Walchand Institute of Technology, Solapur
(An Autonomous Institute)

Affiliated to
Punyashlok Ahilyadevi Holkar Solapur University,
Solapur

Choice Based Credit System (CBCS)

Structure and Syllabus
for
T. Y. B.Tech. Information Technology
W.E.F. 2025-26

Dr. Mrs. M.A. Nirgude
Dean Academics

Department of

Information Technology

Vision

- To be a frontier in Information Technology, to produce globally competent engineers with an aptitude for leadership and research, who will be instrumental in continuous socio-economic development.

Mission

- M1: To impart quality education in Information Technology in accordance with the needs of the society through blended mode.
- M2: To inculcate critical thinking and creativity for identifying various issues and to provide sustainable solutions by becoming a lifelong learner.
- M3: To enhance career opportunities through academia-industry interaction and research, while embodying professional ethics.

Department of
Information Technology

Program Educational Objectives (PEOs)

- **PEO 1:** Graduates will exhibit strong fundamental knowledge and skills in the field of Information Technology to pursue successful professional careers, higher studies and research.
- **PEO 2:** Graduates will exhibit capabilities to understand and resolve the various issues through their problem-solving skills.
- **PEO 3:** Graduates will be sensitive to ethical, societal and environmental issues while serving at their professional work and society.

Knowledge and Attitude Profile (WK)

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

Program Outcomes (POs)	
PO 1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO 2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO 3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO 4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO 5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO 6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO 7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO 8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO 9	**Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning difference

PO 10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO 11	**Life-long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Program Specific Outcomes (PSOs)

- **PSO 1:** Graduates will be able to apply fundamentals of mathematics, algorithms and computational systems to Information Technology.
- **PSO 2:** Graduates will be able to provide a solution to the problem in the areas of Networking, Database management, System Software, Web Technology,
- **PSO 3:** Graduate will be able to design and develop IT solution for societal problem/s while encouraging usage of Free and Open-Source Software (FOSS).

Department of
Information Technology

Legends Used

L	Lecture Hours / week
T	Tutorial Hours / week
P	Practical Hours / week
FA	Formative Assessment
SA	Summative Assessment
ESE	End Semester Examination
ISE	In Semester Evaluation
ICA	Internal Continuous Assessment
POE	Practical and Oral Exam
OE	Oral Exam
MOOC	Massive Open Online Course
HSS	Humanity and Social Science
NPTEL	National Programme on Technology Enhanced Learning
F.Y.	First Year
S.Y.	Second Year
T.Y.	Third Year
B. Tech.	Bachelor of Technology

Department of Information Technology

Course Code Format

2	3	I	T	U/P	2	C	C	1	T/L
Year of Syllabus revision		Program Code		U-Under Graduate P-Post Graduate	Semester No./ Year 1/2/3/…8	Course Type		Course Serial No 1-9	T-Theory, L-Lab session A-Tutorial P-Programming / Design

Program Code	
IT	Information Technology
Course Type	
BS	Basic Science
ES	Engineering Science
HU	Humanities & Social Science
MC	Mandatory Course
CC	Program Core Compulsory Course
SN*	Self-Learning (**N*** *indicates the serial number of electives offered in the respective category)*
EN*	Program Core Elective Course (**N*** *indicates the serial number of electives offered in the respective category)*
SK	Skill-Based Course
SM	Seminar
MP	Mini project
PR	Project
IN	Internship
ON*	Open Elective (**N*** *indicates the serial number of electives offered in the respective category)*

MD	Multidisciplinary Minor
EM	Entrepreneurship/Economics/Management Courses
FP	Community Engagement Project / Field Project
AE	Ability Enhancement Course
VE	Value Education Course
IK	Indian Knowledge System
VS	Vocational & Skill Enhancement Course
RM	Research Methodology
HN	Honors’ Degree Course
HR	Honors Research

Sample Course Code	
23ITU5CC1T	System Software

Department of Information Technology

B. Tech. Semester V

Course Code	Name of Course	Engagement Hours			Credits	SA		FA		Total
		L	**T**	**P**		**Theory**	**OE/ POE**	**ISE**	**ICA**	
23ITU5CC1T	System Software	2	-	-	2	60	-	40	-	100
23ITU5CC2T	Database Engineering	2	-	-	2	60	-	40	-	100
23ITU5CC3T	Design and Analysis of Algorithm	2	-	-	2	60	-	40	-	100
23ITU5EN*4T	Program Elective I	3	-	-	3	60	-	40	-	100
23ITU5IK5T	Indian Knowledge System II – Vedic Mathematics	2		-	2			50		50
23##U5MD6T	Multidisciplinary Minor III	3		-	3	60		40		100
23MCU5OE7T	Open Elective III (MOOC)				2	50				50
	Subtotal	**14**	**-**	**-**	**16**	**350**		**250**	**-**	**600**
Laboratory Courses										
23ITU5CC1L	System Software	-	-	2	1	-	-	-	25	25
23ITU5CC2L	Database Engineering	-	-	2	1	-	25	-	25	50
23ITU5CC3L	Design and Analysis of Algorithm	-	-	2	1	-	25	-	25	50
23ITU5EN*4L	Programme Elective - I	-	-	2	1	-	-	-	25	25
23ITU5CC8P	Advanced Java	2	-	2	3	-	25	25	25	75
	Subtotal	**2**	**-**	**10**	**7**	**-**	**75**	**25**	**125**	**225**
	Grand Total	**16**	**-**	**10**	**23**	**350**	**75**	**275**	**125**	**825**

Department of Information Technology

B. Tech. Semester VI

Course Code	Name of Course	Engagement Hours			Credits	SA		FA		Total
		L	T	P		Theory	OE/ POE	ISE	ICA	
23ITU6CC1T	Artificial Intelligence	3	-	-	3	60	-	40	-	100
23ITU6CC2T	Internet of Things	2	-	-	2	60	-	40	-	100
23ITU6CC3T	Software Engineering	2	-	-	2	-	-	25	-	25
23ITU6EN*4T	Programme Elective -II	3	-	-	3	60	-	40	-	100
23ITU6EN*4A	Programme Elective –II (Tutorial)	-	1	-	1	-	-	-	25	25
23ITU6EN*5T	Programme Elective -III	3	-	-	3	60	-	40	-	100
23ITU6EN*5A	Programme Elective -III(Tutorial)	-	1	-	1	-	-	-	25	25
23##U6MD6T	Multidisciplinary Minor - IV	2	-	2	3	60	-	40	25	125
	Subtotal	**15**	**2**	**2**	**18**	**300**	**-**	**225**	**75**	**600**
Laboratory Courses										
23ITU6CC1L	Artificial Intelligence	-	-	2	1	-	-	-	25	25
23ITU6CC2L	Internet of Things	-	-	2	1	-	-	-	25	25
23ITU6CC3L	Software Engineering	-	-	2	1	-	25	-	50	75
23ITU6VS7P	Web UI/UX Technology	1	-	2	2		50	-	25	75
	Subtotal	**1**	**-**	**8**	**5**	**-**	**75**	**-**	**125**	**200**
	Grand Total	**16**	**2**	**10**	**23**	**300**	**75**	**225**	**200**	**800**

Note:

- N* indicates the serial number of electives offered in the respective category
- ## indicates program code of offering Program

- **List of Program Elective I**

List of Program Elective I (Semester –V)	
Course Code	**Course Title**
23ITU5E14T	Machine Learning
23ITU5E24T	Cloud Computing
23ITU5E34T	High Performance Computing

- **List of Program Elective II**

List of Program Elective II (Semester –VI)	
Course Code	**Course Title**
23ITU6E14T	Reinforcement Learning
23ITU6E24T	Network Security
23ITU6E34T	Data Science

- **List of Program Elective III**

List of Program Elective III (Semester –VI)	
Course Code	**Course Title**
23ITU6E15T	Natural Language Processing
23ITU6E25T	Block-chain Technologies
23ITU6E35T	Tools for Data Analytics

- **For 23MCU5OE7T: Open Elective III (MOOC) in Semester V:**
  1. Students are required to enroll in one of the courses of a minimum duration of 8 weeks offered on the SWAYAM/NPTEL platform. The list of courses will be finalized and released by Board of Studies each year.
  2. List of MOOCs will be provided by the department depending on the availability of the courses in that semester under NPTEL / Swayam or other recognized MOOC Platforms as per suggestions by the BoS.
  3. Students may enroll for the course in Semester IV or V. They must complete all assignments and appear for the certification examination conducted by SWAYAM/NPTEL.
  4. Students must pass the examination by the end of Semester V. The marks earned by the student in final assessment of this MOOC will be appropriately scaled and transferred to **Open Elective III (MOOC)** in Semester V.

- **For Courses offered during Semester VIII for A.Y. 2026-27**

Sr. No	*Course Code*	*Option I*	*Option -II*
1	23ITU8CC1T	MOOC Certification*	‘‘Management Information Systems’course offered by the department during Semester VIII in Self Learning / Online mode and successfully passing ESE for the same.
2	23ITU8EN*2T	MOOC Certification*	‘Program Elective Course VI’course offered by the department during Semester VIII in Self Learning / Online mode and successfully passing ESE for the same. Program Elective Course VI options: Recommendation Systems Business Network Security Business Data Analytics and Finance Management

* Student has to complete this MOOC in any semester of – V, VI, VII or VIII. In semester VIII, the marks earned by the student in final assessment of this MOOC will be appropriately scaled and transferred for 23ITU8CC1T and / or 23ITU8E32T along with the required credits.

The list of approved MOOCs will be finalized and released by Board of Studies each year.

- **Multidisciplinary Minor (MDM) Courses**

Sr. No	**MDM Program**	**MDM I (Sem III)**	**MDM II (Sem IV)**	**MDM III (Sem V)**	**MDM IV (Sem VI)**	**MDM V (Sem VII)**
1	Mechanical and Automation Engineering	Manufacturing Processes and Mechanisms	Machine Drawing and 3D Modeling	Automotive Engineering and Robotics	Additive Manufacturing	Thermal Engineering
2	Civil Engineering	Smart Buildings	Geoinformatics	Environmental Impact Assessment	Infrastructural Systems	Disaster Preparedness and planning
3	Electronics and Telecommunication Engineering	Fundamentals of Electronic Circuits	Electronics Design and Prototyping	Introduction to Embedded Systems	Fundamentals of Communication Techniques	Enclosure and Communication Design for IoT
4	Electronics and Computer Engineering	Analog Electronics	Digital Electronics	Microcontrollers and Peripherals	Advanced Controllers and Interfacing	Electronic System Design

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5CC1T: System Software

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	3	**ICA**	25 Marks

Introduction:

This course introduces different language processors and operating system. It deals with methods used to Analyse, synthesize, design and develop prototypes of language processors. It also covers the operating systems details like Process Management, deadlock, Memory Management and Unix System commands.

Course Prerequisite:

Discrete Mathematical Structures and Theory of Computation.

Course Objectives:

1. Explore the fundamental concepts and architecture of system software, focusing on operating systems.
2. Analyse process management, memory management, and file systems within operating systems.
3. Find the design and implementation aspects of assemblers, loaders, linkers, and macro processors.
4. Apply knowledge of system software components to real-world scenarios and system-level programming.

Course Outcomes:

After completing the course, students will be able to

1. Demonstrate the structure and functionalities of operating systems and system software components.
2. Illustrate process scheduling, synchronization, and memory management techniques.
3. Analyse the roles of assemblers, loaders, linkers, and macro processors in program execution.
4. Develop and debug system-level programs using appropriate tools and techniques.

Unit – I	Introduction to System Software and Operating Systems	5 Hours

Introduction system software and software tools, language processing activities, Fundamentals of language processing, fundamentals of language specification, Language processor development Tools: Editors, Debug Monitors, Programming Environments, and User Interfaces.

Overview of operating systems: functions, types, and structures, System calls and operating's system services.

Unit – II	**Process Management**	**5 Hours**
Process concepts and states, CPU scheduling algorithms: FCFS, SJF, Round Robin, Priority Scheduling, Process synchronization: critical section problem, semaphores, monitors, Deadlock: prevention, avoidance, detection, and recovery.		
Unit – III	**Memory Management**	**5 Hours**
Memory allocation techniques: contiguous, paging, segmentation, Virtual memory concepts: demand paging, page replacement algorithms, Thrashing and memory management in modern operating systems.		
Unit – IV	**File Systems**	**5 Hours**
File system architecture: file types, access methods, directory structures, File allocation methods and free space management, I/O systems: buffering, caching, spooling, and device drivers.		
Unit – V	**Assemblers and Loaders**	**6 Hours**
Design and implementation of two-pass and single-pass assemblers, Loader schemes: absolute, relocating, and dynamic linking, Role of linkers and loaders in program execution.		
Unit – VI	**Macro Processors and Compilers**	**4 Hours**
Macro definitions and expansions, design of macro processors. Use and overview of interpreters, Introduction of scanning and parsing, overview of compiler phases: lexical analysis, syntax analysis, semantic analysis.		
Internal Continuous Assessment (ICA): ICA consists of minimum 8 practical based on curriculum. Recommended practical’s		

Text Books

1. System Software: An Introduction to Systems Programming Leland L. Beck Pearson Education.
2. Operating System Concepts Abraham Silberschatz, Peter B. Galvin, Greg

Reference Books

1. Modern Operating Systems Andrew S. Tanenbaum Pearson Education
2. Operating Systems: Internals and Design Principles William Stallings Pearson Education
3. System Programming and Operating Systems D.M.Dhamdhere Second Edition (TMGH)
4. System Programming J. J. Donovan Mc-Graw Hill Publications

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5CC2T: Database Engineering

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours /week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	3	**ICA**	25 Marks
		POE	25 Marks

Introduction:

This course provides a comprehensive understanding of database systems, focusing on relational database design, SQL and data management principles. Students will learn about database architectures, E-R modelling, normalization, indexing and hashing to optimize performance. The syllabus also covers transaction management, ensuring data consistency and reliability. By the end of the course, students will have the skills needed to design, query and manage databases efficiently.

Course Prerequisite:

A basic understanding of data structures, algorithms and programming is recommended.

Course Objectives:

1. To understand the basics of database design, structure, implementation and applications.
2. To develop the logical design of the database using data modelling concepts such as entity relationship diagrams.
3. To study and use Structured Query Language to query, update and manage a database.
4. To apply normalization techniques to normalize the database.
5. To familiarize the students with the fundamentals of database transaction processing and learn techniques for concurrency control along with different indexing techniques.

Course Outcomes:

After completing the course, students will be able to

1. Apply the principles of database system.
2. Design database using E-R modelling and apply normalization techniques on a given scenario.
3. Formulate SQL queries to perform CRUD operations.
4. Apply appropriate indexing technique to optimize the performance of the database.
5. Demonstrate transaction processing techniques.

Unit – I	**Introduction**	**3 Hours**

Database System Applications, Purpose of Database Systems, View of data, Database Architectures, Database users and administrators, Introduction to NoSQL databases.

Unit – II	**Database Design and E-R Model**	**5 Hours**

Overview of design process, ER Model, Constraints, removing redundant attributes in entity sets, E-R diagrams, Reduction to relational schema, E-R design issues and Extended E-R features. Relational Model: Structure of relational databases, Database schema, keys, Schema diagrams.

Unit – III	**SQL and Advanced SQL**	**7 Hours**
SQL: Overview, SQL data definition, Basic structure of SQL Queries, Additional basic operations, Set operations, NULL values, Aggregate functions, Nested sub queries, Modification of the databases, Join operations, Views, Transactions, Integrity constraints, SQL data types and schemas.		
Unit – IV	**Relational Database Design**	**6 Hours**
Features of good Relational Designs, Atomic Domains, First Normal Form, Decomposition using Functional dependencies, Second Normal Form, BCNF, Third Normal Form, Functional-dependency theory.		
Unit – V	**Indexing and Hashing**	**4 Hours**
Basic Concepts, Ordered Indices, B+ Tree Index Files, B Tree Index Files, Multiple Key Access, Static Hashing, Dynamic Hashing, Bitmap indices.		
Unit – VI	**Transactions**	**5 Hours**
Transaction concepts, simple transaction Model, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity.		

ICA consists of minimum 8 practical assignments based on curriculum.
Recommended assignments:
Consider a real-life database application for the following assignments.

1. Design an E-R diagram and create a corresponding data dictionary.
2. Execute SQL DDL commands to create and update the schema.
3. Execute SQL DML commands with different clauses and aggregate functions.
4. Execute SQL queries using set operations, nested subqueries and joins.
5. Execute SQL queries to create Views and to set Integrity constraints.
6. Identify set of functional dependencies, attribute closures, Prime & non-prime attributes, Super keys, Candidate key for every relation of the database application selected in Assignment No.1.
7. Apply normalization techniques to the created database.
8. Write a program for database connectivity.
9. Write a program to apply a B+ tree index to the database.
10. Write a program to apply dynamic hashing to the database.
11. Write a program to simulate concurrency control protocol.

Text Books

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan McGraw Hill International Edition Sixth Edition.

Reference Books

1. Fundamentals of Database Systems Ramez El Masri, S. B. Navathe Pearson Education Fifth edition.
2. Database Management Systems Ramkrishnan Gehreke Tata McGraw Hill Third Edition.
3. Principles of Database Systems by J. D. Ullman Galgotia Publications
4. SQL The Complete Reference, 3rd Edition by James R Groff, PaulN. Weinberg and Andy Oppel Database system concepts by PeterRob, Carlos Coronel Cengage Learning ninth edition.
5. SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management, by Andreas Meier, Michael Kaufmann, Springer Vieweg; 1st ed. 2019 edition

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5CC3T: Design and Analysis of Algorithm

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	3	**ICA**	25 Marks
		POE	25 Marks

Introduction:

This course introduces key algorithmic strategies, performance analysis, and modern applications, enabling students to compare, select, and implement efficient algorithms for various problems.

Course Prerequisite:

- Basic programming knowledge
- Understanding of data structures

Course Objectives:

1. To introduce fundamental concepts of algorithms.
2. To study various design algorithm design paradigms.
3. To analyse algorithms in terms of time and space complexity using asymptotic notation.
4. To use various algorithm design paradigm& obtain the solution for a given problem.

Course Outcomes:

After completing the course, students will be able to

1. Derive time and space complexity of a given algorithm
2. Obtain a solution for a given problem by applying appropriate algorithm design paradigm.
3. Design a solution for a given problems using suitable technique.
4. Determine the optimal solution for a given problem

Unit – I	**Introduction to Algorithms**	**8 Hours**

Algorithm Specification: Pseudo code Conventions, Recursive Algorithm, Performance Analysis: Space Complexity, Time Complexity, Calculating worst case, best case and average case complexities, Asymptotic Notations, Performance Measurement

Unit – II	**Divide and conquer**	**7 Hours**

The general method, Binary search, Finding the maximum and Minimum, Quicksort, Selection Sort, Merge Sort

Unit – III	**Greedy Algorithms**	**8 Hours**

The general method, Knapsack Problem, Job Sequencing with deadlines, Minimum cost spanning trees – Prim ‘s and Kruskal ‘s Algorithms, Optimal storage on tapes, Optimal merge patterns, Single source shortest paths.

Unit – IV	**Dynamic Programming**	**8 Hours**
The general method, Multistage graphs, All pair shortest paths, Optimal binary search trees, 0/1Knap sack, Reliability design, The Traveling Sales person problem. Flow shop scheduling.		
Unit – V	**Backtracking**	**8 Hours**
The general method, 8-queen problem, sum of subsets, Knapsack Problem, Hamilton Cycle, and Graph Coloring.		

ICA shall consist of a minimum of ten practical assignment problems.

1. Utilize an algorithm design paradigm to offer a solution to the problem using C, C++, Python, Java, or any other programming language in which the student is proficient.
2. Effectively assess performance of provided solutions w.r.t programming language ‘s runtime implementation.
3. Compare and choose an appropriate algorithm design paradigm on time and space complexity

Text Books

1. Fundamentals of Computer Algorithms, Horowitz, Sahni & Rajasekaran Galgotia Publications
2. Fundamental of Algorithm, Gilles Brassard, Paul Bratley Pearson Publication
3. Introduction to Algorithms, Thomas Cormen Pearson Publication

Reference Books

1. Introduction to Design and Analysis of Algorithm, Goodman McGrawhill
2. Design and analysis of algorithms, Aho, Hopfcraft and Ullman Addison wesley
3. Design & Analysis of Algorithms, Sharma, Khanna Publishing House, N.Delhi
4. Design& Analysis of Algorithms, S. Sridhar, Oxford

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5E14T: Programme Elective I - Machine Learning

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

This course provides a foundational understanding of machine learning models as well as demonstrates how these models can solve the real time problems.

Course Prerequisite:

Basics of Programming, Probability, Statistics

Course Objectives:

1. To introduce various types of machine learning algorithms.
2. To perform data pre-processing operations.
3. To introduce methods to evaluate and tune machine learning models.

Course Outcomes:

After completing the course, students will be able to

1. Perform various data pre-processing tasks.
2. Apply various supervised machine learning algorithms to solve real world problem.
3. Implement unsupervised learning techniques to discover hidden patterns in data.
4. Evaluate and optimize machine learning models using performance metrics.

Unit – I	**Introduction to Machine Learning**	**3 Hours**

What is Machine Learning? How do machine learn, Types of Machine Learning: Supervised learning, unsupervised learning, Reinforcement learning, Problems not to be solved using Machine Learning, Applications of Machine learning.

Unit – II	**Data Pre-processing**	**8 Hours**

Introduction, Machine Learning Activities, Basic Types of Data in Machine Learning, Exploring Structure of Data, Data Quality and Remediation, Basics of Feature Engineering: Introduction, Feature Transformation, Feature Subset Selection

Unit – III	**Modelling and Evaluation**	**7 Hours**

Introduction, selecting a Model, training a Model (for Supervised Learning), Model Representation and Interpretability, Evaluating Performance of a Model, Improving Performance of a Model

Unit – IV	Supervised Learning: Classification	7 Hours

Introduction, Examples of Supervised Learning, Classification model, Classification learning steps, Common Classification Algorithms-Naïve Bayes classifier- Bayes theorem, Handling continues numeric features in Bayes classifier, k-Nearest Neighbour's (k-NN), Decision tree, Random Forest model, Support Vector Machine.

Unit – V	Supervised Learning: Regression	8 Hours

Introduction, Examples of regression, Common regression algorithms-Simple Linear regression, Multiple linear regression, Assumptions in regression analysis, Main Problems in regression analysis, Improving accuracy of the linear regression model, Polynomial regression model, logistic regression model.

Unit – VI	Unsupervised Learning	7 Hours

Introduction, Unsupervised vs. Supervised learning, Applications of Unsupervised learning, Clustering: Clustering as a Machine learning task, different types of clustering techniques, Partitioning methods, k-means algorithm, k-medoids, Hierarchical clustering, Density based methods-DBSCAN, finding patterns using association rule-Definition of common terms, Association rule, Apriori algorithm

ICA consists of minimum 10 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:

1. Basic mathematics for Machine Learning – Simulating solutions using Python to: I. Matrix operations II. Problems using Probability
2. Data exploration, Visualization and Pre-processing
3. Feature Engineering
4. Naive-Bayes classifier
5. Decision Tree regressor
6. K-Nearest Neighbour (KNN) classifier
7. Logistic Regression classifier
8. Support vector machine (SVM) Classifier
9. Linear and Multilinear Regression
10. Ensemble Models

Text Books

1. Machine Learning Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, Pearson Publication
2. Machine Learning for Dummies John Paul Mueller, Luca Massaron, First edition, Dummies

Reference Books

1. Machine Learning by Tom M. Mitchell, Second edition, McGraw Hill Education
2. Introduction to Machine Learning (Second Edition) Ethem Alpaydın, The MIT Press Cambridge, Massachusetts London, England
3. Introduction to Machine Learning with Python Andreas C. Muller & Sarah Guido, O' Really

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5E24T: Programme Elective I - Cloud Computing

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

Cloud computing has rapidly become a transformative force in the ICT industry, emerging as a key paradigm for delivering computing resources on demand. A wide range of IT vendors now offer services that include computation, storage, and application hosting, often with global reach and robust Service Level Agreements (SLAs) that ensure performance and high availability. This course provides a comprehensive introduction to the fundamentals of cloud computing. Topics include virtual machine provisioning and migration services, classification of cloud services and applications, integration strategies for private and public cloud environments, as well as critical issues related to cloud security and legal considerations in cloud computing.

Course Prerequisite:

To effectively understand and work with cloud computing technologies, students should have foundational knowledge in the following areas: Basic Computer Networks, Operating Systems, Database Knowledge Computer Architecture, Programming Skills, Basics of Virtualization, Familiarity with Linux/Unix Commands.

Course Objectives:

1. To introduce the fundamentals and architecture of cloud computing, including its layers, types, and historical evolution.
2. To explore virtualization technologies and virtual machine provisioning and migration in the context of cloud infrastructure.
3. To analyse major cloud service providers (AWS, Azure, GCP) in terms of services, pricing models, and deployment models (IaaS, PaaS, SaaS).
4. To understand the security and compliance requirements in cloud environments, including IAM, audit, and application-level security.
5. To provide hands-on exposure to popular cloud services such as AWS EC2, S3, Lambda, RDS, and developer tools, and explore private-public cloud integration using platforms like Aneka and Comet.

Course Outcomes:

After completing the course, students will be able to

1. Explain cloud computing fundamentals, including architecture, service models, infrastructure management, and migration strategies.
2. Configure and manage virtual machine provisioning and migration tasks in simulated or real-world cloud environments.
3. Evaluate hybrid cloud solutions using Aneka and Comet Cloud in terms of resource provisioning, autonomic behaviour, and real-world applicability.
4. Analyse cloud security challenges and legal concerns, including data privacy, compliance standards, and identity management.

Unit – I	**Introduction to Cloud Computing**	**8 Hours**
Cloud Computing in a Nutshell, Roots of Cloud Computing, Layers and Types of Clouds, Desired Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks, migrating into a Cloud, Introduction, Broad Approaches to Migrating into the Cloud, The Seven-Step Model of Migration into a Cloud.		
Unit – II	**Virtual Machines Provisioning and Migration Services**	**4 Hours**
Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action, Provisioning in the Cloud Context.		
Unit – III	**Virtualization in Cloud Computing**	**6 Hours**
Introduction, Characteristics of virtualized environments, Virtualization and cloud computing, Pros and cons of virtualization, Technology examples.		
Unit – IV	**Integration of Private and Public Clouds**	**8 Hours**
Aneka: Introduction, Technologies and Tools for Cloud Computing, Aneka Cloud Platform, Aneka Resource Provisioning Service, Hybrid Cloud Implementation. Comet: Comet Cloud Architecture, Autonomic Behaviour of Comet Cloud, Overview of Comet Cloud-based Applications.		
Unit – V	**Security in Cloud Computing**	**7 Hours**
Infrastructure Security: Network Level, Host Level, Application Level, Identity and Access Management (IAM), Security Management in the Cloud, Audit and Compliance.		
Unit – VI	**Cloud offered services.**	**7 Hours**
Compute Services (Amazon EC2, EC2 Global View, Elastic Beanstalk, AWS Lambda), Storage Services (Amazon S3, EFS, S3Glacier), Database Services (Amazon RDS, Elastic Cache, Dynamo DB, Networking Services (Amazon VPC, Route 53, Developer Tools and SDKs (AWS CLI)		

Internal Continuous Assessment (ICA):

Minimum 10 assignments must be of nature, which require students to identify and implement the use case scenarios for Cloud and Cloud enabled technologies mentioned above.

Text Books

1. Cloud Computing: Principles and Paradigms Rajkumar Buyya, James Broberg, & Andrzej M. Goscinski, First Edition, Wiley Publication.
2. Cloud Security and Privacy Tim Mather,Subra Kumarasway, Shahed Latif, O'Reilly Media
3. Cloud Computing Bible Barrie Sosinsky Wiley Publication.
4. Cloud Computing for Dummies Judith Hurwitz, Daniel Kirsch, Second Edition, Wiley Publication

Reference Books

1. Cloud Native Patterns: Designing Change-tolerant Softwar Cornelia Davis, First Edition, O' Reilly Media.
2. Cloud Security, A comprehensive Guide to Secure Cloud Computing Krutz, Ronald L. Vines, Russell Dean
3. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, McGraw Hill.
4. Cloud Computing: A Practical Approach, Anthony T. Velte, Toby J. Velte, & Robert Elsenpeter, McGraw Hill.

e-Resources

AWS Documentation: URL- https://docs.aws.amazon.com/

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
(An Autonomous Institute)
Third Year B.Tech. (Information Technology), Semester-V

23ITU5E34T: Programme Elective I - High Performance Computing

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

This course provides the fundamentals of high-performance and parallel computing.

Course Prerequisite:

Data structures, Basic Programming knowledge

Course Objectives:

1. To be introduced with current trends in parallel computer architectures and programming models (i.e., languages and libraries) for shared memory, many core/multicore architectures.
2. To understand parallel program design methodology. Also, to calculate speedup and efficiency of parallel algorithm.
3. To learn various parallel algorithms for matrices, graphs.

Course Outcomes:

After completing the course, students will be able to

1. Describe different parallel paradigms, inter connection networks, and tools for parallel programming.
2. Illustrate the design methodology and relevant parallel programming techniques to be used for parallelization of a given problem.
3. Analyse a given problem for possibilities of parallel computations.
4. Evaluate different parallel algorithms using performance metrics.

Unit – I	**Introduction**	**8 Hours**

What is parallel computing? The scope of parallel computing? Issues in parallel computing. Taxonomy of parallel architecture, Memory bound vs Compute bound problems, Dynamic interconnection networks, static interconnection networks, Routing mechanism for static network. Communication cost in static interconnection network.

Unit – II	**Parallel programming models and paradigms**	**6 Hours**

Introduction, parallel applications and development, code granularity and level of parallelism, parallel programming models and tools, methodical design of parallel algorithm, parallel program paradigm, programming skeleton and templates.

<table>
<tr><td><b>Unit – III</b></td><td><b>Parallel programming libraries</b></td><td><b>6 Hours</b></td></tr>
<tr><td colspan="3">OpenMP, MPI, Thread basics, Work Sharing constructs, Scheduling, Reduction, Mutual Exclusion Synchronization & Barriers, The MPI Programming Model, MPI Basics, Global Operations, Asynchronous Communication, Modularity, Other MPI Features, Performance Issues, Thread programming C++11 Threads /OpenMP, MPI - two-sided communication, one side communication-based programming model aka PGAS (Partitioned Global Address Space) eg: OpenSHMEM / NVSHMEM</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Performance and scalability of parallel systems</b></td><td><b>8 Hours</b></td></tr>
<tr><td colspan="3">Performance Metrics for parallel systems. The effect of Granularity and Data Mapping on Performance. The Scalability of parallel systems, Iso-efficiency metric of scalability, sources of parallel overhead, Minimum execution time and minimum cost-optimal execution time, parallel work efficiency, Amdahl limiters, communication-computation overlap/pipelining.</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Parallel programming using accelerators</b></td><td><b>6 Hours</b></td></tr>
<tr><td colspan="3">Introduction of CUDA/OpenCL, Chapel, etc. Basics of GPGPU, CUDA Programming model, CUDA memory type, CUDA and/or OpenCL for GPGPU hardware, case study</td></tr>
<tr><td><b>Unit – VI</b></td><td><b>Algorithms</b></td><td><b>6 Hours</b></td></tr>
<tr><td colspan="3">Dense matrix algorithms, sorting, graph algorithms, prefix sum with decoupled lookback, parallel radix sort/batcher's sort</td></tr>
<tr><td colspan="3">ICA consists of minimum 10 assignments based on the above topics:</td></tr>
</table>

Text Books

1. Introduction to Parallel Computing - y Ananth Grama, Anshul Gupta, George Karypis, & Vipin Kumar, Second Edition, Pearson Publication
2. High Performance Cluster Computing: Programming and Applications Volume 2 Buyya Rajkumar
3. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs - Shane Cook
4. Introduction to PARALLEL PROGRAMMING Peter Pacheco.

Reference Books

1. Parallel Programming in C with MPI and OpenMP Michael J. Quinn, McGraw-Hill.

e-Resources

1. Single-pass Parallel Prefix Scan with Decoupled Look-back https://research.nvidia.com/publication/single-pass-parallel-prefix-scan-decoupled-lookback
2. parallel radix sort/batcher's sort. https://developer.download.nvidia.com/video/gputechconf/gtc/2020/presentations/s215 72-afaster-radix-sort-implementation.pdf
3. High Performance Computing, Charles Severance, http://cnx.org/content/col11136/latest/
4. MPI: The Complete Reference, Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, and Jack Dongarra, http://www.netlib.org/utk/papers/mpi-book/mpi-book.html

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-V

23ITU5IK5T: Indian Knowledge System II - Vedic Mathematics

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours/week	**ESE**	-
Practical/Tutorial	-	**ISE**	50 Marks
Credits	2	**ICA**	-

Introduction:

Vedic Mathematics is a super-fast way of calculation. There are just 16 Sutras or Word Formulae which solve all known mathematical problems in the branches of Arithmetic, Algebra, Geometry, and Calculus. They are easy to understand, easy to apply, and easy to remember.

Course Prerequisite:

Student shall have knowledge of basic mathematics.

Course Objectives:

1. Introduce students to the foundational principles and principles of Vedic Mathematics.
2. Develop a mental calculation skill and enhance number sense.
3. Promote problem-solving abilities through mental and arithmetic techniques.
4. Build confidence in handling mathematical calculations quickly and accurately.
5. Foster an interest in mathematics by making it more approachable and engaging.

Course Outcomes:

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

1. Describe history and evolution of Vedic mathematics.
2. Perform Vedic arithmetic calculations with speed and accuracy.
3. Apply various Multiplication and Division methods to solve problems.
4. Calculate powers and roots of a number.
5. Solve simultaneous and quadratic equations.

Unit – I	**Introduction of Vedic Mathematics**	**5 Hours**

History and Evolution of Vedic Mathematics, Introduction of Basic Vedic Mathematics Techniques in Multiplication (Special Case, Series of 9, Series of 1 etc), Tables etc., Comparison of Standard Methods with Vedic Methods, sixteen sutras.

Unit – II	**Vedic Arithmetic Operations**	**5 Hours**

Various techniques to carry out basic operations covering Addition, Subtraction, Multiplication, Division, Complements and Bases, Vinculum number, General multiplication (Vertically Cross-wise).

Unit – III	**Multiplication and Division**	**5 Hours**
Multiplications by numbers near base, verifying answers by use of digital roots, Divisibility tests, Division of numbers near base, Comparison of fractions.		
Unit – IV	**Power and roots**	**10 Hours**
Applications of Vinculum, Different methods of Squares (General method, Base method, Duplex method etc.), Square roots, Cubes, Cube roots, General division		
Unit – V	**Vedic Algebra**	**5 Hours**
Simultaneous Equations, Quadratic Equations		
Text Books		
1. Bhatiya Dhaval, Vedic Mathematics Made Easy, Jaico Publishing House 2. Thakur Rajesh Kumar, Vedic Mathematics for students taking Competitive Examinations. Unicorn Books 2015 or Later Edition 3. Gupta Atul, Power of Vedic Mathematics with Trigonometry, Jaico Books 4. V. G. Unkalkar, Magical World of Mathematics (Vedic Mathematics), Vandana Publishers, Bangalore 5. Vedic Math Genius by Kenneth Williams, Inspiration Books, England		
Reference Books		
1. Vedic Mathematics: Sixteen Simple Mathematical Formulae from The Vedas Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaj Motilal Banarsidass Publishers Pvt. Ltd., Delhi 2. Lilavati: A Treatise of Mathematics of Vedic Tradition Krishnaji Shankara Patwardhan published Motilal Banarsidass Publishers Pvt. Ltd., Delhi 3. The Power of Vedic Maths by Atul Gupta, Jaico Publishing house, Mumbai 4. Vertically And Crosswise by A. P. Nicholas, K. R. Williams, J. Pickles, Inspiration Books, Scotland		

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
(An Autonomous Institute)
Third Year B.Tech. (Information Technology), Semester-V

23ITU5CC8P: Advanced Java

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours /week	**ISE**	25 Marks
Practical	2 Hours /week	**ICA**	25 Marks
Credits	3	**POE**	25 Marks

Introduction:

Advanced Java is an upgraded and specialized version of Java, primarily designed for developing web-based applications. It encompasses key technologies such as Servlets, JSP (Java Server Pages), JDBC, and more. These components significantly enhance the functionality and performance of web applications. For developers, mastering these advanced Java concepts is essential, as they enable the creation of dynamic, interactive, and robust web solutions.

Course Prerequisite:

Students should have knowledge of Core Java with Swing.

Course Objectives:

1. Implement JDBC for database connectivity.
2. Develop web applications using Servlets.
3. Create dynamic server-side pages with JSP.
4. Build modular web applications using the Spring Framework.
5. Design scalable applications with Hibernate ORM.

Course Outcomes:

After completing the course, students will be able to,

1. Demonstrate JDBC connectivity for efficient CRUD operations and result set handling with relational databases.
2. Develop dynamic web apps with Java Servlets, using session management, HTTP methods, and servlet lifecycle.
3. Create server-side apps with JSP using scripting elements, directives, implicit objects, and error handling.
4. Apply Spring Framework to build modular web apps with Spring MVC and DAO via XML and annotations.
5. Build robust, scalable applications using Hibernate ORM by mapping Java classes to database tables and executing database operations using Hibernate APIs.

Unit – I	**Introduction to JDBC**	**6 Hours**

JDBC Introduction, JDBC features, JDBC Drivers, setting up a database and creating a schema, Writing JDBC code to connect to DB, CRUD Operations with JDBC, Statement types in JDBC, Types of Rowset, ResultSet in JDBC.

Unit – II | **Servlets and Session Management** | **6 Hours**

Servlets Introduction, Need and Working, HTTP Methods; GET, POST, PUT, DELETE, TRACE, OPTIONS. GET/POST request; differences between the two, Servlet Lifecycle, Servlet-Context and Servlet-Config, Forwarding and Redirection of requests.

Unit – III | **Java Server Pages** | **6 Hours**

Introduction, Difference between JSP and servlet, Life cycle of JSP, JSP elements- Scriptlets, Expressions, Declarations, JSP Directives, Working with JSP basic tags and Implicit objects, Exception Handling in JSP.

Unit – IV | **Spring Framework** | **6 Hours**

Introduction to Spring, Spring Architecture explanation and all its components, Spring MVC, Spring DAO, setting up of Spring framework Download JARs, Configure XML files.

Unit – V | **Hibernate Framework** | **6 Hours**

ORM, ORM principle, ORM Implementation, Introduction to Hibernate, Hibernate Architecture, Persistent classes, Hibernate CRUD, setting up connection to DB using Hibernate.

Internal Continuous Assessment (ICA):
ICA consists of minimum 8 to 10 assignments based on the above given topics.

Text Books

1. Java The Complete Reference, Ninth Edition, Herbert Schildt, McGraw Hill Education
2. Head First Servlets and JSP, Kathy Sierra, Bryan Basham, Bert Bates, O'Reilly Media, Inc.

Reference Books

1. Spring Micro services in Action, Second Edition, John Carnell, Manning Publications.
2. Core and Advanced Java, Black Book, Dreamtech Press.
3. Java Programming for Core and advanced learners, Sagayaraj, Denis, Karthik and Gajalakshmi, Universities Press.
4. Java Web Services: Up and Running, Second Edition, Martin Kalin, O'Reilly Media, Inc.

e-Resources

1. https://hibernate.org/orm/documentation/6.0/
2. https://docs.spring.io/spring-framework/docs/current/reference/html/

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6CC1T: Artificial Intelligence

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

This course presents a basic introduction to the techniques used in developing Artificial Intelligent systems. It is a walkthrough to problem spaces and search algorithms, Knowledge representation, reasoning, logic programming and applications of Artificial Intelligence.

Course Prerequisite:

Student shall have some exposure to algorithms and programming.

Course Objectives:

1. To describe the fundamental concepts, scope, problems, and various approaches of Artificial Intelligence (AI) and classify types of intelligent agents.
2. To develop abilities to formulate, analyse, and solve AI problems using search strategies and logical reasoning frameworks.
3. To explain the methods for modelling, evaluating, and inferring under uncertainty and to apply decision-making techniques in AI systems.
4. To illustrate various machine learning paradigms and demonstrate basic models for supervised, unsupervised, and reinforcement learning.

Course Outcomes:

After completing the course, students will be able to

1. Define the scope of Artificial Intelligence and classify intelligent agents based on their behaviours.
2. Apply appropriate search techniques and analyse logical reasoning methods to solve structured AI problems.
3. Model uncertain knowledge using probabilistic reasoning and evaluate decision-theoretic approaches for AI-based decision-making.
4. Differentiate various learning techniques and implement basic supervised, unsupervised, and reinforcement learning models for sample applications.

Unit – I	**Introduction to Artificial Intelligence: Foundations, Challenges, and Intelligent Agents**	**7 Hours**

Foundations, scope, problems, and approaches of AI. **Intelligent agents:** reactive, deliberative, goal-driven, utility-driven, and learning agents.

Unit – II	**Problem-solving through Search**	**8 Hours**
Forward and backward, state-space, blind, heuristic, problem-reduction, A, A*, AO*, minimax, constraint propagation, neural, stochastic, and evolutionary search algorithms, sample applications.		
Unit – III	**Knowledge Representation and Reasoning**	**8 Hours**
Ontologies, foundations of knowledge representation and reasoning, representing and reasoning about objects, relations, events, actions, time, and space; first order logic, situation calculus, description logics, reasoning with defaults, reasoning about knowledge, sample applications.		
Unit – IV	**Representing and Reasoning with Uncertain Knowledge**	**7 Hours**
Probability, connection to logic, independence, Bayes rule, Bayesian networks, probabilistic inference, and sample applications.		
Unit – V	**Decision-Making**	**8 Hours**
Basics of utility theory, decision theory, sequential theory, decision problems, elementary, game, sample applications.		
Unit – VI	**Learning and Knowledge Acquisition**	**7 Hours**
Forms of Learning: Supervised, Unsupervised, Semi-supervised. Statistical learning, Reinforcement Learning: Q-learning, sample applications.		

ICA consists of minimum 10 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:

- Intelligent agents
- Problem solving through search
- First order logic
- Bayesian Networks
- Decision and Game theory
- Statistical Learning
- Q-learning.

In addition to above students shall undertake a case study on **“Applications of AI: The Present and Future**

Text Books

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Third Edition, Prentice Hall
2. A First Course in Artificial Intelligence, Deepak Khemani, McGraw Hill Education (India)
3. Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PHI.

Reference Books

Artificial Intelligence, Elaine Rich and Kevin Knight, Tata McGraw Hill

e-Resources
• **NPTEL Course: Artificial Intelligence (IIT Kharagpur)** **Instructor:** Prof. P. Dasgupta **Course Link:** Artificial Intelligence - NPTEL • Course: Artificial Intelligence: Search Methods for Problem Solving (IIT Madras) Instructor: Prof. Deepak Khemani Course Link: Artificial Intelligence: Search Methods for Problem Solving - NPTEL

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6CC2T: Internet of Things

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours/week	**ESE**	60 Marks
Practical	2 Hours/week	**ISE**	40 Marks
Credits	3	**ICA**	25 Marks

Introduction:

This course introduces the fundamentals, architecture, and applications of the Internet of Things, equipping students with knowledge to design, develop, and implement IoT systems.

Course Prerequisite:

Fundamentals of Communication and Computer Network

Course Objectives:

1. To introduce the basic concepts, evolution, and characteristics of the Internet of Things and its impact on modern technologies
2. To provide knowledge of various sensors, actuators, and their integration in real-time IoT systems.
3. To familiarize students with communication models and protocols used in IoT networks.
4. To study the genesis and impact of IoT applications and architectures in the real world.
5. To develop skills in designing and implementing real-world IoT applications using suitable programming languages, tools, and hardware

Course Outcomes:

After completing the course, students will be able to

1. Interpret the fundamentals, evolution, and architecture of IoT systems.
2. Identify and describe various sensors, actuators, and their roles in IoT applications
3. Analyse communication protocols and networking aspects pertinent to IoT.
4. 4. Design and develop IoT applications integrating hardware and software components

Unit – I	**Introduction to Internet of Things (IoT)**	**4 Hours**

Definition, applications and IoT characteristics, and evolution of IoT, IoT enabling technologies and ecosystem components, Overview of IoT architecture and deployment models, IoT levels

Unit – II	**Sensors, Actuators and Micro Controllers**	**5 Hours**

Sensor Technology, Types and characteristics of sensors and actuators, Sensor data acquisition and signal conditioning, Integration of sensors and actuators in IoT systems, RFID, WSN Technology Micro Controllers (Arduino uno/Raspberry-Pi/ARM), Real-time systems, and embedded software

Unit – III	IoT Networking and Communication Protocols	6 Hours
Networking fundamentals for IoT Communication protocols: MQTT, CoAP, HTTP, etc. Wireless technologies: Bluetooth, Zigbee, Wi-Fi, LoRaWAN		
Unit – IV	**IoT Architecture and Platforms**	**6 Hours**
IoT reference architectures, Cloud and edge computing in IoT, IoT platforms and middleware solutions, Data storage and analytics in IoT.		
Unit – V	**IoT Security and Business model**	**5 Hours**
Security and privacy considerations in IoT applications, Security and Vulnerabilities, Use case and Misuse cases, IoT Security Tomography and Layered attacker model, Business model and business model innovation for IoT, Value Creation in the IoT, Business model scenarios for IoT.		
Unit – VI	**Case Studies**	**4 Hours**
Domain Specific IoTs: Home Automation, Smart Cities, Environments, Energy, Agriculture, Industry, Health and Lifestyle, Future trends and challenges in IoT.		
Internal Continuous Assessment (ICA): The ICA shall consist of design and implementing a mini project based on the techniques and tools covered in above chapters.		
Text Books		
1. Internet of Things: A Hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press (Unit 1 and 6). 2. IoT Architecture and Design Principles, Raj Kamal, McGraw Hill Education (Unit 2, 3 & 5). 3. Building the IoT with IPv6 and MIPv6, Daniel Minoli, Wiley Publication (Unit 4).		
Reference Books		
1. The Internet of Things: Applications and Protocols, Wiley publications. Author(s): Oliver Hersent, David Boswarthick, Omar Elloumi		

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
(An Autonomous Institute)
Third Year B.Tech. (Information Technology), Semester-VI

23ITU6CC3T: Software Engineering

Teaching Scheme		**Examination Scheme**	
Lectures	2 Hours/week	**ESE**	-
Practical	2 Hours/week	**ISE**	25 Marks
Credits	3	**ICA**	50 Marks
		OE/POE	25 Marks

Introduction:

Software engineering integrates engineering principles and computer science concepts to develop reliable, usable and dependable software. This course introduces essential software engineering concepts and methodologies, aligned with the software development life cycle. This course equips students with skills to address real-world software engineering challenges.

Course Prerequisite:

Fundamental knowledge of programming principles is required.

Course Objectives:

1. Analyse various software development lifecycle models to select and apply the most suitable model for a given project.
2. Prepare Software Requirements Specification (SRS) and Software Design Specification (SDS) documents for a given problem.
3. Apply appropriate testing methods.
4. Emphasize quality management throughout the software development process.

Course Outcomes:

After completing the course, students will be able to

1. Select and apply the appropriate lifecycle model for software development.
2. Prepare SRS and SDS accordingly for a given problem.
3. Select and apply appropriate software testing methods.
4. Ensure the quality of a product by applying the quality management process.

Unit – I	**Introduction to Software Engineering**	**6 Hours**

Introduction, Software Engineering Challenges and Approach, Software Process, Characteristics of Software Process, Software Development Process Models: Waterfall Model, Prototyping Model, Iterative Development Model: Incremental Model, Spiral Model, Time Boxing Model, Agile Process Model.

Unit – II	**Software Requirement Analysis & Specification**	**6 Hours**

Need for SRS, Characteristics of Good SRS, Requirement Process, Requirements Specification, Functional Specification with Use Cases, Other Approaches for Analysis: Data Flow Diagram (DFD), Entity Relationship Diagram (ERD).

Unit – III	**Software Architecture and Design**	**8 Hours**
Introduction to Software Design, Software Architecture: Role of Software Architecture, Architecture Views, Component & Connector View, Architecture Style for Component & Connector View, Documenting Architecture Design, Function-Oriented Design: Design Principles, Module Level Concepts, Design Notation and Specification, Structured Design Methodology, Object-Oriented Design: Analysis and Design, OO Concepts, Design Concepts.		
Unit – IV	**Testing**	**4 Hours**
Testing Fundamentals, Testing Process, Black-Box Testing, White-Box Testing, Object-Oriented Software Testing Methods, Functional Testing, Unit Testing, System Testing, User Satisfaction Testing.		
Unit – V	**Project Planning and Management**	**6 Hours**
Project Management Process, The Inspection and Audit Process, Software Configuration Management Process, Effort Estimation, Project Schedule and Staffing, Quality Planning: Quality Concepts, Qualitative Quality Management Planning, CMM Project Management Process, Risk Management Planning, Project Monitoring Plan, Detailed Scheduling.		

The ICA (Internal Continuous Assessment) involves at least 10 assignments where students must create software engineering applications that solve real-world problems based on the following topics:

1. Introduction to Software Engineering
2. Software Engineering Challenges and Approach
3. Software Process and Characteristics
4. Software Development Models
5. Need for Software Requirement Specification (SRS)
6. Characteristics of Good SRS and Requirement Process
7. Functional Specification with Use Cases, DFD, ERD
8. Software Architecture and Its Role
9. Design Concepts in Software Engineering
10. Fundamentals of Software Testing
11. Software Project Planning and Management
12. Quality Planning in Software Projects

Text Books

An Integrated Approach to Software Engineering - 3rd edition: Pankaj Jalote (Narosa Publishers)

Reference Books

1. Ian Sommerville, Software Engineering Pearson Education Asia, Sixth edition
2. Software Engineering Fundamentals Ali Behforooz & Frederick J. Hudson Oxford University Press
3. Project Management with Scrum - Ken Schwaber

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E14T: Programme Elective II - Reinforcement Learning

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Tutorial	1 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

Reinforcement learning is an area of machine learning, where an agent or a system of agents learns to archive a goal by interacting with their environment. In recent years there has been success in reinforcement learning research in both theoretical and applied fields. This course primarily focuses on training students to frame reinforcement learning problems and to tackle algorithms from dynamic programming, Monte Carlo and temporal-difference learning.

Course Prerequisite:

A basic course on Artificial Intelligence & Machine learning

Course Objectives:

1. Learn how to define RL tasks and the core principles behind Reinforcement Learning
2. Identify the core principles for arriving at Reinforcement learning.
3. Create codes to simulate various algorithms demonstrating Reinforcement learning
4. Analyse and find different methods to apply Reinforcement Learning to generate problem solutions.

Course Outcomes:

After completing the course, students will be able to

1. Demonstrate the use of Reinforcement Learning tasks.
2. Describe the core principles of applying Reinforcement Learning.
3. Implement in code commonly used principles in Reinforcement Learning
4. Identify current advanced techniques and applications in Reinforcement Learning

Unit – I	**Introduction to Reinforcement Learning**	**8 Hours**

Reinforcement Learning, Examples, Elements of Reinforcement Learning, The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, Value Functions, Optimal Value Functions, Optimality and Approximation. History of Reinforcement Learning.

Unit – II	**Evaluation Feedback**	**6 Hours**

A k-armed Bandit Problem, Action-value Methods, The 10-armed Test-bed, Incremental Implementation

<table>
<tr><td><b>Unit – III</b></td><td><b>Finite Markov Decision Processes</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Unified Notation for Episodic and Continuing Tasks, Policies and Value Functions.</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Dynamic Programming</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration, Efficiency of Dynamic Programming, Introduction to Monte Carlo Methods</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Temporal-Difference Learning</b></td><td><b>8 Hours</b></td></tr>
<tr><td colspan="3">TD Prediction, Advantages of TD Prediction Methods, Optimality of TD (0), SARSA: On-policy TD Control, Q-learning: Off-policy TD Control.</td></tr>
<tr><td><b>Unit – VI</b></td><td><b>Applications and Case Studies</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">TD-Gammon, Samuel’s Checkers Player, Watson’s Daily-Double Wagering, Mastering the Game of Go and Alpha Go.</td></tr>
<tr><td colspan="3"><b>ICA consists of under mentioned practical based on curriculum. Recommended practical’s:</b>
1. Creating the Reinforcement Learning Environment
2. Simulating the k-armed Bandit Problem
3. Simulating Finite Markov Decision Processes on environment created
4. Applying Dynamic Programming to Generate optimized solutions
5. Applying Temporal-Difference Learning to the created environment</td></tr>
<tr><td colspan="3"><b>Text Books</b></td></tr>
<tr><td colspan="3">Reinforcement Learning: An Introduction (Second edition + Upcoming Edition) by: Richard S. Sutton and Andrew G. Barto, MIT Press Publication
(The book is available at http://incompleteideas.net/book/the-book-2nd.html Upcoming edition's January 1 2018 draft available at http://incompleteideas.net/book/bookdraft2018jan1.pdf]</td></tr>
<tr><td colspan="3"><b>Reference Books</b></td></tr>
<tr><td colspan="3">1. Reinforcement Learning: With Open AI, TensorFlow and Keras Using Python Abhishek Nandy, Manisha Biswas. Apress Publication
2. Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds.
3. Artificial Intelligence: A Modern Approach, Stuart J. Russell & Norvig.
4. Deep Learning, Ian Goodfellow, Yoshua Bengio & Aaron Courville.</td></tr>
</table>

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
(An Autonomous Institute)
Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E24T: Programme Elective II - Network Security

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Tutorial	1 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks

Introduction:

This course introduces the foundational concepts of network security and cryptography, along with their applications in protecting information systems. It covers cryptographic algorithms, security protocols, and practical measures to mitigate vulnerabilities across network and transport layers.

Course Prerequisite:

Students should have basic knowledge of computer networks, fundamentals of operating systems, and introductory programming concepts.

Course Objectives:

1. **To explain** the fundamental principles of network security models and classical encryption techniques.
2. **To illustrate** the design and working of block ciphers, public key cryptography, and key distribution mechanisms.
3. **To develop** understanding of network, internet, and transport-level security standards and protocols.
4. **To analyze** wireless network vulnerabilities and recommend security solutions for mobile and cloud environments.

Course Outcomes:

After completing the course, students will be able to

1. Describe classical encryption techniques, cryptographic principles, and cyber security frameworks.
2. Apply symmetric and asymmetric key cryptography techniques to secure information systems.
3. Demonstrate understanding of network and transport layer protocols for secure communication.
4. Analyze wireless and mobile security standards and propose appropriate countermeasures for network security threats.

Unit – I	Security Fundamentals	8 Hours

Computer Security Concepts, OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Classical Encryption Techniques including Symmetric Cipher Model and Cryptography, Substitution Techniques such as Caesar Cipher, Monoalphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Cipher, Transposition Techniques, Rotor Machines, and Steganography.

Unit – II	**Block Ciphers and the Data Encryption Standard**	**8 Hours**
Traditional Block Cipher Structure covering Stream Ciphers and Block Ciphers, Motivation for the Feistel Cipher Structure, The Feistel Cipher, Data Encryption Standard (DES) including DES Encryption, DES Decryption, and The Strength of DES, Public-Key Cryptography and RSA including Principles of Public Key Cryptosystems, Description of the RSA Algorithm, Computational Aspects, and The Security of RSA.		
Unit – III	**Key Management and Distribution**	**8 Hours**
Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, and Public-Key Infrastructure (PKI).		
Unit – IV	**Network and Internet Security**	**7 Hours**
Network Access Control and Cloud Security covering Network Access Control, Extensible Authentication Protocol (EAP), IEEE 802.1X Port-Based Network Access Control,		
Unit – V	**Wireless Network Security**	**8 Hours**
Wireless LAN Security with IEEE 802.11 Overview and Security Standards, Mobile Device Security, Electronic Mail Security including Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME), IP Security Basics including Overview of IP Security, IP Security Policy, and Encapsulating Security Payload (ESP).		
Unit – VI	**IP Security**	**7 Hours**
IP Security Overview and Architecture, IP Security Policy, Encapsulating Security Payload (ESP), Combining Security Associations, and Internet Key Exchange (IKE).		

ICA consists of minimum 10 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:

Security fundamentals, classical encryption techniques, block ciphers, public-key cryptography, key management and distribution, network access control, cloud security mechanisms, transport layer security protocols (SSL, TLS, HTTPS, SSH), wireless network security, mobile device security, email security (PGP, S/MIME), IP security protocols (ESP, IKE).

Text Books

1.Computer Security: Principles and Practices Willaim Stallings Pearson Publication

2.Network Security and Cryptography, Bernard Menezes Cengage Learning,

3.Cyber Security-Understanding Cybercrimes Computer Forensics and Legal Perspectives Nina Godbole

Reference Books
Information Systems Security Nina Godbole Wiley Publication
e-Resources
NPTEL Course: **"Network Security and Cryptography"** Instructor: Prof. D. Mukhopadhyay, IIT Kharagpur Link: https://nptel.ac.in/courses/106105031

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E34T: Programme Elective II - Data Science

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Tutorial	1 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

This course introduces students to Data Science fundamentals, covering essential techniques for data collection, pre-processing, analysis, visualization, and machine learning.

Students will learn to work with real-world datasets using Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-Learn, gaining hands-on experience in data-driven decision-making.

The course emphasizes both theoretical and practical aspects of data science, preparing students for roles in AI, Data Analytics, and Big Data applications.

Course Prerequisite:

Basic programming knowledge in Python (loops, functions, data structures) and familiarity with libraries like NumPy and Pandas is recommended.

Fundamental understanding of Mathematics and Statistics, including probability, linear algebra, and hypothesis testing.

Basic SQL knowledge to interact with structured databases and perform queries for data extraction and manipulation.

Course Objectives:

1. Understand the fundamental concepts and importance of Data Science.
2. Learn how to pre-process, clean, and visualize data effectively.
3. Develop a strong foundation in probability and statistics for data analysis.
4. Gain knowledge of core machine learning techniques and their applications.

Course Outcomes:

After completing the course, students will be able to

1. Describe the fundamentals and real-world applications of Data Science.
2. Apply data pre-processing techniques, handle missing values, and create visualizations.
3. Develop and implement basic machine learning models to analyse performance.
4. Utilize big data technologies and advanced data science techniques.

Unit – I	**Introduction to Data Science**	**05 Hours**

Definition and Importance of Data Science, Applications of Data Science in Various Domains, Data Science Lifecycle, Overview of AI, ML, and Deep Learning.

<table>
<tr><td><b>Unit – II</b></td><td><b>Data Pre-processing and Visualization</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Data Collection, Cleaning and Pre-processing, Handling Missing Values and Outliers, Data Transformation and Feature Engineering, Data Visualization using Matplotlib and Seaborn, Introduction to Exploratory Data Analysis (EDA)</td></tr>
<tr><td><b>Unit – III</b></td><td><b>Probability and Statistics for Data Science</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Descriptive and Inferential Statistics, Probability Distributions (Normal, Binomial, Poisson), Hypothesis Testing and Confidence Intervals, Correlation and Regression Analysis, Bayesian Statistics and Probability.</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Machine Learning Basics</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Introduction to Supervised and Unsupervised Learning, Regression Models (Linear, Polynomial, Logistic), Classification Models (Decision Trees, SVM, k-NN, Naïve Bayes), Model Evaluation Metrics, Hyperparameter Tuning and Model Optimization</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Working with Big Data</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Introduction to Big Data and Hadoop, Working with SQL and NoSQL Databases, Basics of Apache Spark for Data Processing, Cloud-based Data Science (AWS, Google Cloud, Azure), Data Streaming and Real-Time Analytics.</td></tr>
<tr><td><b>Unit – VI</b></td><td><b>Advanced Topics in Data Science & Case Studies and Applications</b></td><td><b>09 Hours</b></td></tr>
<tr><td colspan="3">Feature Engineering and Selection, Introduction to Deep Learning and Neural Networks, Time Series Analysis, Natural Language Processing (NLP) Basics, Real-world Data Science Applications in Healthcare, Finance and Marketing, Ethical Considerations in Data Science</td></tr>
<tr><td colspan="3">ICA consists of minimum 10 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:
1. Exploring and Understanding a Dataset-Load a real-world dataset using Pandas. ,Display basic statistics (mean, median, mode, standard deviation)., Identify missing values and data types.
2. Data Science Lifecycle-Collecting and Processing Data, Scrape data from a website/API using Beautiful Soup/ Requests, Store data in CSV/JSON format, Perform initial Exploratory Data Analysis (EDA).
3. Data Cleaning and Pre-processing-Handle missing values using imputation techniques.
Remove duplicates and inconsistent data. Perform feature scaling (Normalization, Standardization).
4. Data Visualization using Matplotlib and Seaborn-Generate different plots, Use Seaborn for correlation analysis, create interactive visualizations using seaborn and matplotlib
5. Feature Engineering and Dimensionality Reduction-Apply One-Hot Encoding and Label Encoding, Perform Principal Component Analysis (PCA) for dimensionality reduction, Interpret the results of reduced features.
6. Statistical Analysis on a Dataset-Compute mean, median, variance, standard deviation. Analyse data distribution using histograms and KDE plots, Identify skewness and kurtosis.
7.Hypothesis Testing and Probability Distributions-Perform Z-test, T-test, and Chi-square test.
Generate and visualize probability distributions (Normal, Binomial, Poisson), Interpret hypothesis test results.</td></tr>
</table>

8. Correlation and Regression Analysis-Compute Pearson and Spearman correlation coefficients, Perform Simple and Multiple Linear Regression, Interpret regression coefficients and model assumptions.

9. Supervised Learning: Regression Models-Implement Linear and Polynomial Regression.
Evaluate model performance using R^2 score and RMSE, Visualize regression results.

10.Supervised Learning: Classification Models-Train Logistic Regression, Decision Tree, and SVM on a dataset, evaluate models using Confusion Matrix, Precision, Recall, F1-score, compare performance using ROC-AUC curves.

11.Unsupervised Learning: Clustering Techniques
Implement K-Means Clustering and determine optimal clusters using Elbow Method.

12. Handling Large Datasets with SQL and NoSQL-Write SQL queries (SELECT, GROUP BY, HAVING), Perform CRUD operations on NoSQL (MongoDB). Process large datasets using Pyspark.

Text Books

1. Introduction to Machine Learning with Python, Andreas C. Müller & Sarah Guido, O'Reilly Media
2. Python for Data Analysis, Wes McKinney, Third edition, O'Reilly Media
3. The Elements of Statistical Learning , Trevor Hastie, Robert Tibshirani, Second edition, Springer
4. Data Science for Business, Foster Provost, Tom Fawcett, First edition:, O'Reilly Media,
5. Practical Statistics for Data Scientists , Peter Bruce, Andrew Bruce. Second edition, O'Reilly Media

Reference Books

1.Data Science from Scratch : First Principles with Python, Joel Grus , 2nd edition, O'Reilly Media

2. Machine Learning, Tom M. Mitchell. First Edition, McGraw-Hill Education

3.Pattern Recognition & Machine Learning , Christopher Bishop, First Edition, Springer (Information Science & Statistics series)

4. Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, First edition, The MIT Press (Adaptive Computation and Machine Learning series)

5. Big Data: Principles and Best Practices, Marr Bernard. Wiley (as Big Data in Practice)

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
(An Autonomous Institute)
Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E15T: Programme Elective III - Natural Language Processing

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Tutorial	1 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

Natural Language Processing (NLP) is essentially a method for teaching computers how to understand human languages and interpret text. This course covers fundamentals of NLP including language morphology and language modelling, syntax analysis, semantic analysis, and the use of NLP for information retrieval. It also covers the most popular and effective current techniques, strategies, and toolkits for NLP to develop the various real-world NLP applications.

Course Prerequisite:

Basic mathematics, algorithms and programming skills, Theory of computation and parsing.

Course Objectives:

1. Learn the fundamentals of Natural Language Processing
2. Learn the methods to form words and categorize them.
3. Learn methods to model languages.
4. Analyse and find different algorithms to be applied for generating solutions to real world problems.

Course Outcomes:

After completing the course, students will be able to
1. Demonstrate the fundamentals of Natural Language Processing.
2. Analyse how the words are formed morphologically and how they are related to each other.
3. Develop strategies for language modelling, syntax and semantic analysis.
4. Design and implement and analyse the Natural Language Processing algorithms for real word applications

Unit – I	**Natural Language Processing**	**8 Hours**

Introduction to NLP, Machine Learning and NLP, why NLP is hard? Programming languages Vs Natural Languages, Are natural languages regular? Finite automata for NLP, Stages of NLP, challenges (Open Problems) in NLP. Basics of Text Processing: Tokenization, Stemming, Lemmatization, Part of Speech Tagging

<table>
<tr><td><b>Unit – II</b></td><td><b>Syntax and semantic Analysis</b></td><td><b>6 Hours</b></td></tr>
<tr><td colspan="3"><b>Morphological Analysis:</b> What is Morphology? Types of Morphemes, Inflectional morphology & Derivational morphology, Phonetics, HMM, Morphological parsing with Finite State Transducers (FSD. Syntactic Analysis: Syntactic Representations of Natural Language, Parsing Algorithms, Probabilistic context-free grammars and Statistical parsing.
<b>Lexical Semantic,</b> Relations among lexemes & their senses, Homonymy, Polysemy, Synonymy, Hyponymy, Word-Net, Word Sense Disambiguation (WSD), Dictionary based approach, Latent Semantic Analysis.</td></tr>
<tr><td><b>Unit – III</b></td><td><b>Language Modelling</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Probabilistic language modelling, Markov models, Generative models of language, Log-Liner Models, Graph-based Models. N-gram Models: Simple n-gram models, Estimation parameters and smoothing, evaluating language models.
<b>Word Embeddings/Vector Semantics:</b> Bag-of-words, TFIDF, word2vec, doc2vec, Contextualized representations, transformers (BERT, GPT), attention mechanisms, and pre-trained models,
<b>Topic Modelling:</b> Latent Dirichlet Allocation (LDA), Latent Semantic Analysis, Non –Negative Matrix Factorization</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Information Retrieval using NLP</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Information Retrieval: Introduction, Vector Space Model.
Named Entity Recognition: NER System building process, Evaluating NER System. Entity Extraction, Relation Extraction, Reference Resolution, Co reference resolution, Cross Lingua information retrieval (CLIR). Ethics and bias in NLP (e.g., fairness, privacy concerns), evaluation metrics (BLEU, ROUGE, F1 for NER) for practical implementation.</td></tr>
<tr><td><b>Unit – V</b></td><td><b>NLP Tools and Techniques</b></td><td><b>8 Hours</b></td></tr>
<tr><td colspan="3">Prominent NLP libraries: Natural Language Toolkit (NLTK), spaCy, TextBlob, Gensim. Linguistic Resources: Lexical Knowledge Networks, WordNets, Indian Language WordNet, (IndoWordnet), VerbNets, PropBank, Treebanks, Universal Dependency Tree banks.
Word Sense Disambiguation: Walker's algorithm, WordNets for Word Sense Disambiguation.</td></tr>
<tr><td><b>Unit – VI</b></td><td><b>Applications of NLP</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Machine Translation: Rule based techniques, Statistical Machine Translation (SMT), Cross Lingual Translation. Sentiment Analysis, Question Answering, Text Entailment, Discourse Processing, Dialog and Conversational Agents, Natural Language Generation, question answering with transformers, zero-shot learning, and multilingual NLP models.</td></tr>
<tr><td colspan="3">ICA shall include at least 8 of the following assignments. The assignment’s objective should align with course’s outcomes and focus on higher order bloom’s cognitive levels.
1. Perform Exploring and processing of text data.
2. Perform language modelling using one-hot encoding, N-grams, count vectorizer, TFID vectorizer and word embedding techniques.
3. Study and implementation of Noun-phrase extraction, text-similarity, part of speech tagging, chunking and named entity recognition.</td></tr>
</table>

4. Write a simple English to French translation algorithm using pre-computed word embeddings and locality sensitive hashing to relate words via approximate k-nearest neighbour’s search.
5. Study and implementation of sentimental analysis
6. Study and implementation of SPAM-HAM email classification
7. Write a better auto-complete algorithm using an N-gram language model
8. Study and implementation of text to speech and speech to text conversion
9. Study of Stanford Parser and POS Tagger. https://nlp.stanford.eclu/software/lex-parser.html
10. Study of language modelling for Indian languages
11. Study and implementation of multi-class classification using NLP and Naïve Bayes classifier
12. Study and implementation of text summarization and document clustering.

Text Books

1. Natural Language Processing, Dr. Smita M. Chaudhari, Devika A. Verma,Nitin N. Sakhare Nirali Prakashan,Pune.
2. Natural Language Understanding, Allen, James, Second Edition, Benjamin/Cumming,

Reference Books

1. Natural Language Processing with Python. Bird, S., Klein, E., Loper, Sebastopol, O'Reilly Media.
2. Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python, Akshay Kulkarni, Adarsha Shivananda , Apress, Bengolore.
3. Linguistics, An Introduction, Radford, Andrew, Cambridge University Press, 1999.
4. Statistical Language Learning, Charniack, Eugene , MIT Press, 1993.
5. Natural Language Processing Succinctly, Joseph D. Booth Foreword by Daniel Jebaraj , Succinctly EBook series, SyncFusion.
6. Github link for practical: https://github.com/kb1907/Natural-Language-Processing-Specialization

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E25T: Programme Elective III - Blockchain Technology

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Tutorial	1 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

The course is designed to introduce students to the concept of Blockchain and explain the fundamentals of Blockchain and its implementation. Students will be learning the importance of consensus in transactions, how transactions are stored on Blockchain, the history of Bitcoin and how it's used. Students will be introducing to Ethereum development platform and develop and deploy smart contracts from web and console.

Course Prerequisite:

The students should have a good knowledge of distributed systems, networking, cryptography, data structures, basic knowledge of Linux commands.

Course Objectives:

1. Understand how blockchain systems (Bitcoin and Ethereum) work,
2. Design, build, and deploy smart contracts and distributed applications,
3. Integrate ideas from blockchain technology into their own projects.

Course Outcomes:

1. After completing the course, students will be able to
2. Explain the fundamentals, decentralization and cryptographic functions of Blockchain Technology
3. Interpret the cryptographic hash functions & distributed consensus algorithms
4. Analyse the hash function, bitcoin network and payments.
5. Design, build, and deploy a smart contract, Hyperledger in distributed application.

Unit – I	**Introduction**	**8 Hours**

Introduction to Blockchain and Distributed Ledger, Definition of Blockchain and Distributed Ledger Technology (DLT) Core Blockchain Properties: Decentralization, Transparency, Immutability, and Security Blockchain Structure: Blocks, Transactions, Hashing, Consensus Mechanisms, Blockchain Applications: Finance (DeFi, CBDCs), Types of Blockchain: Public, Private, Consortium, Hybrid Blockchains, When to Use and When Not to Use Blockchain: Limitations, Scalability Challenges, and Ethical Concerns.

<table>
<tr><td><b>Unit – II</b></td><td><b>Decentralization and Symmetric Cryptography</b></td><td><b>8 Hours</b></td></tr>
<tr><td colspan="3">Methods of Decentralization: Full vs. Partial Decentralization Routes to Decentralization: Tokenization, DAO (Decentralized Autonomous Organizations), Interoperability Blockchain and Full Ecosystem Decentralization: Web3, Decentralized Finance (DeFi), NFTs, Layer 2 Solutions, Smart Contracts: Definition, Benefits, Risks, Automation in Blockchain, Platforms for Decentralization: Ethereum, Solana, Polkadot, Avalanche, Cosmos Fundamentals of Cryptography: Confidentiality, Integrity, Authentication, Encryption Methods: Stream & Block Ciphers, Advanced Encryption Standard (AES): Structure and Working</td></tr>
<tr><td><b>Unit – III</b></td><td><b>Asymmetric Cryptography</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Cryptographic Hash Functions: Properties and Role in Blockchain, Secure Hash Algorithms (SHA-256, SHA-3), Merkle Trees & Patricia Trees: Importance in Blockchain Storage and Validation, Distributed Hash Tables (DHTs): Decentralized Data Lookup, Digital Signatures and Security: RSA and Elliptic Curve Digital Signature Algorithm (ECDSA), Zero-Knowledge Proofs & zk-SNARKs: Privacy in Blockchain Transactions, Financial Markets and Trading in Blockchain: Algorithmic Trading, High-Frequency Trading (HFT), Automated Market Makers (AMMs), Decentralized Exchanges (DEXs) vs. Centralized Exchanges (CEXs)</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Bitcoin Network and Payments</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Bitcoin Fundamentals: History, Whitepaper, Market Adoption, Digital Keys and Addresses: Public & Private Key Management, Bitcoin Transactions: UTXO Model, Transaction Validation, Structure of a Bitcoin Block: Timestamp, Nonce, Hash, Merkle Root, Bitcoin Mining: Proof-of-Work (PoW), Mining Pools, 51% Attack Risk, Bitcoin Network and Wallets: Hot Wallets, Cold Wallets, Custodial vs. Non-Custodial Wallets, Bitcoin Payment Innovations: Lightning Network, Layer 2 Scaling, Alternative Consensus, Mechanisms: Proof of Storage, Proof of Stake (PoS), Proof of Deposit, Proof of Burn, Proof of Activity, Delegated Proof of Stake (DPoS).</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Smart Contracts and Ethereum</b></td><td><b>8 Hours</b></td></tr>
<tr><td colspan="3">Smart Contracts: Purpose, Automation, Security Risks, Types of Smart Contracts: Deterministic vs. Non-Deterministic, Ethereum Introduction: Blockchain Layers, Gas Fees, Bitcoin vs. Ethereum Architecture: Account Model vs. UTXO Model, Ethereum Virtual Machine (EVM): Execution and Security, Ethereum Consensus Mechanism: Proof of Stake (PoS), Ethereum 2.0, Ethereum Development: Solidity Language, Hardhat & Truffle Frameworks, Developing and Executing Smart Contracts, State and Data Structures in Ethereum, Layer 2 Scaling Solutions: Rollups, Plasma, Sidechains</td></tr>
<tr><td><b>Unit – VI</b></td><td><b>Hyperledger</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">Hyperledger Introduction: Permissioned Blockchains, Hyperledger Stack: Fabric, Sawtooth, Indy, Burrow, Iroha, Multi-Chain Blockchain Networks: Interoperability and Cross-Chain Communication, Innovation in Hyperledger: Private Transactions, Chain code, Smart Contract Execution, Hyperledger as a Protocol: Governance Models, Business Use Cases.</td></tr>
<tr><td colspan="3">ICA consists of minimum 8 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:

1. Creating Merkle tree
2. Creation of Block</td></tr>
</table>

3. Block chain Implementation Programming code
4. Creating ERC20 token
5. Java code to implement blockchain in Merkle Trees
6. Java Code to implement Mining using block chain
7. Java Code to implement peer-to-peer using block chain
8. Creating a Crypto-currency Wallet

Text Books

1. Mastering Blockchain Distributed ledger technology, decentralization, smart contracts, Imran Bashir, Packt Publishing Ltd.

Reference Books

1. ETHEREUM: A Secure Decentralized Transaction Ledger, DR. Gavin Wood, Yellow paper.2014.
2. A survey of attacks on Ethereum smart contracts , Nicola Atzei, Massimo Bartoletti & Tiziana Cimoli.

e-Resources

1. https://nptel.ac.in/courses/106/104/106104220/
2. https://www.investopedia.com/terms/b/blockchain.asp

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6E35T: Programme Elective III -Tools for Data Science

Teaching Scheme		**Examination Scheme**	
Lectures	3 Hours/week	**ESE**	60 Marks
Tutorial	1 Hours/week	**ISE**	40 Marks
Credits	4	**ICA**	25 Marks

Introduction:

This course provides an in-depth understanding of various tools used in the Data Science lifecycle, covering data handling, processing, visualization, machine learning, and deployment.

Students will gain hands-on experience with Python, R, SQL, Big Data frameworks (Hadoop, Spark), and cloud-based AI platforms (AWS, Google Cloud, Azure) to build and deploy data-driven solutions.

The course bridges the gap between theoretical concepts and practical applications, helping students develop industry-ready skills in data manipulation, analytics, and machine learning model deployment.

Course Prerequisite:

Basic programming knowledge in Python or R, including variables, loops, functions, and data structures.

Fundamental understanding of databases, including SQL, MySQL, PostgreSQL & MongoDB.

Basic knowledge of statistics and machine learning concepts, such as data pre-processing, regression, and classification.

Course Objectives:

1. Introduce students to various tools and technologies used in the Data Science pipeline.
2. Provide hands-on experience with Python, R, Jupyter, SQL, and cloud-based data science platforms.
3. Enable students to work with Big Data, Machine Learning, and AI tools for real-world problem-solving.
4. Familiarize students with data visualization, data pre-processing, and model deployment techniques.

Course Outcomes:

After completing the course, students will be able to

1. Understand the role and functionality of different data science tools.
2. Perform data manipulation and visualization using Python, R, and SQL.
3. Work with Big Data tools such as Hadoop, Spark, and cloud platforms.
4. Develop and deploy machine learning models using industry-standard tools.

Unit – I	**Introduction to Data Science Tools**	**05 Hours**

Overview of Data Science Lifecycle and tools used at each stage, Introduction to programming tools: Python, R, and Julia for Data Science, Introduction to Jupyter Notebook, Google Colab, and RStudio. Data manipulation tools: Pandas, NumPy, and Dplyr (R), Version control with Git and GitHub for data science projects.

<table>
<tr><td><b>Unit – II</b></td><td><b>Data Handling and Processing Tools</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Working with Databases: SQL & NoSQL (MySQL, PostgreSQL, MongoDB), Using Apache Hadoop and Apache Spark for big data processing, Hands-on with Pyspark for distributed data handling, Data extraction, transformation, and loading (ETL) techniques, Cloud-based storage and data pipelines (AWS S3, Google BigQuery).</td></tr>
<tr><td><b>Unit – III</b></td><td><b>Data Visualization and Exploration</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Data visualization tools: Matplotlib, Seaborn, and Plotly in Python, Using Tableau or Power BI for interactive data visualization, building dashboards and reports for business intelligence, Data exploration with Pandas Profiling, Real-time visualization using Dash and Streamlit.</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Machine Learning and AI Tools</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Introduction to Scikit-Learn, TensorFlow, and Keras for ML and Deep Learning, Auto ML tools: Google AutoML, H2O.ai, and ML flow, Model evaluation and hyperparameter tuning using GridSearchCV , Hands-on with Google Colab Pro & Vertex AI for cloud-based ML.
Deploying ML models with Flask, FastAPI, and Streamlit.</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Big Data, Cloud, and Deployment Tools</b></td><td><b>07 Hours</b></td></tr>
<tr><td colspan="3">Cloud-based Data Science: AWS, Google Cloud, and Microsoft Azure AI, working with Google BigQuery and AWS Redshift, Introduction to Docker and Kubernetes for scalable model deployment, Serverless computing with AWS Lambda and Google Cloud Functions. Case studies on end-to-end ML pipelines in production.</td></tr>
<tr><td colspan="3">ICA consists of minimum 10 assignments requiring students to develop machine learning applications for real world problem/use- case/scenario based on the following topics:

1. Setting up Jupyter Notebook and Google Colab for Data Science.
2.Performing data manipulation with Pandas and NumPy.
3 Executing SQL queries for data retrieval and analysis.
4.Handling Big Data using Apache Spark and PySpark.
5.Creating visualizations using Matplotlib, Seaborn, and Tableau.
6.Building a machine learning model using Scikit-Learn.
7.Deploying a trained ML model using Flask/Streamlit.
8.Implementing an end-to-end data science pipeline on Google Cloud.</td></tr>
<tr><td colspan="3"><b>Text Books</b></td></tr>
<tr><td colspan="3">1. Python for Data Science Handbook, Jake VanderPlas
2.Data Science for Business, Foster Provost & Tom Fawcett
3.Machine Learning with PyTorch and Scikit-Learn, Sebastian Raschka.</td></tr>
<tr><td colspan="3"><b>Reference Books</b></td></tr>
<tr><td colspan="3">1.Hadoop: The Definitive Guide, Tom White
2.Practical Statistics for Data Scientists, Peter Bruce & Andrew Bruce
3.Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron</td></tr>
</table>

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR

(An Autonomous Institute)

Third Year B.Tech. (Information Technology), Semester-VI

23ITU6VS7P: Open-Source Technology-Web UI/UX Technology

Teaching Scheme		**Examination Scheme**	
Lectures	1 Hour/week	**ESE**	-
Practical	2 Hours/week	**ISE**	-
Credits	2	**ICA**	25 Marks
		POE	50 Marks

Introduction:

Web Technology alludes to the different tools and techniques that are used during the process of communication between various sorts of gadgets over the web or internet. This course includes different technologies which help to build front end and back end for websites and web applications.

Course Prerequisite:

Fundamental programming skills, logical reasoning abilities and a basic acquaintance with internet and web browsing mechanisms is required.

Course Objectives:

1. Inculcate skills necessary to design, develop and style a web-based user interface
2. Develop ability to identify appropriate client/server-side scripting web technologies suitable for a given use case.
3. Develop skills required to create light weight, efficient and scalable Mobile browser-based APIs web applications

Course Outcomes:

After completing the course, students will be able to

1. Design web pages using HTML, CSS and JavaScript
2. Analyze client/server-side scripting technologies to meet requirements of web application and choose an appropriate one.
3. Develop web and mobile application using client/server-side scripting technologies for a given problem.

Unit – I	**Basics of Web Technology and HTML5/CSS3**	**2 Hours**

Introduction to Web Technology: History and Evolution of the Web, How the Internet Works, Understanding Web Servers and Clients, HTML5: HTML Basics and Structure, Semantic Elements in HTML5, Forms and Input Types, Multimedia: Audio, Video, Canvas, CSS3: CSS Basics and Syntax, Selectors and Specificity, Box Model and Layout Techniques, Flexbox and Grid Layout, Animations and Transitions

<table>
<tr><td><b>Unit – II</b></td><td><b>JavaScript Basics and Advanced JavaScript</b></td><td><b>3 Hours</b></td></tr>
<tr><td colspan="3">JavaScript Basics: Introduction to JavaScript, Variables, Data Types, and Operators, Control Structures (Loops, Conditionals), Functions and Scope, Advanced JavaScript: ES6+ Features: Arrow Functions, Let/Const, Template Literals, Asynchronous JavaScript: Promises, Async/Await, Error Handling and Debugging, JavaScript Modules and Imports/Exports</td></tr>
<tr><td><b>Unit – III</b></td><td><b>Front-End Frameworks: React.js</b></td><td><b>3 Hours</b></td></tr>
<tr><td colspan="3">Introduction to React.js: Overview and Benefits of React, React Components and JSX, State and Props, Event Handling, Conditional Rendering and Lists, Advanced React Concepts: Lifecycle Methods, React Hooks (useState, useEffect, etc.), Context API for State Management, React Router for Single Page Applications</td></tr>
<tr><td><b>Unit – IV</b></td><td><b>Mobile Development with React Native</b></td><td><b>4 Hours</b></td></tr>
<tr><td colspan="3">Introduction to React Native: Overview of React Native and JavaScript/TypeScript, Setting Up the React Native Development Environment, React Native Components and State Management, Building Layouts with Flexbox in React Native, Handling User Input and Gestures, Advanced React Native Concepts, Navigation and Routing (React Navigation),Networking and API Integration (Fetch, Axios),State Management Solutions (Context API, Redux, Recoil),Animations in React Native (Reanimated, Lottie),Testing and Debugging React Native Apps</td></tr>
<tr><td><b>Unit – V</b></td><td><b>Basics of Back-End Development and Node.js</b></td><td><b>3 Hours</b></td></tr>
<tr><td colspan="3">Basics of Back-End Development: Overview of Server-Side Development, Understanding RESTful APIs, Authentication and Authorization Concepts, Node.js and Express.js: Introduction to Node.js and npm, Building a Simple Server with Node.js, Express.js for Routing and Middleware, Connecting to Databases (MongoDB, MySQL), Building and Consuming APIs</td></tr>
<tr><td colspan="3">ICA consists of minimum 10 assignments requiring students to develop applications for real world problem/use- case/scenario based on the following topics:
1. Basic Static Website: Design and Develop a Static Website Using HTML5 and CSS3.
2. Dynamic Website: Develop a Dynamic Website with Enhanced Interactivity Using JavaScript.
3. Single Page Application: Develop a SPA Using React.js and Manage State with Hooks/Redux.
4. Mobile Application: Develop a Mobile Application Using React Native.
5. Full-Stack Application: Develop a Full-Stack Application Using Node.js, Express.js, and a Database.</td></tr>
<tr><td colspan="3"><b>Text Books/Reference Books</b></td></tr>
<tr><td colspan="3">1. Responsive Web Design with HTML5 and CSS3, Ben Frain, Packt Publication.
2. JavaScript and jQuery: Interactive Front-End Web Development, Jon Duckett.</td></tr>
<tr><td colspan="3"><b>e-Resources</b></td></tr>
<tr><td colspan="3">1. Official documentation of Node.js: https://nodejs.org/en/
2. Official documentation of ReactJS: https://reactjs.org/docs/getting-started.html\
3. Official documentation of Flutter: https://docs.flutter.dev/</td></tr>
</table>