

1. Understand the fundamental principles of remote sensing, including electromagnetic radiation, sensor systems, and data acquisition.
2. Analyze and interpret remote sensing data for various applications such as land cover classification, environmental monitoring, and disaster management.
3. Apply image processing techniques to enhance and classify remote sensing imagery.
4. Evaluate the advantages and limitations of different remote sensing platforms (satellite, aerial, UAV).
5. Demonstrate proficiency in using remote sensing software (e.g., QGIS, ENVI, Google Earth Engine) for data analysis.

Course Contents

Unit 1: Introduction to Remote Sensing

- Definition, history, and significance of remote sensing.
- Electromagnetic spectrum and its relevance to remote sensing.
- Interaction of EMR with atmosphere and Earth's surface (absorption, scattering, reflection).
  [4 Hrs]

Unit 2: Remote Sensing Platforms and Sensors

- Types of platforms: Satellites (Landsat, Sentinel, MODIS), Aerial, UAVs.
- Sensor characteristics: Spatial, spectral, temporal, and radiometric resolution.
- Passive vs. active remote sensing (Optical, Thermal, Microwave, LiDAR). **[5 Hrs]**

Unit 3: Image Acquisition and Preprocessing

- Data acquisition methods and formats (GeoTIFF, HDF, NetCDF).
- Radiometric and geometric corrections.
- Image enhancement techniques (histogram equalization, filtering). **[6 Hrs]**

Unit 4: Image Classification and Analysis

- Supervised vs. unsupervised classification.
- Feature extraction and object-based image analysis (OBIA).
- Accuracy assessment (confusion matrix, Kappa coefficient).
  [6Hrs]

Unit 5: Emerging Trends in Remote Sensing

- Hyperspectral and microwave remote sensing.
- Machine learning and AI in remote sensing.
- Cloud computing for remote sensing (Google Earth Engine). **[4 Hrs]**

Unit 6: Applications of Remote Sensing (5 Hrs)

- Land use/land cover mapping.
- Vegetation monitoring (NDVI, EVI).
- Disaster management (floods, wildfires, deforestation).

- Urban planning and climate change studies. **[3 Hrs]**

Text Books

- "Remote Sensing and Image Interpretation" – Lillesand, Kiefer, Chipman.
- "Introduction to Remote Sensing" – James B. Campbell.

Reference Books:

- "Remote Sensing of the Environment: An Earth Resource Perspective" – John R. Jensen.
- "Hyperspectral Remote Sensing: Principles and Applications" – Marcus Borengasser.

(PEC 01-) DE1 -Remote Sensing Laboratory

Teaching Scheme	Evaluation Scheme
Lab Hours: 2 Hrs. / Week	**Lab:** Continuous Assessment: 50, ESE Oral: 50 Marks

Course Outcomes

By the end of the lab course, students will be able to:

1. Demonstrate proficiency in using tools like QGIS, Google Earth Engine to process and analyze satellite/aerial imagery.
2. Perform preprocessing (radiometric & geometric corrections), enhancement (filtering, histogram adjustments), and classification (supervised/unsupervised) on remote sensing data.
3. Calculate and interpret vegetation (NDVI, EVI), water (NDWI), and urban indices to assess environmental changes.
4. Generate confusion matrices and compute accuracy metrics (Overall Accuracy, Kappa Coefficient) to validate remote sensing outputs.
5. Design and execute a real-world remote sensing application by integrating acquired skills in data processing, analysis, and visualization.

Suggested List of Assignments

1. Basics of Remote Sensing Software

- Install and explore QGIS/ Any other open-source GIS Tool.
- Load and visualize satellite imagery (Landsat/Sentinel).

2. Image Preprocessing

- Perform radiometric correction (DN to reflectance).
- Apply geometric correction (reprojection, resampling).

3. Image Enhancement Techniques

- Apply histogram equalization and contrast stretching.
- Perform edge detection and filtering.

4. Spectral Indices Calculation

- Compute NDVI, NDWI, and other vegetation/water indices.
- Analyze changes over time using multi-temporal imagery.

5. Image Classification

- Perform unsupervised classification (K-means, ISODATA).
- Conduct supervised classification (Maximum Likelihood, SVM).

6. Accuracy Assessment

- Generate a confusion matrix.
- Calculate overall accuracy and Kappa coefficient.

7. Mini Project

This is a suggested list. The instructor is expected to continuously update it.

(PEC-01) DE1- Advanced Data Structures

Teaching Scheme:	**Examination Scheme:**
Lectures: 2 Hrs/week	TA: 20 Marks
	Mid sem: 30 Marks
	End Sem Exam - 50 marks

Course Outcomes

Students will be able to:

1. Illustrate advanced data structures to solve complex problems in various domains.
2. Develop logic for graphical modelling of real time scenarios
3. Select the non-linear data structures for real world scenarios.
4. Implement the suitable data structure for any practical problem with given constraints.
5. Analyse the algorithmic solutions for resource requirements and optimization

Course contents

Introduction: Abstract data type, Algorithm; Asymptotic notations: time and space complexity, time analysis of recursive algorithms, amortized analysis; Recurrence relation: Master theorem, recursion tree, substitution method. Data Structure: Tree, BST, Graph
[4 Hrs]

Search Trees: Balanced Binary search trees-(AVL trees, red-black tree, splay trees); Multi-way search trees-(B-trees, 2-3 trees); specialized search trees-(Treaps, skip list); Multidimensional search trees-(K-D trees, segment trees).
[6 Hrs]

Heap: Overview, Leftist heap, skew Heap, Binomial Heap, Fibonacci Heap; Applications-Priority Queue, Graph algorithms, Huffman coding

[6 Hrs]

Data Structure for Strings: Introduction to string data structures, Tries, Compressed Tries, Suffix Trees, Suffix Arrays; Applications-Search Engines, Bioinformatics, Pattern Matching: KMP algorithm; Internet Packet Forwarding.

[6 Hrs]

Advanced Graph: Disjoint set union problem, Maximal flow problem: Ford Fulkerson Algorithm, Hamiltonian Path and circuit problem, Introduction to Hypergraphs, Applications-Social Network Analysis, A* for AI path finding), Computational Geometry: Geometric Data Structure, Plane Sweep Paradigm

[6 Hrs]

Text Books

- Peter Brass, "Advanced Data Structures", Cambridge University Press, ISBN- 978-0-521-88037-4
- Handbook of Data Structures and Applications; by Dinesh P. Mehta (Editor) , Sartaj Sahni (Editor) ; Chapman and Hall/CRC; ISBN-10: 1584884355;ISBN-13: 978-1584884354

Reference Books

- E. Horowitz, S. Sahni, S.Anderson-freed, "Fundamentals of Data Structures in C", Second Edition, University Press, ISBN 978-81-7371-605-8
- T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, 2nd edition, Prentice-Hall India, 2001.
- S. Sahni, Data Structures, Algorithms and Applications in C++,2nd edition, Universities Press,2005.
- Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd edition, Addison Wesley Educational Pub
- J. P. Tremblay and P. G. Sorenson, "An Introduction to Data Structures with applications", Second Edition, Tata McGraw Hill, 1981

(PEC-01) Advanced Data Structure Laboratory

Teaching Scheme:	**Examination Scheme:**
Laboratory: 2 Hrs/week	Assignments: 40 Marks
	Course Project: 30 Marks
	End Semester Exam: 30 Marks

Course Outcomes

Students will be able to

1. Illustrate the ADT/libraries, search trees and graph to design algorithms for a specific problem.
2. Choose most appropriate data structures and apply algorithms for graphical solutions of the problems.
3. Demonstrate, compare and implement algorithms for various operations in different implementations of variants of tree, graph, heap, etc.
4. Apply nonlinear data structures to solve real world complex problems.
5. Analyse the efficiency of most appropriate data structure for creating efficient solutions for engineering Design situations

Suggested List of Assignments

1. Write the following functions for a binary search tree implementation: Searches the maximum value in the tree, preorder traversal without using recursion, Search the string in the tree and returns a pointer to the node, print the binary tree so that it looks like a tree.
2. Implement red-black tree for banking system-Efficient transaction processing. Perform the operations such as insert, search, sort, update based on timestamp
3. Implement search tree which is suitable for E-commerce platform - a product search system in an e-commerce database
4. Start with an empty AVL tree and perform series of insertions like: December, January, April, March, July, August, October, February, November, May, June. Display the tree.
5. Develop C functions to insert and delete into/from a max heap under the assumption that a dynamically allocated array is used, the initial capacity of this array is 1, and array doubling is done whenever we are to insert into a max heap that is full.
6. Use Treap to implement a priority-based task scheduler where each task has a unique ID (key) and a priority. Higher-priority tasks should be completed first while maintaining the properties of a Binary Search Tree (BST).
7. Implement Hospital Emergency Queue to cater services to the patients as Insert, delete and find highest priority
8. Create Binomial Heap and Insert keys in Binomial Heap, Find Minimum key of Binomial Heap, unite two Binomial Heap: Take Binomial Heaps as parameter and unite it. Extract Minimum Key, Decrease Key, Delete Key
9. A large logistics company faces the following challenges: 1. Efficiently schedule delivery trucks based on priority and delivery deadlines. 2. Handle real-time changes increased delivery volumes. 3. Ensure efficient load balancing across multiple distribution hubs. 4. Allow dynamic merging of delivery queues when combining or splitting routes. Implement a suitable data structure.
10. Create a dictionary of English words. Insert all words from the dataset into the trie. Perform Search and auto-suggest operation.

Suggested list of course projects: E-commerce Platform, Memory Management, Dynamic Visualizations of various data structure, Shortest Pathfinding Using A* for Optimal Routing, etc.

Suggested list of demonstrations by instructor.

1. Write a program to demonstrate the balanced Binary search tree and various rotations after insertion or deletion of an element. Analyse the time and space complexity.
2. Write a program to demonstrate the usage of multi-way search trees – Implement insertion, deletion, searching operations.
3. Demonstrate the variants of heap and its operations.
4. Implement the tries data structure with insertion, deletion operation. Analyse the time and space complexity.

This list is a guideline. The instructor is expected to improve it continuously.

(PEC-01) DE1- Parallel Computing

Teaching Scheme
Lectures: 2 Hrs. / Week

Evaluation Scheme
Theory: Teacher's Assessment: 20 Marks, Mid Sem Exam/ Practical Assignments: 30 Marks, ESE: 50 Marks

Course Outcomes

Students will be able:

6. To understand different parallel architectures and models of computation.
7. To analyse the various classes of algorithms and gain knowledge about various parallel programming paradigms
8. To program shared memory multicore computer system using OpenMP.
9. To develop practical programming skills using MPI.
10. To apply knowledge of parallel programming to solve different problems.

Course Contents

Introduction to Parallel Computers: Need for parallel computing - SM-SIMD algorithms - Shared memory SIMD - Tree and mesh interconnection computers - Classifying MIMD Algorithms - parallel computational models such as PRAM - LMCC - Hypercube - Cube Connected Cycle - Butterfly - Perfect Shuffle Computers - Tree model - Pyramid model - Fully Connected model - PRAM - CREW - EREW models - simulation of one model from another one.

[5Hrs]

Introduction to Parallel Programming: Strategies - Mechanism - Performance theory - Parallel Programming Patterns: Nesting pattern - Parallel Control Pattern - Parallel Data Management - Map: Scaled Vector - Mandelbrot - Collative: Reduce - Fusing Map and Reduce

- Scan - Fusing Map and Scan - Data Recognition: Gather - Scatter - Pack - Stencil and Recurrence - Fork-Join - Pipeline. **[6Hrs]**

Threading Fundamental concepts: - Parallel programming models - Designing for threads - Scheduling - Threading and parallel programming constructs – Synchronization-Threading APIs. **[5Hrs]**

OpenMP Programming: OpenMP - Threading a loop - Thread overheads - Performance issues - Library functions - Solutions to parallel programming problems - Data races - deadlocks and live locks - non-blocking algorithms - Memory and cache related issues. **[6Hrs]**

MPI Programming: Distributed memory – Cluster - MPI Model - collective communication - data decomposition - communicators and topologies - point-to-point communication - MPI Library.
[6Hrs]

Self-Study: C-DAC's PARAM Utkarsh Supercomputer case study and programming models. Intel's TBB or any other point in context of recent findings.

Text Books

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition, 2017, Pearson.
- Michael J. Quinn, Parallel Computing: Theory and Practice, Second Edition, 2017, McGraw Hill Education.

Reference Books

- Herlihy, M., Shavit, N., Luchangco, V., & Spear, M., The art of multiprocessor programming, Second Edition, 2021, Morgan Kaufmann.
- Michael McCool, James Reinders, Arch Robison, Structured Parallel Programming: Patterns for Efficient Computation, 2012, Morgan Kaufmann.

CT (DE1) Parallel Computing Lab

Teaching Scheme	**Evaluation Scheme**
Lab Hours: 2 Hrs. / Week	**Lab:** Continuous Assessment: 50, ESE Oral: 50 Marks

Course Outcomes:

Students will be able to:

1. Understand basics of performance analysis of computing systems.

2. Demonstrate performance statistics using perf utility.
3. Implement shared memory programs using pthreads and OpenMP.
4. Implement message passing programs in distributed environment.
5. Demonstrate the different steps involved in building of a simple cluster.

List of Assignments:

1. Study of different benchmarks used to evaluate performance of different systems.
2. Performance statistics observation using perf utility.
3. Program to execute matrix multiplication using pthreads.
4. Program to execute matrix multiplication using OpenMP and comparison with pthread program.
5. Program to execute Pi computation and prefix sum using OpenMP.
6. Program to execute section, task and synchronization constructs of OpenMP.
7. Case Study of Cluster building steps - MPI Cluster setup and overview of different routines.
8. Program to implement point to point communication using MPI routines.
9. Program to implement collective communication using MPI routines.

(CT-DE1) Mobile and Ad-Hoc Networks

Teaching Scheme:	**Examination Scheme:**
Lectures: 2 Hrs/week	Mid Semester Exam: 30 marks Teacher's Assessment: 20 marks End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:
1. Describe primitive Ad-Hoc network, its specific characteristics and features exhibited by the networks.
2. Identify the different types of mobile Ad-Hoc networks and their challenges.
3. Differentiate the Media Access Control protocols for Ad-Hoc networks.
4. Identify the different transport layer functionalities and protocol and related security.
5. Trace the issues & challenges in providing QoS in wireless network and their classification.

Course Contents

Unit 1: Introduction:
Fundamentals of wireless communication technology, Characteristics of the wireless channel, multiple access techniques, IEEE 802 networking standard, IEEE 802.11 standard, IEEE 802.16 Standard [5 Hrs]

Unit 2: Cellular Wireless Networks and Wireless Internet:
The cellular concept, Cellular architecture, Generations of cellular systems, Wireless in local loop, Wireless ATM, Wireless internet, Mobile IP, TCP in wireless domain, WAP, Optimizing web over wireless. [5 Hrs]

Unit 3: Introduction To Ad-Hoc Networks:
Characteristics, applications, Issues in Ad Hoc wireless networks medium access protocols: design issues, Goals and classification, Contention based protocols- with reservation, Scheduling algorithms, Protocols using directional antennas. [4 Hrs]

Unit 4: Routing Protocols:
Design issues, Goals and classification. Proactive Vs reactive routing, Unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithm, Energy aware routing algorithm, Hierarchical routing, QoS aware routing. [5 Hrs]

Unit 5: Other Features:
Issues in designing transport layer Ad-Hoc networks, TCP over wireless Ad-Hoc networks, Security issues in Ad-hoc networks, Secure routing protocols, Energy management in Ad-Hoc networks, Battery management, Transmission power management and system power management. [5 Hrs]

Unit 6 : Wireless Sensor Networks:
Architecture, Data dissemination, Data gathering, MAC protocols for sensor networks, Location discovery, Quality of a sensor network. [4 Hrs]
Self Study: Quality of service in Ad Hoc wireless networks, Energy management in Ad Hoc wireless networks, Recent advances in wireless networks.

Text Books

· C. Siva Ram Murthy and B. S. Manoj, Ad hoc Wireless Networks Architectures and protocols, 2nd edition, Pearson Education. 2007
· Stefano Basagni, Marco Conti, Silvia Giordano and Ivan sSojmenovic, Mobile Ad-hoc Networking, Wiley-IEEE Press, 2004.

Reference Book

· F.Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004
· Mohammad Ilyas, The Handbook of Ad-hoc Wireless Networks, CRC press, 2002.
· Research, “Wireless Commun, and Mobile Comp. Special Issue on Mobile Ad-hoc Networking Research, Trends and Applications, Vol. 2, no. 5, 2002, pp. 483 – 502. 5.
· A survey of integrating IP mobility protocols and Mobile Ad-hoc networks, Fekri M. bduljalil and Shrikant K. Bodhe, IEEE communication Survey and tutorials, no: 12007.

(CT-DE1) Mobile and Ad-Hoc Networks Laboratory

List of Assignments

Using any simulation tool demonstrates the following:
1. Create a sample wireless topology.
2. Create a mobile Ad-hoc network.
3. Implement the Ad-hoc On-demand Distance Vector protocol.
4. Implement the Transmission Control Protocol.
5. Implement the User Datagram Protocol.
6. Implement the Low Energy Adaptive Hierarchy protocol.
7. Implement the Power Efficient Gathering in Sensor Information System.
8. Implement the Sensor Protocol for Information via Negotiation (SPIN).

ADDITIONAL EXPERIMENTS:
1. Implement a Power Efficient and Delay Aware MAC
2. Implement a Predictive Wake-up MAC protocol.
3. Implement a Proactive and Reactive based MAC protocol.
4. Implement a Scheduling based protocol for WSNs.

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT-DE1) System Administration

Teaching Scheme:	**Examination Scheme:**
Lectures : 2 Hrs/week	Mid Semester Exam: 30 marks Teacher's Assessment: 20 marks End Semester Exam: 50 Marks Mode of Exams: Offline/Online
Lab: 2 hrs/week	Continuous Internal Evaluation: 50 Marks External Oral: 50 Marks

Course Outcomes:

Students will be able to:
1. Install various Linux/Unix distribution servers and desktop systems on commodity hardware, and carry out basic administration tasks of the user, network, process, storage management.
2. Setup a LAN-based laboratory using DHCP.
3. Install and configure a web server, database server, DNS, NFS, NIS, LDAP based system, secure a Desktop and a server system completely using existing tools.
4. Learn how to manage the Linux file system and the storage of data
5. Provide security administration for Linux

Contents

Unit 1:Basic System Administration: Partitioning, Installation of multiple operating systems on desktops, Various operating system services: Cron, CPU utilization, User management, Backup, Log management, Boot loader, Process management, File system namespace, Kernel upgrades. [5 Hrs]

Unit 2: Network Administration: Configuration of network hardware, Linux router, Managing routes, DHCP - Dynamic Host Configuration Protocol, DNS-Domain Name Server, NFS - Network File System, NIS - Network Information Service, Email Setup-Send Mail, Issues with mail services, POP and IMAP Server
[5 Hrs]

Unit 3 :File system Administration: Formatting, Partitioning, Managing file systems, Defragmentation, Quotas, Journaling file system, Logical volume management, Disk layouts, File system check, Archiving and compressing files, SAN, NAS, Case Studies: ext2, ext4, ntfs, samba, cifs, lvm, fat32. **[5 Hrs]**

Unit 4: Security Administration: Methods of attack, Network security tools- Nmap, Snort, Nessus, Wireshark/tcpdump, Firewall, IPtables, NAT, IP filtering, Setting up linux firewall, Mandatory access control (MAC), SELinux, Cryptographic security tools Authentication mechanisms, LDAP, Proxy servers. **[5 Hrs]**

Unit 5: Server Administration: Apache webserver configuration, Database servers: MySQL and PostgreSQL **[3 Hrs]**

Unit 6: Devices Administration: Device resources, udev: Device files, Installing and configuring printers, Managing printers via the web interface, Scanners, PCI devices, LAN cards, Device troubleshooting, Plug and Play devices. **[3 Hrs]**

Self Study: LDAP, Proxy servers. **[4 Hrs]**

Text books:

- Evi Nemeth, Garth Snyder, Ben Whaley, Trent R. Hein, "UNIX and Linux System Administration Handbook", Pearson Education, Fourth edition, ISBN-13: 978-8131761779
- Arnold Robbins, Nelson H. F. Beebe, "Classic Shell Scripting", Shroff/O'ReillyFirst edition, ISBN-13: 978-8173668463
- Wale Soyinka, "Linux Administration: A Beginner's Guide", McGraw-Hill Osborne Media Publication Sixth Edition, ISBN-13: 978-0071767583
- Olaf Kirch & Terry Dawson, "Linux Network Administrator's Guide", O′ Reilly 2nd Edition June 2000,ISBN-10: 1565924002,ISBN-13: 978-1565924000

- W.Preston, “Using SANs and NAS”,O′ Reilly; First Edition, February 2002,ISBN-10: 0596001533,ISBN-13: 978-0596001537

Suggested List of Assignments in the Laboratory:

1. Set a desktop with following software configuration options: triple boot with Windows, Ubuntu and Fedora operating systems; Grub timeout set to 5 seconds with default Fedora Linux; Each step of boot process secured with passwords; Following software installed in each OS: office, internet browser, c compiler, terminal, ssh server, telnet server, web server, database server.

2. Set up a LAN based computer laboratory with 5 computers - using DHCP first and then using static IPs; one of the machines should work as DHCP server; set up LDAP+NFS based authentication for managing single identity on all computers; setup quotas on NFS systems.

3. Demonstrate use of three-tier architecture using LAMP and WAMP suite using same code base.

4. Setup a disk management system using LVM such that all options of LVM are demonstrated.

5. Write a program to browse an ext4 file system and locate data of a deleted file.

6. Setup a proxy server and firewall with set of policies to block video content in the network.

7. Setup apache web server to serve 3 websites, with 3 domain names at a time from a single machine; demonstrate the use of at least 5 configuration options of apache.

8. Setup an email server and demonstrate use of at least 3 email clients to send email using it.

9. Setup SE Linux.

This list is a guideline. The instructor is expected to improve it continuously

(CT-DE1) Data Visualization

Teaching Scheme	**Evaluation Scheme**
Lectures: 3Hrs/ Week	Theory: Mid Sem: 30 Marks, ESE: 50 Marks, Assessment: 20 Marks

Course Outcomes:

Students will be able to:

CO1: Understand the fundamentals of data visualization and its significance
CO2: Apply data preprocessing techniques and aesthetics principles to prepare data for visualization
CO3: Design appropriate visualizations to represent distributions, amounts, and proportions
CO4: Analyse complex datasets using advanced visualization techniques
CO5: Create and interpret dashboards using Power BI for business intelligence

Contents

Introduction to Data Visualization **[6hrs]**
Importance of Data Visualization in Data Science & Business; Data for Graphics; Design Principles; Value for Visualization; Categorical, Time Series, and Statistical Data Graphics; Introduction to Visualization Tools

Data Pre-processing and Aesthetics
[6hrs]
Data Cleaning and Aggregation; Advanced Visuals: Pairplots, Violin Plots, Heatmaps; Interactive Visualizations; Graphical Perception Theory; Grammar of Graphics (Layering, Aesthetics, Geometries); Aesthetics and types of data; Scales map data values onto aesthetics;

Visualization Design **[6hrs]**
Visualizing amounts: Bar plots; Grouped and stacked bars; Dot plots and heat maps; Visualizing distributions: Histograms and density plots, Visualizing a single distribution, Visualizing multiple distributions at the same time, Empirical cumulative distribution functions and q-q plots, Empirical cumulative distribution functions, Highly skewed distributions, Quantile–quantile plots

Advanced Visualization Design **[6 hrs]**
Grouped and Visualizing many distributions at once: Visualizing distributions along the vertical axis, Visualizing distributions along the horizontal axis; Visualizing proportions: A case for pie charts, side-by-side bars, stacked bars and stacked densities, Visualizing proportions separately as parts of the total; Visualizing nested proportions; Visualizing associations among two or more quantitative variables: Scatter plots, Correlograms, Dimension reduction, Paired data.
Visualizing time series, trends and geospatial data **[6 hrs]**
Time series: Individual time series, Multiple time series and dose–response curves, Time series of two or more response variables; Trends: Smoothing, Showing trends with a defined functional form, Detrending and time-series decomposition; Geospatial data: Projections, Layers, Choropleth mapping, Cartograms.

Power BI for Business Intelligence & Dashboarding **[6 hrs]**
Power BI Components: Desktop, Service, Gateway; Data Loading from Excel, SQL, Web APIs. Power Query for Data Shaping; Visuals: Cards, KPIs, Gauges, Slicers, Drill-down; DAX for Custom Measures and Calculations; Power BI Embedded and Row-Level Security

Self study:
Case Studies Projects: Supply Chain & Logistics Dashboard. Dataset: Shipment and delivery logs; Tools: Power BI or Dash; Focus: Route mapping, delivery status KPIs, filtering by region

Textbooks:
1. Claus O. Wilke - Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures 1st Edition, ISBN-10- 1492031089
2. Kieran Healy-"Data Visualization: A Practical Introduction", Princeton University Press,ISBN 978-0-691-18161-5, ISBN (pbk.) 978-0-691-18162-2, 2018

Reference Books:
1. Edward R. Tufte – *The Visual Display of Quantitative Information*
2. Alberto Cairo – *The Functional Art: An Introduction to Information Graphics and Visualization*

DE1: Data Visualization Laboratory

Teaching Scheme	**Evaluation Scheme**
Lectures: 2Hrs/ Week	Continuous Assessment: 50 ESE Oral: 50 Marks

Course Outcomes:

Students will be able to:

CO1: Analyse the given dataset to understand basics of dataset in hand
CO2: Apply data preprocessing techniques to prepare data for visualization
CO3: Demonstrate various visualizations techniques to represent distributions, amounts, and proportions in the given dataset.
CO4: Demonstrate various visualizations techniques applied to grouped data.
CO5: Demonstrate used of advanced tools to apply the visualization techniques to a real life problem.

Lab sessions will consist of practically implementing concepts studied in theory lectures.

1. Load a dataset (e.g., Iris, Titanic, Sales). Analyse various features present in dataset and create various charts like Bar Chart (categorical data), Line Chart (time series) and Box plot analysis for the given dataset.
2. Take any sample dataset clean and aggregate it, create various multivariate analysis plots to gain insight in the dataset. Compare different chart types for the same data.
3. Demonstrate complex visual designs using various plots to present proportions and relationships among various features in dataset.
4. Take any time series dataset and display data points over time allowing for the visualization of trends, cycles, variations and prepare a report for the same.
5. Mini Project for getting Business Insights Dashboard Using Visualization Techniques can be implemented using various tools like Power BI.

Above list serves as a guideline and is intended to be continuously refined by the instructor.

MDM-Artificial Intelligence
(Machine Learning and Deep Learning Mastery)

Teaching Scheme:
Lectures: 3Hrs/week
Lab: 2Hrs/week

Examination Scheme:
MID-SEM Exam – 30Marks
TA: 20 Marks
ESE: 50 Marks
Lab CIE: 100 Marks
Mode of Exams: Online

Course Outcomes

Students will able to:

1. Understand Supervised Learning techniques such as Regression, Classification (Logistic Regression, SVM).
2. Apply Unsupervised Learning methods like Clustering (K-Means, DBSCAN).
3. Utilize Decision Trees and Random Forests for problem-solving.
4. Build neural networks from scratch, understanding the forward and backward passes.
5. Handle large datasets through preprocessing, data wrangling, and feature selection.
6. Execute hands-on projects using real-life datasets for ML/DL applications.
7. Convert machine learning and deep learning projects into research papers, focusing on writing impactful papers.

Course Contents

Machine Learning Techniques: Supervised Learning: Regression, Classification (Logistic Regression, SVM),Unsupervised Learning: Clustering (K-Means, DBSCAN), Decision Trees and Random Forests

[12 hrs]

Deep Learning Techniques: Build a neural network from scratch: Forward pass, Build a neural network from scratch: Backward pass, Assemble and train the entire neural network, Testing the entire neural network

[14 hrs]

Big Data Techniques: Handling Large Datasets: Data Preprocessing, Data Wrangling, Feature Selection

[6 hrs]

Machine Learning - Deep Learning Lab: Hands-on Projects: Real-life datasets for ML/DL applications

[5 hrs]

Projects to Research Papers: Converting Projects into Research Papers: Writing impactful papers.

[5 hrs]

Reference Material

Sr. No.	Title	Creators	Publisher
1	Introduction to NLP and CV	Dr. Raj Dandekar, Dr. Rajat Dandekar, Dr. Sreedath Panat	Vizuara AI Labs org
2	Natural Language Processing with Python	Steven Bird	O'Reilly
3	Jay Alammar blogs	Jay Alammar	Github, Youtube

•

6.(MD M-02) – Quantum Computing

Teaching Scheme:

Lectures: 3 Hrs/week

Examination Scheme:

Mid Semester Exam: 30 marks

Teacher's Assessment: 20 marks

End Semester Exam: 50 Marks

Course Outcomes:

Students should be able to,

1. Analyze simple states of superposition and the effect of doing the measurement in different basis states.
2. Design and analyze quantum circuits with single and two-qubit gates.
3. Equip students with the linear algebra background required for this course.
4. Analyze quantum circuits with superposition and entanglement.
5. Analyze quantum algorithms and complexity.
6. Understand QAOA and error correction scheme.

Detailed Contents

Unit 1: Course Introduction: Comparison about classical physics and Quantum Physics, Quantum Mechanics and computations. [2 hrs]

Unit 2: Superposition and Single Qubit: Superposition, Polarization of light, Single qubit notation, Measurement of Qubit, BB84 Quantum Key Distribution, Bloch Sphere Notations, Q-sphere, Difference between Bloch and Q-sphere. [7hrs]

Unit 3: Quantum Gates and Circuits : Model of computation (movement on Bloch Sphere) X, Y, Z, H gates CNOT, Toffoli, Fredkin SWAP gate Simple circuits Quantum Adder Reversible circuits, Basics of quantum circuit design. [7hrs]

Unit 4: Basics of Linear Algebra :
Dirac Notation Vectors Complex Conjugate & Norm Analysing Pauli gates Analysing Cascade of gates Analysing Two-qubit gates Tensor Product (example), Introduction of Three qubit gates. [7hrs]

Unit 5: **Superposition and Entanglement**
Introduction of quantum mechanics, Quantum state formation, Entangled States, Testing for Entangled States, Bell Pair and Bell States EPR Paradox & Bell Theorem Conditional Instructions Quantum Teleportation Superdense Coding. [5hrs]

Unit 6: Quantum Algorithms
Deutsch Deutsch-Jozsa Bernstein, Vazirani Grover, Simon's Algorithm, Period Finding, Shor's Algorithm, QFT (Basics). [5 hrs]

Unit 7: QAOA and Error Correction
Introduction of Quantum Approximate Optimization Algorithm (QAOA), Maxcut problem, Overview of QAOA Optimizations for QAOA, what is Quantum errors, analyse the simple error correction scheme, Unique challenges in QEC Shor's bit-flip code Shor's phase-flip code Shor 9-qubit code Steane code Concatenation code [6 hrs]

Text Book:

1. P. Wittek, Quantum Computing. Education, 2007
2. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University press, 2007.
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020

Reference Books

1. "Quantum Computing: A Gentle Introduction" by Eleanor Rieffel and Wolfgang Polak, MIT press (2014)
2. Quantum Computation and Quantum Information, M. A. Nielsen &I. Chuang, Cambridge University Press (2013)

MD – M02 Quantum Computing Lab

Course Objectives: Learner are able to understand fundamental difference between classical and quantum computing by learning various concepts such as quantum bits, superposition, entanglement.

Course Outcomes:

Students should be able to,

1. Survey and installation of available quantum platform.
2. Install Qiskit for various task execution such as qubit visualization.
3. Connect real quantum hardware and simulators.
4. Design and analyze multi-qubit circuits on quantum hardware.
5. Design and implement various quantum algorithms

List of Experiments

Practical 1: Introduction of Available Quantum Platforms and Programming languages Contents—

IBM Quantum Platform, Quantum Cloud Connects, Local machine connections, AWS quantum, Google quantum, Quantum Technologies.

Case study on various quantum technologies.

Practical 2: Installation of Qiskit Platform Contents—

Qiskit installation via local classical computer, Installation of Miniconda/Anaconda v.0, Install IBM quantum platform through qiskit.

Learn the steps of qiskit installation and IBM quantum platform to access quantum hardware through local computer desktop.

Practical 3: Qubit Visualization/Bloch Sphere Single Qubit State - Contents—Basic Rotations on One Qubit and Measurements on the Bloch Sphere

Design following single qubit states.

1. **Bit Flip**
2. $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

3. $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Practical 4: Quantum Circuits Using Multi-Qubit Gates
Contents—
Design and Model of computation (movement on Bloch Sphere) X, Y, Z, H gates CNOT, Toffoli, Fredkin SWAP gate Simple circuits Quantum Adder Reversible circuits, Basics of quantum circuit design.
Construct the Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Practical 5: Write a function that builds a quantum circuit on 3 qubits and creates the GHZ-like state:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle):$$

Practical 6: Execution of task on Real Quantum hardware
Contents—
Map the problem to a quantum-native format, Optimize the circuits and operators., Execute using a quantum primitive function, Analyze the results.

Python code for connecting various quantum hardware, Check availability of hardware, Search for less busy hardware.

Practical 7: Execute simple "hello world" Circuit on /quantum computer.

Practical 8: Implementation of Quantum Algorithms
Contents—
Deutsch Deutsch-Jozsa Bernstein, Vazirani Grover, Simon's Algorithm, Period Finding, Shor's Algorithm, QFT (Basics)
Design and implement Oracles and the Deutsch-Jozsa algorithm

Practical 9: Design quantum teleportation.

Practical 10: Design and analyse circuit for Grover search

TextBook:

1. P. Wittek, Quantum Computing. Education, 2007
2. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University press, 2007.
3. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020

Reference Books

1. "Quantum Computing: A Gentle Introduction" by Eleanor Rieffel and Wolfgang Polak, MIT press (2014)

2. Quantum Computation and Quantum Information, M. A. Nielsen &I. Chuang, Cambridge University Press (2013).

(PCC-11) Computer Networks

Teaching Scheme:
Lectures: 3 Hrs/week
Lab: 2Hrs/week

Examination Scheme:
MID-SEM Exam – 30 marks
TA: 20 Marks
ESE: 50 marks
Mode of Exams: Offline/Online
Continuous Internal Evaluation: 50 Marks
External Oral: 50 Marks

Course Outcomes

Students will able to:

1. Examine the challenges associated with each layer of the TCP/IP stack and critically assess various application layer protocols utilized on the Internet.
2. Gain a comprehensive understanding of the transport layer's roles, focusing on TCP and UDP protocols, mechanisms for reliable data transfer, and strategies for congestion and flow control.
3. Develop network designs incorporating subnetting as per given specifications, and understand the key network layer protocols employed in the Internet.
4. Understand link layer protocols and core concepts of wireless networking and Ethernet switching, and trace the complete end-to-end journey of packets across the Internet.
5. Evaluate the challenges specific to wireless networks and the protocols designed to address them.

Course Contents

Introduction to Computer Networks and Internet: Basic concepts, Machine Learning methods: Supervised, Unsupervised, Semi-supervised, Inductive, Reinforcement Learning.

[4 hrs]

Application Layer:: Basics of Socket Programming, Transport Layer Programming Interface (TCP, UDP), Protocols: HTTP (Overview, Persistent and Non-Persistent, Message Format, Cookies, Cachess), SMTP (Overview, Message Formats), IMAP, POP, DNS; FTP; Telnet, SSH; Peer-to-Peer Applications.

[7 hrs]

Transport Layer: Relationship Between Transport and Network Layer, TCP and UDP; Multiplexing and Demultiplexing; Principles of Reliable Data Transfer; Go-Back-N and Selective Repeat; TCP: Segment Structure, Round Trip Time Estimation, Reliable Data Transfer, State Transitions, Flow Control, Congestion Control, Fairness; UDP: Segment Structure

[4 hrs]

Network Layer, Subnets: Concept of IP Address, Netmask, Subnet; CIDR; Design of a

LAN and WAN; Subnetting Problems.

[4 hrs]

Network Layer, Routers, IPv4, IPv6: Functions of a Router: Forwarding and Routing; Inside of a Router: Port Processing, Switching; IPV4: Datagram Format, Fragmentation; Network Address Translation; IPv6 Introduction; Multicasting. [4 hrs]

Network Layer, Routing algorithms: Link State, Distance Vector Routing; OSPF, BGP, RIP; Routing Policies

[8 hrs]

Link Layer: Review of fundamentals of link layer protocols; Ethernet Switches, LANs, Link Layer Switches, Complete tracking of traversal of a packet over internet between two applications

[6 hrs]

Wireless Networks: Wireless Links and Network Characteristics, Bit Error Rate, SNR; Problems of Wireless Links (Interference, Signal Strength fading, Multipath Propagation, Hidden Terminal problem), 802.11 (Architecture, MAC protocol, Frame Structure).

[6 hrs]

Self-Study:

BitTorrent Protocol; Conte Distribution Networks, Queueing: Causes, Delays; Transport layer implementation Overview in OS Kernels; VLANs; Routing Protocols for Ad Hoc Wireless Networks.

[6 hrs]

Text Books

- J.F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet”, Pearson, ISBN-13: 9780201976991
- Behrouz A. Forouzan, Firouz Mosharraf, Computer Networks: A Top-Down Approach, Tata McGraw-Hill Education Pvt. Ltd, ISBN 10: 1259001563 / ISBN 13: 9781259001567

Reference Books

- Larry Peterson Bruce Davie, Computer Networks A Systems Approach, Elsevier, ISBN: 9780123850591
- Behrouz A. Forouzan, Firouz Mosharraf, Computer Networks: A Top-Down Approach, Tata McGraw-Hill Education Pvt. Ltd, ISBN 10: 1259001563 / ISBN 13: 9781259001567
- Kevin R. Fall, W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Pearson, ISBN-13: 978-0321336316/ISBN-10: 0321336313
- Behrouz Forouzan, Data Communications and Networking, Tata McGraw-Hill, ISBN 13: 978-0073250328/ISBN-10: 0073250325

- William Stallings, “Data and computer Communication”, Pearson Education, ISBN-81 297-0206-1
- Alberto Leon Garcia and Indra Widjaja, “Communication Networks, Fundamental Concepts, and Key Architectures”, Tata McGraw-Hill, ISBN-10: 007246352X
- Peter Loshin, IPv6 Theory, Protocol, and Practice, Elsevier, ISBN: 9781558608108

List of Practical Assignments:

1. Implement a TCP-IP client-server application. The server accepts a string as a request and sends the capitalized version back.
2. Configure a network on your computer using DHCP and static IP. Learn the following networking commands and configuration files: ifconfig, ping, host, traceroute, Telnet, netstat, nslookup, ssh, scp, wget, /etc/hosts, /etc/network/interfaces
3. Install and demonstrate the following networking applications: web server (apache2 or nginx), ftp server, ssh server, email client
4. Access the website http://go.com (or any website on HTTP, not HTTPS). Use Wireshark to inspect the flow of packets for a GET request using a browser. Identify all packets for this request. Capture the data from all packets and construct the HTML page. Trace the TCP sequence numbers and verify that they match the expected behaviour.
5. Send an email using SMTP over the server at new.toad.com (an open SMTP server).
6. Implement a packet sniffer to capture packets of a specified link layer, network layer, transport layer, or application layer protocol.
7. Write a program to simulate either Go-Back-N or Selective Repeat protocols.
8. Implement a given subnet using the ns3 simulator and demonstrate the flow of packets.
9. Critically analyse the COEP network design.
10. Course Mini project: Implement any one of these: a web server (HTTP Protocol), email client (IMAP and POP protocols), a BitTorrent client, DNS server and client (like nslookup), etc, simulations of transport layer/network layer protocols; Adding features/bug fixes in apache2/nginx like projects

This list is a guideline. The instructor is expected to improve it continuously.

(PCC-08) Design and Analysis of Algorithms

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hrs/week	TA: 20 Marks
Lab: 2Hrs/week	MSE/ Practical Assignments: 30 Marks

ESE: 50 Marks
Mode of Exams: Off Line/Online

Course Outcomes

Students will able to:

1. Understand the fundamental principles of algorithm design and analysis, including time and space complexity.
2. Apply asymptotic notation to evaluate and compare the performance of algorithms
3. Explain standard algorithm design techniques and analyze their performance.
4. Analyze the correctness and efficiency of algorithms, including worst-case, average-case, and best-case analysis.
5. Formalize computational problems, apply reductions, and evaluate problem complexity within established classes.

Course Contents

Introduction: Objectives of time and space analysis of algorithms; Order notations (O, Ω, θ notations) ; (Best average and worst case) time complexity of algorithms such as bubble sort, selection sort, insertion sort, heap sort etc.; Time complexity of recursive programs using recurrence relations, Amortized analysis.

[6 hrs]

Design Techniques-I: Divide and Conquer: Quicksort, Mergesort, Strassen‟s matrix multiplication; Greedy Algorithms: Knapsack problem, Job sequencing with deadlines, Optimal merge patterns, Single source shortest paths.

[7 hrs]

Design Techniques-II: Dynamic Programming: All pairs shortest paths, 0-1 Knapsack, Traveling salesperson problem, Chained matrix multiplication, Longest common subsequence, Bellman-Ford algorithm, Subset sum.

[8 hrs]

Design Techniques-III: Backtracking: N-queens problem, Hamiltonian cycles, Graph Coloring; Branch-and-Bound: 0/1 Knapsack problem, Traveling salesperson problem.

[7 hrs]

Selected Algorithms from various areas: String Matching: The naïve string-matching algorithm, The Robin-Karp algorithm, The Knuth- Morris-Pratt algorithm; Network Flow Algorithms: Ford-Fulkerson algorithm, Push-relabel algorithm. [6 hrs]

Complexity Theory: Polynomial time, NP-hard and NP-complete problems, proving NP completeness using reduction technique (e.g. SAT, Independent Set, 3VC, etc.) [6 hrs]

Self-Study: [6 hrs]

Sorting in linear time, Elementary graph algorithms, Minimum spanning tree, Number -Theoretic algorithms: GCD algorithm, Chinese remainder theorem, Primality testing.

Text Books

- Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press, 2nd edition (2008) , ISBN-13: 978- 8173716126
- Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, “Introduction to Algorithms”, PHI, 3rd edition, ISBN-13: 978-8120340077

Reference Books

- Gilles Brassard and Paul Bratley, “Fundamentals of Algorithmics”, PHI, ISBN-13: 978- 8120311312
- Jon Kleinberg and Éva Tardos, “Algorithm Design”, Pearson Education India, ISBN13: 978-9332518643

(PCC-13) Artificial Intelligence

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Teachers Assessment – 20 Marks
Mid Term Exam/Assignment – 30 marks
End Sem Exam - 50 marks

Course Outcomes:

Students will be able to:

1. Compare AI with human intelligence and traditional information processing and discuss its strength and limitations
2. Apply the basic principles, models and algorithms of AI to recognize, model and solve problems
3. Demonstrate knowledge of basics of the theory and practice of Artificial Intelligence
4. Design and carry out an empirical evaluation of different algorithms on a problem formalisation, and state the conclusions that the evaluation supports.
5. Apply AI techniques and problem solving strategies to common AI applications.

Content:

Introduction: What is AI, History, AI problems, Production Systems, Problem characteristics, Intelligent Agents, Agent Architecture, AI Application (E-Commerce, & Medicine), AI Representation, Properties of internal representation, Future scope of AI , Issues in design of search algorithms.

[6 Hrs]

Heuristic search techniques: Heuristic search, Hill Climbing, Best first search, mean and end analysis, Constraint Satisfaction, A* and AO* Algorithm**,** Knowledge Representation**:** Basic concepts, Knowledge representation Paradigms, Structured representation of knowledge, ISA hierarchy, Frame notation, Resolution

[6 Hrs]

Learning & Planning: What is Learning, Types of Learning (Rote, Direct instruction Analogy, Induction, Deduction), Planning: Block world, strips, Implementation using goal stack, Non linear planning with goal stacks, Hierarchical planning, Least commitment strategy.

[8 Hrs]

Advanced AI Topics: Game playing: Min-max search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search, Natural Language Processing (NLP): Introduction, Steps in NLP, Syntactic Processing, Semantic analysis, Discourse & Pragmatic Processing. Perception and Action: Perception, Action, Robot Architecture, Machine Learning: Definition of learning systems. Goals and applications of machine learning. Aspects of developing a learning system: training data, concept representation, function approximation.

[8 Hrs]

Neural Networks and Deep Learning: Neurons and biological motivation. Linear threshold units. Perceptrons: representational limitation and gradient descent training. Multilayer networks and backpropagation, Hidden layers and constructing intermediate, distributed representations, Overfitting, learning network structure, Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data

[8 Hrs]

AI Ethics, Future Trends, Applications and Case Studies: Bias and fairness in AI, Ethical considerations in AI deployment, Future trends: Explainable AI, AI in healthcare, autonomous systems, AI in robotics, computer vision, and speech recognition, Case studies: Self-driving cars, recommendation systems, fraud detection

[6 Hrs]

Text Books:

- Elaine Rich and Kerin Knight, Artificial Intelligence, 3rd Edition, McGraw Hill. ISBN13: 9780070087705
- Eugene, Charniak, Drew Mcdermott, Introduction to artificial intelligence, Addison-Wesley. ISBN 0-07-052263-4.

Reference Books:

- Stuart Russell and Peter Norvig, Artificial Intelligence : A Modern Approach, Prentice Hall, 3rd Edition. ISBN 0-13-103805-2.
- Tom Mitchell, Machine Learning, McGraw Hill. *ISBN*-10 : 1259096955
- Herbert A. Simon, The Sciences of the Artificial, MIT Press, 3rd Edition, 1998. *ISBN*: 9780262190510.
- George F Luger, Artificial Intelligence : Structures and Strategies for Complex Problem Solving, Pearson Edu., 4th Edition. *ISBN*-13: 978-0-321-54589-3.
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016. ISBN: 0262035618.

- Reinforcement Learning: An Introduction – Richard Sutton, Andrew Barto

(CT-) Artificial Intelligence Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/week

Examination Scheme:
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

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

1. Develop an understanding what is involved in learning models from data.
2. Implement a wide variety of learning algorithms.
3. Apply principles and algorithms to evaluate models generated from data.
4. Apply the algorithms to a real-world problem.
5. Implement a mini project in the field of AI applications.

Suggested List of Assignments:

1. Implement A* algorithm .
2. Implement AO* algorithm .
3. Implementation of other Searching algorithm.
4. Implementation of Min/MAX search procedure for game Playing .
5. Implementation of variants of Min/ Max search procedure.
6. Implementation of mini Project using the concepts studied in the AI course.

This list is a guideline. The instructor is free to assign new assignments.
This list is a guideline. The instructor is expected to improve it continuously.

(MDM-03) – Quantum Computing
(Quantum Algorithms and Applications)

Teaching Scheme:
Lectures: 3 Hrs/week
Lab: 2Hrs/week

Examination Scheme:
Internal Evaluation: 10 Marks
MSE: 30 Marks
ESE: 60 Marks

Pre-requisite: Quantum Computing, Machine learning, Cryptography, Material Science

Course Objectives: Graduate will be able to understand the core quantum algorithms and their application across disciplines.

Course Outcomes

Students will able to:

1. Demonstrate the need of core quantum algorithms.
2. Apply quantum algorithm to analyse various Machine Learning subroutines, algorithms.
3. Understand the pros and cons of Quantum Cryptography
4. Integrate quantum machine learning to investigate new approach in material science discipline.
5. Compare the classical complexities in various discipline such as ML, Cryptography and Material Science.

Course Contents

Course Introduction: This course covers quantum machine learning, Cryptography, Material Science, which accelerates and improves machine learning performed on the "classical" computers we use on a daily basis by utilizing the capabilities of quantum computing and quantum physics.

[2 hrs]

Introduction of Quantum Algorithms: Quantum Computing basics, Algorithm Basics, Quantum Algorithms: Deutsch-Jozsa, Simon, Grover, Shor, and their cryptanalytic implications. Implication of Grover's and Simon's algorithms towards classical symmetric key cryptosystems, Implication of Shor's algorithm towards factorization and Discrete Logarithm based classical public key cryptosystems.

[5 hrs]

Classical ML vs. Quantum Machine Learning: Introductory ideas about Turing machine, computational complexity. Basics of classical machine learning, Machine Learning terminologies, Introduction to different strands of quantum machine learning. Introduces feature vector, data encoding, some general property of Hadamard operation, measuring output.

[8 hrs]

Quantum Encoding, Classification and Clustering: Algorithm for amplitude encoding of data in superposition, quantum random access memory (QRAM), Quantum variational circuits, quantum parameterized circuits with various gates, ansatz, Feature maps, Quantum encoding techniques, Quantum Kernels. Building Quantum Classifiers, uploading data in quantum states, quantum feature map design, quantum classification, quantum clustering, preprocessing models, training classical model, training quantum model, handling sampling noise, selection of appropriate feature map, QSVM, QK-mean

[5 hrs]

Variational Classifiers to Linear Classifiers in QML: Introduces the classical concepts of linear classifier, the two formulations of its optimisation function, along with the widely used Kernel trick. Extends the idea of the classical kernel trick to a quantum version, first explaining the concept of quantum feature maps and applying them in introducing a model called Variational Quantum Classifier (VQC). VQC is indeed a linear classifier, kernel trick use in a VQC; quantum kernel estimator (QKE). [4 hrs]

Introduction to Quantum Cryptography: QKD (Quantum Key Distribution), BB84, Ekert, Semi-Quantum QKD protocols and their variations, Issues of Device Independence, Other cryptologic issues, such as Quantum secret sharing and multiparty computation, Introductory topics in Post-Quantum Cryptography.

[7 hrs]

Quantum Material Science: Discovery of new materials, Chemical compounds with specific properties, such as high thermal conductivity or superconductivity, QML for Chemical Compounds analysis, Case study on Computational complexities in prediction with classical computing methods in material science discipline. [8 hrs]

Course Conclusion [1 hr]

Text Books

- Quantum Supervised Machine Learning (Quantum Science and Technology), Schuld, Maria, Petruccione Francesco
- Book on Quantum Cryptography by Vidick and Wehner
- Quantum Mechanics for Material Science, by Gianluca Stefanucci, Springer Link: Quantum Mechanics for Material Science: An Introduction | SpringerLink

Reference Books

- P. Wittek, Quantum Machine Learning. Education, 2007
- Quantum Computation and Quantum Information, Michael Nielsen and Isaac Chuang, (Cambridge Univ Press, 2010)
- Machine Learning with Quantum Computers, Maria Schuld, Francesco Petruccione, (Springer, 2021)`

Implementation of Quantum Algorithm using Simulation Laboratory

Lab Scheme:
Lab: 2Hrs/week

Examination Scheme:
Internal Evaluation: 25 Marks
Practical & Oral: 25 Marks

Pre-requisite: Quantum Algorithms and QML, Database management

Course Objectives: Learners will be able to design and implement practical solution for various real-world complex problem using quantum mechanism.

Course Outcomes:
Students should be able to,

1. Design and analyse performance of variational quantum circuits.

2. Apply classical machine learning models on real dataset.
3. Design and implement quantum feature map for classical data encoding.
4. Training quantum model of ML for classification and clustering.
5. Analyse performance of quantum cryptography algorithms,
6. Demonstrate quantum advantage in material science discipline

List of Practical Assignments:

1. **Course Introduction:** This course helps learners to find classically challenging problems to find quantum solution for it using simulators.
2. **Introduction to Variational Algorithms and Circuits:** Variational quantum algorithms, in particular Variational Quantum Eigen solvers, how to create and work with parameterized circuits and quadratic programs in Qiskit, how to solve optimization problems using QAOA

   Practical 1: Construct parameterized quantum circuits and measure the performance.

   Practical 2: Construct parameterized quantum circuit and assign the values and perform Pauli rotations on parameters.
3. **Training a Classical Model on a Real Dataset:** Select dataset, Upload dataset in qiskit platform, Exploratory data analysis as preprocessing, Training a Classical Machine Learning Model,

   Practical 3: Download dataset of your choice and perform exploratory data analysis on it. (Example datasets: Iris, Brest cancer, Digits.
4. **Data Encoding and Quantum Feature map:** Quantum parameterized circuits with various gates, ansatz, Feature maps, Quantum encoding techniques such as zfeaturemap, zzfeaturemap, paulifeatureMap, Understand the quantum advantages on data using featuremaps with various conditions.

   Practical 4: Design zfeaturemap on given dataset features and display quantum circuit for that.

   Practical 5: Data Encoding Encode the data point x= $(-0.1,0.2)$ using the `ZZFeatureMap` with 4 repetitions and default data mapping function.

   Practical 6: Obtain quantum circuits using zzfeaturemap on given dataset for linear, circular and block entanglement.
5. **Training a Quantum Models, Classification and Clustering:** Quantum variational circuits, Building Quantum Classifiers, use objective functions against iteration, Quantum kernels and estimation, QSVM, Pegasos QSVM, QK-mean.

   Practical 7: Set Up the Quantum Kernel and Quantum Kernel Trainer, Train the Quantum Kernel, Fit and Test the Model, Visualize the Kernel Training Process. Obtain QSVM results.
6. **Case study on Quantum Cryptography:** Introduction of QKD (Quantum Key Distribution), BB84.

Practical 8: Case study on quantum key distribution. QKD and bb84 protocol simulation using qiskit

7. **Case study on challenges in material science:** Introduction for various classically challenging tasks in material science discipline.

   Practical 9: Simulate given chemical component using qiskit.

   Practical 10: Develop a problem statement based on the difficulties in material science engineering after consulting with subject-matter experts.

(MDM-03) -Artificial Intelligence
(Natural Language Processing (NLP) and Computer Vision (CV) Mastery, Capstone Project)

Teaching Scheme:
Lectures: 3Hrs/week
Lab: 2Hrs/week

Examination Scheme:
MID-SEM Exam – 30Marks
TA: 20 Marks
ESE: 50 Marks
Lab CIE: 100 Marks
Mode of Exams: Online

Course Outcomes

Students will able to:

8. Understand basic Natural Language Processing (NLP) techniques, including text preprocessing, tokenization, and parsing.
9. Apply various NLP techniques like N-grams, TF-IDF, and Word Embeddings (Word2Vec, GloVe).
10. Explore advanced language models such as RNNs, LSTMs, and Transformers for NLP applications.
11. Understand fundamental Computer Vision (CV) techniques, including image preprocessing and feature extraction.
12. Apply advanced CV techniques like Convolutional Neural Networks (CNN), Transfer Learning, and Object Detection.
13. Explore cutting-edge topics in CV like Generative Adversarial Networks (GANs) and Vision Transformers.
14. Implement NLP and CV models using TensorFlow and Keras in hands-on lab projects.
15. Work on an industrial capstone project involving real-world NLP or CV applications.

Course Contents

Natural Language Processing (NLP) Techniques: Introduction to NLP: Text Preprocessing, Tokenization,and Parsing, NLP Techniques: N-grams, TF-IDF, Word Embeddings (Word2Vec, GloVe),Language Models: RNN, LSTM, Transformer.

[18 hrs]

Computer Vision (CV) Techniques: Introduction to Computer Vision: Image Preprocessing,Feature Extraction, CV Techniques: CNN, Transfer Learning, Object Detection, Advanced Topics: GANs for Image Generation, Vision,Transformers.

[18 hrs]

NLP & CV Lab: Hands-on Lab: Implementing NLP and CV Models using TensorFlow, Keras [6 hrs]

Capstone Project: Industrial Capstone Project: Implementing a Real-world NLP or CV Project

[6 hrs]

Reference Material

Sr. No.	Title	Creators	Publisher
1	Introduction to NLP and CV	Dr. Raj Dandekar, Dr. Rajat Dandekar, Dr. Sreedath Panat	Vizuara AI Labs org
2	Natural Language Processing with Python	Steven Bird	O'Reilly
3	Jay Alammar blogs	Jay Alammar	Github, Youtube