Course Code:

DATA STRUCTURES AND ALGORITHMS FOR AI

(Professional Core I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	End Examination Marks	:	60 Marks
Credits	:	3	End Exam Duration	:	3 Hours

Course Objectives:

1. To introduce Python constructs
2. To implement efficient data structures that improve the search performance
3. To demonstrate various algorithm models

Unit I – Python Primer and OOP

Python Interpreter, Objects in Python, Constructs of Python, Loops, Functions, Information Passing, Python's Built-In Functions, Simple Input and Output, Files, Exception Handling, Iterators and Generators, Conditional Expressions, Comprehension Syntax, Packing and Unpacking of Sequences, Scopes and Namespaces, Modules and the Import Statement.

Object-Oriented Programming: Class Definitions, Inheritance, Namespaces and Object-Orientation,Shallow and Deep Copying.

Unit II – Linear and Non-linear Data Structures

Linear Data Structures: The Stack Abstract Data Type, Simple Array-Based Stack Implementation. The Queue Abstract Data Type, Array-Based Queue Implementation, The Deque Abstract Data Type, Implementing a Deque with a Circular Array. Singly Linked Lists, Circularly Linked Lists. Implementing a Queue with a Circularly Linked List, Doubly Linked Lists. The Positional List ADT, Sorting a Positional List.

Non-Linear Data Structures: Trees, Binary Trees, Implementing Trees, Tree Traversal Algorithms, Tree Traversals in Python. Data Structures for Graphs, Edge List Structure, Adjacency List Structure, Adjacency Map Structure, Array-Based Sequences, Python's Sequence Types, Referential Arrays, Compact Arrays in Python, Python's List and Tuple Classes, Python's String Class.

Unit III – Multi-dimensional Search Trees

Splay Trees, Splaying, When to Splay, Python Implementation, Red-Black Trees, Python Implementation. Python's Built-In Sorting Functions.

Text Processing, Abundance of Digitized Text,Notations for Strings and the Python str Class. Brute Force, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm. Standard Tries, Compressed Tries, Suffix Trees.

Unit IV – Problem solving Agents and Algorithms

AI Problems, Searching for solutions, Uninformed search strategies, Informed search strategies (Heuristic) search strategies, Heuristic functions,

Local search algorithms and optimization problems, local search in continuous spaces, Searching with non-deterministic actions, Searching with partial observations, Online search agents and unknown environments.

Unit V – Adversarial search and Constraint Satisfaction search

Games, Optimal decisions in games, Alpha-Beta pruning, Imperfect Real-Time Decisions, Stochastic games, Partially Observable Games, State of the art game programs, Alternate Approaches,

Defining constraint satisfaction problems, constraint propagation: inference in CSPs, Backtracking search for CSPs, Local search for CSPs, The structure of CSP problems.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Master the Python language constructs.
CO 2 : Implement List, Tree based data structures and related applications.
CO 3 : Appreciate the capabilities of multi-dimensional search trees.
CO 4 : Solve real world problems related to search through different environments.
CO 5 : Implement adversarial search and CSP search techniques.

Textbooks:

1. Data Structures and Algorithms in Python, M T Goodrich, R Tamashia, M H Goldwasser, An Indian Adaptation, WILEY, 1st July 2021.
2. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, 4th Edition, Prentice Hall, 2020.

References:

1. Core Python Applications Programming, Wesley J Chun, Sunnyvale, California, 3rd Edition, Prentice Hall,2012.
2. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi, 1st Edition, Career Monk Publications,2015.
3. The Algorithm Design Manual, Skiena SS, Springer, 2013.

Course Code:

MACHINE LEARNING

(Professional Core II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To develop an understanding of the basic concepts of machine learning based system design.
2. To analyze about different types Supervised and unsupervised learning models.
3. To differentiate various Association Analysis Methods.
4. To study the reinforcement learning methods.

Unit I – Introduction and Concept Learning

Introduction to Machine Learning: Types of Learning, Well-posed learning problems, Designing a learning system, Issues in machine learning. Tools for Machine Learning: Spark, Python ML libraries, Weka, Concept learning: Introduction, A concept learning task, Concept learning as search. Find-S: Algorithm. Version spaces and the candidate elimination Algorithm and remarks. Inductive bias.

Unit II - Supervised Learning

Decision Tree Learning: Decision Tree Representation, Example Problems for Decision Tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Learning, Inductive Bias, Issues of Decision Tree Learning.

Computational Learning Theory: Probably learning an approximately correct hypothesis, Sample complexity for finite hypothesis spaces, Sample Complexity for Infinite Hypothesis Spaces, Mistake bound model of Learning.

Instance Based Learning: K-Nearest Neighborhood Learning, Distance weighted Nearest Neighbor Algorithm, Radial Basis Functions, Case Based Reasoning, Lazy and Eager learning, Support Vector Machines.

Unit III – Association Analysis

Basic Concepts: Frequent Item sets, Closed Item sets, and Association Rules. Frequent Itemset Mining Methods: Apriori Algorithm, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A Pattern-Growth Approach for Mining Frequent Itemsets, Mining Frequent Itemsets Using Vertical Data Format, Mining Closed and Max Patterns.

Pattern Evaluation Methods: Strong Rules Are Not Necessarily Interesting, From Association Analysis to Correlation Analysis, A Comparison of Pattern Evaluation Measures.

Pattern Mining: Constraint-Based Frequent Pattern Mining, Metarule-Guided Mining of Association Rules. Constraint-Based Pattern Generation: Pruning Pattern Space and Pruning Data Space, Mining High-Dimensional Data and Colossal Patterns, Mining Colossal Patterns by Pattern-Fusion, Mining Compressed or Approximate Patterns, Mining Compressed Patterns by Pattern Clustering, Extracting Redundancy-Aware Top-k Patterns.

Unit IV - Unsupervised Learning

Cluster Analysis: Basic Concepts and Methods, Cluster Analysis, Requirements, Basic Clustering Methods Partitioning Methods, k-Means, k-Medoids, Hierarchical Methods, Agglomerative versus Divisive Hierarchical Clustering. Distance Measures in Algorithmic Methods. BIRCH, Chameleon, Probabilistic Hierarchical Clustering. Density-Based Methods: DBSCAN, OPTICS, DENCLUE. Grid Based Methods: STING, CLIQUE.

Evaluation of Clustering: Assessing Clustering Tendency, Determining the Number of Clusters, Measuring Clustering Quality.

Unit V - Reinforcement Learning

Introduction: The Learning task, Q-Learning: Nondeterministic rewards and actions, The Q-function, An Algorithm for Learning Q, Example, Convergence, Experimentation Strategies, Updating Sequence.

Non-Deterministic Rewards and Actions: Temporal Difference Learning, Generalizing from Example, Relationship to Dynamic Programming.

Course Outcomes: After completion of course, students will be able to:

CO1 : Analyze the concepts of ML and design a ML based system.
CO2 : Analyze and apply suitable supervised learning methods and algorithms.
CO3 : Develop various learning models for association analysis.
CO4 : Analyze and apply suitable unsupervised learning algorithms and methods.
CO5 : Model a reward based strategy to evolve an optimal policy for solving exploration problems.

Textbooks:

1. Machine Learning – Tom M. Mitchell, McGraw Hill Education, First Edition, July 2017.
2. Data mining: Concepts and techniques, Jiawei Han, Jian Pei, Hanghang Tong, Morgan Kaufmann, Fourth Edition, July 2022.

References:

1. Artificial Intelligence and Machine Learning, Vinod Chandra S.S., Anand Hareendran S, 1st Edition, PHI Learning Pvt. Ltd, 2014.
2. Artificial Intelligence - A Modern Approach, Stuart Russel and Peter Norvig, 4th Edition, Pearson, 2020.
3. Reinforcement Learning - An Introduction. R. S. Sutton and A. G. Barto. MIT Press. 1998.

Course Code:

DISTRIBUTED DATABASES

(Professional Elective – I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To expose students to both the abstraction and details of file systems.
2. To provide students with contemporary knowledge in parallel and distributed computing along with Query Processing.
3. To focus on performance and flexibility issues related to systems design decisions.
4. Introduce a variety of methodologies and approaches for reasoning about concurrent and distributed programs.

Unit I: Introduction to DDBMS concepts

Introduction: Distributed Data processing, Distributed database system (DDBMS), Promises of DDBMSs, Complicating factors and Problem areas in DDBMSs, Overview of Relational DBMS Relational Database concepts, Normalization, Integrity rules, Relational Data Languages, Relational DBMS.

Unit II: Distributed DBMS Architecture

DBMS Standardization, Architectural models for Distributed DBMS, Distributed DBMS Architecture. Distributed Database Design: Alternative design Strategies, Distribution design issues, Fragmentation, Allocation. Semantic Data Control: View Management, Data security, Semantic Integrity Control.

Unit III: Overview of Query Processing

Query processing problem, Objectives of Query Processing, Complexity of Relational Algebra operations, characterization of Query processors, Layers of Query Processing. Introduction to Transaction Management: Definition of Transaction, Properties of transaction, types of transaction. Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanisms, locking bases concurrency control algorithms.

Unit IV: Parallel Database Systems

Database servers, Parallel architecture, Parallel DBMS techniques, Parallel execution problems, Parallel execution for hierarchical architecture.

Unit V: Distributed Object Database Management systems

Fundamental Object concepts and Object models, Object distribution design. Architectural issues, Object management, Distributed object storage, Object query processing, Transaction management. Database Interoperability: Database Integration, Query processing.

Course Outcomes: At the end of the course, the student should be able to:

CO 1	:	Understand the concepts and issues related to distributed systems.
CO 2	:	Understand DDB design and handle data fragmentation issues.
CO 3	:	Develop the programs for distributed environment and also query the DDB.
CO 4	:	Understand the concepts of Parallel Database systems.
CO 5	:	Manage performance, reliability and other issues while designing in distributed environment.

Textbooks:

1. Principles of Distributed Database Systems by M. Tamer Ozsu Patrick Valduriez, Springer, Fourth Edition, 18th December 2020.

References:

1. Distributed Databases principles and systems by Stefano Ceri, Giuseppe Pelagatti, Tata McGraw Hill India, January 1, 2011.

Course Code:

INTERNET OF THINGS

(Professional Elective I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	End Examination Marks	:	60 Marks
Credits	:	3	End Exam Duration	:	3 Hours

Course Objectives:

1. To study the fundamentals about IoT.
2. To study about IoT Access technologies.
3. To study the design methodology and different IoT hardware platforms.
4. To study the basics of IoT Data Analytics and supporting services.
5. To study about various IoT case studies and industrial applications.

Unit I Fundamentals of IoT

Evolution of Internet of Things, Enabling Technologies, M2M Communication, IoT World Forum (IoTWF) standardized architecture, Simplified IoT Architecture, Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.

Unit II – IoT Protocols

IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.11ah and Lora WAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks,6LoWPAN, Application Transport Methods: SCADA, Application Layer Protocols: CoAP and MQTT.

Unit III – IoT Physical Devices and End points

Introduction to Arduino, Arduino pin diagram, Arduino IDE, Controlling LED with Arduino, interfacing temperature sensor with Arduino. Introduction to Raspberry PI, Interfaces (serial, SPI, I2C). Programming Raspberry PI with Python - Controlling LED, Interfacing an LED and Switch, Interfacing a Light Sensor with Raspberry PI, Capturing Image using Raspberry PI. Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib, RPi.GPIO.

Unit IV – Data Analytics and supporting devices

Data Analytics: Introduction, Structured Versus Unstructured Data, Data in Motion versus Data at Rest, IoT Data Analytics Challenges, Data Acquiring, Organizing in IoT/M2M, Supporting Services: Computing Using a Cloud Platform for IoT/M2M Applications/Services, Everything as a service and Cloud Service Models.

Unit V – Case Studies/Industrial Applications

IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipment, Industry 4.0 concepts.

Course Outcomes: After completion of course, students will be able to:

CO 1: Understand the characteristics, protocols and communication models required for logical design of IoT.
CO 2: Understand the differences between IoT and M2M networks and configurations.
CO 3: Understand the hardware platforms for implementing and interfacing the IoT based board with different peripheral devices and serial communication devices.

CO 4: Integrate devices and develop an application that can communicate through IoT Cloud.
CO 5: Understand various case studies in IoT design and Security in IoT.

Textbooks:

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017.
2. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015.

References:

1. The Internet of Things: Enabling Technologies, Platforms and Use Cases, Pethuru Raj and Anupama C. Raman (CRC Press), 2017.
2. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012.
3. Getting started with Raspberry Pi, Richardson Matt, Shroff Publishers & Distributers Private Limited, 2nd Edition, 2016.

Course Code:

CLOUD COMPUTING AND WEB SERVICES

(Professional Elective I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the Cloud computing types of cloud, risks and applications.
2. To understand cloud computing architecture and virtualization.
3. To understand cloud computing, cloud management and models.
4. To learn and Understand Cloud Microservices and its architectural challenges.
5. To learn and understand the types of web services, architecture and benefits.

Unit I - Introduction to Cloud Computing

Online Social Networks and applications, Cloud introduction and overview, different clouds, risks, novel applications of cloud computing.

Unit II - Cloud Computing Architecture

Requirements, Introduction to Cloud computing architecture, On demand computing Virtualization at the infrastructure level, Security in Cloud computing environments, Virtualization, operating system level virtualization, CPU Virtualization, Container technologies, A discussion on Hypervisors storage, Virtualization Cloud Computing Defined, The SPI Framework for Cloud Computing, The Traditional Software model, The Cloud Services delivery Model.

Unit III - Cloud Deployment Models

Key Drivers to adopting the Cloud, The Impact of Cloud Computing on users, Governance in the Cloud, Barriers to Cloud Computing adoption in the enterprise

Portability and Interoperability issues, Cloud Management and a Programming Model Case Study, Popular Cloud Services

Unit IV - Cloud Microservices

Microservices: Origins, Microservices Architecture in a nutshell, Successful implementation of Microservices, Orchestration and Choreography, Microservices Migration Plan for Emergence, Breaking a Monolith into modules, Breaking modules into sub-modules, Establishing Microservices Architecture, Advantages and gains with Microservices

Architectural Challenges: Identifying and classifying Challenges, Orchestrated Domains, Choreographed Domains, Challenges in exchanging data between Micro-services, Problems with Sharding.

Unit V - Web services

Service Oriented Architecture – SOA concepts, Key Service Characteristics, Introduction to Web Services– The definition of web services, The basic operational model of Web Services, Core fundamentals of SOAP. Describing Web Services – Web Services life cycle, anatomy of WSDL.

Introduction to Axis– Installing axis web service framework, deploying a java web service on axis.

REST based Web Services

Overview of Representational State Transfer (ReST) – URIs, Statelessness, Resource Oriented Architecture, ReST Benefits, and Limitations.

Course Outcomes: At the end of the course, the student should be able to

CO1 : Make use of cloud concepts, types of cloud, risks, and applications.
CO2 : Utilize Cloud computing architecture in cloud services and virtualization
CO3 : Perceive the importance of Cloud interoperability, portability, management, model, and popular cloud services.
CO4 : Develop applications with the use of cloud micro-services.
CO5 : Examine and apply the web services architecture, design, and its benefits to real-time applications.

Textbooks:

1. Cloud Computing Explained: Implementation Handbook for Enterprises, John Rhoton, Publication Date: November2, 2009.
2. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India, 2008.

References:

1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), TimMather, O'Reilly Media, September 2009.
2. Rajasekharaiah, C. Cloud-Based Microservices: Techniques, Challenges, and Solutions, Apress, 2020.
3. Restful Web Services, Leonard Richardson, Sam Ruby, O Reilly.
4. Cloud Computing, Raj Kumar Buyya, James Broberg, Andrzej Goscinski, Wiley, 2013.

Course Code:

KNOWLEDGE REPRESENTATION AND REASONING IN ARTIFICIAL INTELLIGENCE

(Professional Core II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To develop an understanding of the basic concepts of Artificial Intelligence.
2. To impart knowledge representation, reasoning mechanisms.
3. To master the necessary skill set for handling uncertainty associated with real time systems.
4. To resolve decision making rules and their applications in AI.

UNIT I - Introduction to Artificial Intelligence

Introduction: What Is AI?, The Foundations of Artificial Intelligence, The History of Artificial Intelligence, The State of the Art.

Intelligent agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents, Problem-Solving Agents, Example Problems.

UNIT II – Knowledge Representation and Reasoning

Logical Agents: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic: A Very Simple Logic, Propositional Theorem Proving, Effective Propositional Model Checking.

First-Order Logic: Representation Revisited, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and First-Order Inference, Forward Chaining, Backward Chaining, Resolution.

UNIT III –Quantifying Uncertainty and Probabilistic Reasoning

Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use, Naive Bayes Models, Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Exact Inference in Bayesian Networks, Approximate Inference for Bayesian Networks.

UNIT IV – Probabilistic Reasoning over Time

Time and Uncertainty, Inference in Temporal Models, Hidden Markov Models, Kalman Filters, Dynamic Bayesian Networks.

UNIT V – Decision Making

Making Simple Decisions: Combining Beliefs and Desires under Uncertainty, the Basis of Utility Theory, Utility Functions, Multi-attribute Utility Functions and Decision Networks, The Value of Information, Unknown Preferences.

Making Complex Decisions: Sequential Decision Problems, Value Iteration, Policy Iteration, Partially Observable MDPs (Markov Decision Processes), Decisions with Multiple Agents.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Identify the scope for agent-based engineering solutions using Artificial Intelligence.
CO 2 : Apply knowledge representation and reasoning for solving problems using logic concepts.
CO 3 : Apply probability theory to represent uncertainty and use probabilistic reasoning for solving AI problems.
CO 4 : Use probabilistic reasoning over time.
CO 5 : Make decisions in uncertain environments Artificial Intelligence.

Textbooks:

1. Artificial Intelligence a Modern Approach, Stuart Russel and Peter Norvig, 4th Edition, Pearson, 2020.
2. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivashankar B. Nair, 3rd Edition, McGraw-Hill, 2017.

References:

1. Artificial Intelligence, Saroj Kaushik, 1st Edition, Cengage Learning India, 2011.
2. Artificial Intelligence: A new Synthesis, J. Nilsson, Elsevier Publishers, 1998.

Course Code:

BLOCKCHAIN TECHNOLOGIES

(Professional Elective II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. This course aims to provide conceptual understanding of the function of Blockchains as a method of securing distributed ledgers, how consensus on their contents is achieved, and the new applications that they enable.
2. It covers the technological underpinnings of blockchain operations as distributed data structures and decision-making systems, their functionality and different architecture types.
3. It provides a critical evaluation of existing — "smart contract" capabilities and platforms, and
   examines their future directions, opportunities, risks and challenges.

Unit I - Introduction to Blockchain

The growth of Blockchain technology, Distributed systems, The history of Blockchain and Bitcoin, Types of Block chain, Consensus, CAP theorem and Blockchain

Decentralization: Decentralization using block chain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized Organizations, Platforms for decentralization.

Unit II – Bitcoins

Introducing Bitcoin, Digital keys and addresses, Transactions, Blockchain: The structure of a block, Mining.

Bitcoin Network and Payments: The Bit coin network, Wallets, Bitcoin payments: Innovation in Bitcoin, Bitcoin Clients and APIs: Bit coin installation, Alternative Coins, Bitcoin limitations.

Unit III - Smart Contracts

Smart Contracts History, Definition, Ricardian contracts, Introduction to Ethereum, Components of the Ethereum ecosystem, Further Ethereum, Programming languages.

Unit IV - Ethereum Development Environment

Test networks, setting up a private net, Starting up the private network, Development Tools and Frameworks, Compilers, Solidity compiler (solc), Installation on Linux, Installation on macOS, Integrated Development Environments (IDEs)

Solidity language: Solidity language, Types, Value types, Literals, Enums, Function types, Reference types, Global variables, Control structures, Layout of a Solidity.

Unit V - Hyper Ledger

Projects under Hyperledger, Hyperledger as a protocol, The reference architecture, Requirements and design goals of Hyperledger Fabric, Hyperledger Fabric, Membership services, Blockchain services, Consensus services, Distributed ledger

Beyond Cryptocurrency: applications of blockchain in cyber security, integrity of information, E-Governance and other contract enforcement mechanisms, Limitations of blockchain as a technology, and myths vs. reality of blockchain technology.

Course Outcomes: At the end of the course, the student will be able to:

CO1 : Understand the structure of a blockchain and why/when it is better than a simple distributed database.
CO2 : Evaluate the setting where a blockchain based structure may be applied, its potential and its limitations.
CO3 : Understand what constitutes a —smart contract, what are its legal implications and what it can and cannot do, now and in the near future.
CO4 : Attain awareness of the new challenges that exist in monetizing businesses around blockchains and smart contracts.
CO5 : Describe and understand the differences between the most prominent blockchain structures and permissioned blockchain service providers, as well as rising alliances and networks.

Textbooks:

1. Mastering Blockchain, Imran Bashir, Second Edition, Packt Publishing, March 2018.
2. Mastering Bitcoin Programming the Open Blockchain, Andreas M. Antonopoulos, 2nd Edition, "O'Reilly Media, Inc.", June, 2017.

References:

1. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric
2. Publisher:https://www.packtpub.com/big-data-and-business-intelligence/hands-blockchain-hyperledger
3. Public github repository with code samples: https://github.com/HyperledgerHandsOn/trade-finance-logistics

Course Code:

STATISTICAL MACHINE TRANSLATION

(Professional Elective – II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	End Examination Marks	:	60 Marks
Credits	:	3	End Exam Duration	:	3 Hours

Course Objectives:

1. To introduce the field of Machine Translation with a focus on statistical approaches.
2. To learn three paradigms of machine translation namely word-based translation, phrase-based translation, and syntax-based translation.
3. To experience with building translation systems and working with real-world data.

Unit I - Overview of machine translation

The statistical approach to MT, Word-based alignment and translation: IBM word alignment models, n-gram language models. Absolute discounting and KneserNey smoothing, n-gram language models continued, very large language models.

Unit II - Phrase based translation and discriminative training

Phrase-based MT. Why do we need phrases. Relationship to EBMT. Phrase extraction. Estimating phrase translation probabilities and the problem of overfitting. From the noisy channel to linear models, Phrase features, Phrase reordering models, Phrase based decoding, K- best lists.

Unit III – Entropy and Interlude concepts

Maximum entropy. Minimum error-rate training, Perceptron, max-margin methods, System combination.

Interlude: Subword translation, Transliteration. Integrating traditional translation rules, Integrating morphology into translation, Decoding with lattices for morphology and word segmentation.

Unit IV – Syntax based translation

Syntax based translation, Hierarchical and syntax-based MT, Why do we need syntax, Synchronous context-free grammars and TSGs.

Extracting synchronous CFGs and TSGs from parallel data. Estimating rule probabilities and the problem of overfitting. Extracting synchronous TSGs from tree-tree data and the problem of non-isomorphism.

Unit V – Decoding

CKY decoding, CKY with an n-gram language model. More CKY decoding: Binarization. k-best lists. Decoding with lattices, Source-side tree decoding. Target-side left-to-right decoding, Syntax-based language models, Beyond synchronous CFGs and TSG, Towards semantics based translation.

Course Outcomes: At the end of the course, student will be able to:

CO 1 : Describe and critically discuss the architecture of machine translation systems.
CO 2 : Handle basic tools for training and applying machine translation systems.

CO 3 : Compare different types of machine translation strategies, such as rule-based, statistical and neural machine translation.
CO 4 : Evaluate machine translation output using automatic and manual methods and explain possible causes of translation errors.
CO 5 : Critically read and summarize a scientific works in the field of machine translation.

Textbooks:

1. Statistical Machine Translation by Philipp Koehn, Cambridge University Press, 1st Edition, 17 December 2009.

Course Code:

RESEARCH METHODOLOGY AND IPR

(Common to all)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Understand the significance of Research Methodology and learn systematic approach in selecting research problem.
2. To know the importance of originality in Research work and check for plagiarism.
3. To learn approaches for effective presentation both oral & written mode of Research Synopsis and Research Report.
4. To understand the legal means of protecting intellectual work and their implications.
5. To understand the rudiments of patenting procedure and the laws governing it.

Unit I – Research Problem

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations.

Unit II – Research Ethics

Effective literature studies approaches, analysis Plagiarism, Research ethics.

Unit III – Research Proposal

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee.

Unit IV – Intellectual Property

Nature of Intellectual Property, Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

Unit V – Patent Rights

Scope of Patent Rights, Licensing and transfer of technology, Patent information and databases, Geographical Indications - New Developments in IPR - Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

Course Outcomes: At the end of the course, Student will be able to:

CO 1 : Understand research problem formulation and analyze research related information.
CO 2 : Follow research ethics.
CO 3 : Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
CO 4 : Understanding importance of intellectual property rights.
CO 5 : Understand the importance of patent rights and developments in IPR.

Textbooks:

1. Research Methodology: An Introduction for Science & Engineering Students, Wayne Goddard and Stuart Melville.
2. Research Methodology: A Step by Step Guide for beginners, Ranjit Kumar, 2nd Edition.
3. Intellectual Property Rights Law in India, T Ramappa, 2016.

References:

1. Resisting Intellectual Property, Halbert, Taylor & Francis Ltd ,2007.
2. Industrial Design, Mayell, McGraw Hill, 1992.
3. Intellectual Property in New Technological Age, Robert P. Merges, Peter S. Menell and Mark A. Lemley, 2016.
4. Introduction to Design, Asimov, Prentice Hall, 1962.
5. Product Design, Niebel, McGraw Hill, 1974.

Course Code:

ALGORITHMS IN AI LAB

(Professional Core)

Instruction	:	4 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To master the constructs of Python
2. To insist upon the necessity of efficient implementation of data structure
3. To model the real-world problems using ADTs and algorithms

LIST OF EXPERIMENTS

1. Write a python program that takes two arrays a and b of length n storing integers, and returns the dot product of a and b. That is, it returns an array c of length n such that c[i] = a[i] · b[i], for i = 0... n−1.

2. Write a python program to implement Towers of Hanoi Puzzle.

3. Write a python program to implement the memory allocation in a dynamic environmentusing Dynamic Array.

4. Write a python program to determine if a string s has more vowels then consonants.

5. Write a python program to implement Parentheses Matching and HTML Tags using array-based stack.

6. Write a python program to implement insertion sort algorithm using arrays.

7. Write a python program to implement Tic- Toc- Toe game using two dimensional arrays.

8. Write a python program to implement Stack Abstract Data Type using python list.

9. Write a python program to implement Round Robin scheduler using circular linked list.

10. Write a python program to implement Binary Tree Traversal Techniques for Tic-Tac-Toegame playing using breadth-first traversal order.

11. Write a python program to implement Euler Tours traversal to find order of cities in atraveling sales man problem.

12. Write a python program to evaluate an arithmetic expression using expression tree.

13. Write a python program to implement KMP (Knuth-Morris-Pratt) Algorithm to match thegiven pattern of characters.

14. Write a python program to represent a priority queue environment and agent to supportthe following searching strategies:
    a. A * Search
    b. Best-first search
    c. Alpha-Beta cut off search

15. Write a python program to demonstrate the minimax search procedure to the magicsquare problem.

16. Implementation of DFS (Depth First Search).

17. Implementation of AO* search algorithm.

18. Implementation of problem-solving using Hill climbing.

19. Implementation of Monte Carlo Tree search.

20. Implementation of Constraint Satisfaction Problem.

Course Outcomes: At the end of the course, student will be able to:

CO 1 : Master the Python ADTs and apply them to solve real world problems
CO 2 : Implement linear and non-linear data structures and analyze their efficiencies
CO 3 : Implement tree traversal techniques using various algorithm paradigms
CO 4 : Model real world problems as search problems and implement them by applyingsophisticated search space reduction strategies.
CO 5 : Implement pattern search algorithms and apply heuristics to optimize the exploration mechanisms.

References:

1. Core Python Applications Programming, Wesley J Chun, Sunnyvale, California, 3rd Edition, 2012, Prentice Hall.
2. 2015 Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi, 1st Edition, Career Monk Publications.
3. The Algorithm Design Manual, Skiena SS, Springer, 2013.

Course Code:

MACHINE LEARNING LAB

(Professional Core)

Instruction	:	4 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand concept learning, supervised and unsupervised learning problems that exist in the real-world and implement them.
2. To understand and implement association analysis.
3. To understand and implement reinforcement learning methods.

LIST OF EXPERIMENTS

1. Implement FIND-S Algorithm in python on Play_Tennis dataset.

2. Implement Candidate Elimination Algorithm in python on Play_Tennis dataset.

3. Perform data pre-processing on loan defaulter dataset to clean the data.

4. Perform Exploratory Data Analysis (EDA) on loan defaulter dataset and record the pre-machine learning insights.

5. Implement Chi-Square Test on loan defaulter dataset in python and record the information about the accepted hypothesis.

6. Implement ID3 Decision Tree Algorithm in python on loan defaulter dataset. Also perform following:
   (a) Swap the feature columns in the dataset, learn the classifier and observe the changes.
   (b) Remove all the feature column entries except the entries from class label column, learn the classifier and observe the changes.
   (c) Modify certain class labels from random rows, learn the classifier and observe the changes.
   (d) Learn the overfitted and underfitted classifiers and record the observations.

7. Implement k-Nearest Neighbors classifier to recommend similar novels/books/movies by using Cosine Similarity as the distance measure.

8. Apply GridSearchCV on k-Nearest Neighbors classifier to identify the best parameters.

9. Explore the class imbalance by visualizing the credit card transactions dataset records as points and observe the populous group of points.

10. Find a way to handle class imbalance problem and implement the solution on credit card transactions dataset.

11. Implement Apriori Algorithm on Market-Basket transactions dataset to mine frequent association rules. Use min_support and min_confidence supplied during implementation.

12. Generate WordCloud using python on Shakespeare texts to find the frequent words.

13. Learn the correct number of clusters based on Elbow bent method by using k-means algorithm on Reuters dataset.

14. Implement DBSCAN clustering algorithm on gaussian generated points.

15. Implement Multi-Armed Bandit problem on Online-Ad Selection using Q-learning.

16. Implement Temporal Difference Learning to reschedule the University Examination by following certain assumptions and constraints.

Course Outcomes: At the end of the course, student will be able to:

CO 1 : Implement Concept Learning Algorithms
CO 2 : Implement Decision Tree Supervised Learning Algorithm on real-world dataset
CO 3 : Implement Apriori Algorithm on Market-Basket data
CO 4 : Learn correct number of clusters with k-means Unsupervised Learning algorithm
CO 5 : Apply Q-learning method and TDL methods of reinforcement learning on real-world problems.

References:

1. Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems, by Dipanjan Sarkar, Raghav Bali, Tushar Sharma, 2018.
2. Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python by Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Dmytro Dzhulgakov, Packt Publishers, February 2022.
3. Machine Learning using Python, Manaranjan Pradhan, U Dinesh Kumar, Wiley, 2019.

Course Code:

NATURAL LANGUAGE AND SPEECH PROCESSING

(Professional Core III)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To show how language related algorithms and techniques can be applied to important real-world problems (Spell Checking, Text Document Search, Part-of-Speech Tagging.)
2. To emphasis on scientific evaluation.
3. To learn the fundamentals, phonetics and speech synthesis for processing the speech.

Unit I – Overview

Origins and challenges of NLP-Language and Grammar-Processing Indian Languages NLP Applications, Information Retrieval. Language Modeling: Various Grammar- based Language Models, Statistical Language Model. Tools for NLP (NLTK).

Unit II - Word Level Analysis

Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and correction-Words and Word classes, Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar, Constituency, Parsing, Probabilistic Parsing.
Word level analysis methods in NLTK tool and compare different PoS taggers.

Unit III - Semantic Analysis and Speech Fundamentals

Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation, Discourse Processing: cohesion, Reference Resolution, Discourse Coherence and Structure.
Semantic analysis methods in NLTK tool.

Speech Fundamentals: Production and Classification of speech sounds, Acoustic Phonetics – Vowels, diphthongs, fricatives, Speech parameters – Speech production, Labelling.

Unit IV – Phonetics

Speech Sounds and Phonetic Transcription, Articulatory Phonetics – The Vocal Organs, Consonants: The Place of Articulation, Consonants: Manner of Articulation, Vowels.

Unit V – Speech Synthesis

Text Normalization – Sentence Tokenization, Non-Standard Words, Homograph Disambiguation, Phonetic Analysis – Dictionary Lookup, Names, Grapheme-to-Phoneme, Speech Corpora.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Understand regular expressions, information retrieval and extraction.
CO 2 : Understand parts- of- speech tagger based on HMMs and transformation-based learning
CO 3 : Comprehensive treatment of natural language generation and understand the speech fundamentals.
CO 4 : Understand how phones are useful for studying about speech sounds and how they are produced.
CO 5 : Understand Speech Synthesis as text analysis and waveform synthesis.

Textbooks:

1. Natural Language Processing and Information Retrieval, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2008.
2. Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Daniel Jurafsky and James H Martin, 2nd Edition, Pearson, 2013.

References:

1. Natural Language Processing with Python: Analyzing text with Natural Language Toolkit, Steven Biord, Ewan Klein and Edward Loper, O'Reilly, 2011.
2. Natural Language Understanding, James Allen, 2nd Edition, Benjamin /Cummings publishing company, 1995.
3. Introduction to Digital Speech Processing, Lawrence R. Rabiner and Ronald W. Schafer in Foundations and Trends in Signal Processing 1:1-2 (2007), Now Publishers.
4. Fundamentals of Speech Recognition, Lawrence Rabiner & Biing-Hwang Juang, Prentice-Hall International, Inc., 1993.

Course Code:

IMAGE PROCESSING AND COMPUTER VISION

(Professional Core IV)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Learn and understand the appropriate background image representation and data structures.
2. Learn and understand image preprocessing and segmentation.
3. To understand mathematical morphology and shape representation in image processing applications.
4. To understand real-world computer vision problems, and images using various techniques.
5. To study applications of 3-D vision and motion analysis.

Unit I – Image representation and data structures

Image Representations, image digitization, digital image properties, color images, cameras: an overview, Linear integral transforms, Images as a stochastic process, Levels of image data representation, Traditional image data structures, Hierarchical data structures.

Unit II – Image Preprocessing and segmentation

Image preprocessing: Pixel brightness transformations, Geometric transformations, Local pre-processing, Image restoration.

Segmentation: Thresholding, Edge-based segmentation, Region-based segmentation, matching, mean shift segmentation, active contour models, geometric deformable models, Fuzzy connectivity, toward 3D graph-based image segmentation. Graph cut segmentation, optimal single and multiple surface segmentation.

Unit III – Mathematical morphology and Shape representation

Mathematical morphology: Basic morphological concepts, Four morphological principles, Binary dilation, and erosion, Grayscale dilation and erosion, Skeletons and object marking, granulometry.

Shape Representations: Region identification, contour-based shape representation and description, region-based shape representation and description, shape classes.

Unit IV

Object recognition: Knowledge representation, statistical pattern recognition, Neural nets, syntactic pattern recognition, recognition as graph matching, optimization techniques in recognition, fuzzy systems, boosting in pattern recognition, random forests.

Unit V

Use of 3D vision and motion analysis:
Shape of X, full 3D objects, 2D view-based representations of a 3D scene, 3D reconstruction from an unorganized set of 2D views, and structure from motion.

Motion analysis: differential motion analysis methods, optical flow, analysis based on corresponding of interest points, detection of specific motion patterns, video tracking, and motion models to aid tracking.

Course Outcomes: At the end of the course, the student should be able to:

CO1 : Utilize adequate background knowledge about image representation and data structures.
CO2 : Apply image preprocessing and segmentation to real-world problems.
CO3 : Examine and analyze various mathematical morphology and shape representations in image processing applications.
CO4 : Design and develop solutions for real-world computer vision problems detecting images using various techniques.
CO5 : Choose and develop computer vision applications in 3 D vision and motion analysis.

Textbooks:

1. Image Processing, Analysis, & Machine Vision, Milan Sonka, Vaclav Hlavac and Roger Boyle, Cengage Publishers, 4th Edition, 2017.

References:

1. Digital Image Processing, Rafael C. Gonzalez and Richard E. Woods, Third edition. Pearson Education, 2007.
2. Programming Computer Vision with Python, Jan Erik Solem, Creative Commons,2012 OpenCV-Python Tutorials Documentation (Release 1), Alexander Mordvintsev & Abid K, 2017.

Course Code:

BIG DATA ANALYTICS AND PLATFORMS

(Professional Elective III)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the nature of Big Data and platform requirements
2. To learn the architecture of Hadoop Ecosystem.
3. To apply programming paradigm and tools for Big Data analytics.
4. To study the NoSQL database system.
5. To familiarize tools for stream and graph data analytics.

Unit I - Introduction to Bigdata and Hadoop

Classification of Digital Data, Structured Data, Semi-Structured data, Unstructured data, Characteristic of Data, Evolution of Big Data, Definition of Big Data, 3Vs of Data- Volume, Velocity and Variety, Big Data requirement, Traditional Business intelligent versus Big Data. Introduction to Big Data Analytics

Introduction to Hadoop: Hadoop System Architecture, Overview of HDFS, MapReduce Model, Hadoop Evolution, Various Tools in Hadoop Ecosystem and their functional aspects.

Unit II – HDFS and Anatomy of YARN

The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures.

Anatomy Of A Yarn: Resource Requests, Application Lifespan, Building YARN Applications, YARN Compared to MapReduce, Scheduling in YARN, Scheduler Options, Capacity Scheduler Configuration, Fair Scheduler Configuration, Delay Scheduling, Dominant Resource Fairness

Unit III – Map-Reduce and PIG

Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features

Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators

Unit IV - HIVE and HBASE

HIVE: Introduction, Data Types and File Formats, Databases in Hive, HiveQL: Data Definition, Data Manipulation, Queries, Views, Indexes, UDF, Schema Design

HBASE: Introduction, HBase v/s RDBMS, Components, Architecture, Data Model, HBase Shell, HBase Client API, Data Loading Techniques, CRUD operations.

Unit V – Core and Stream Data Analysis Tools and Graph Data Analysis Tools

Introduction to Core Spark concepts, Standalone Applications, Programming with RDDs - Creating RDDs, RDD Operations, Common Transformations and Actions.

Spark Streaming – Architecture and Abstraction, Transformations in spark streaming, output operations, input sources, Streaming UI, performance considerations.

Graph Data Analysis Tools: Apache Giraph Architecture, Spark-GraphX, Examples of Graph Data Analytics using Giraph and Spark-GraphX.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Understand Big Data characteristics and Hadoop Ecosystem.
CO 2 : Describe underlying architecture of Hadoop.
CO 3 : do experiments on Map Reduce, PIG.
CO 4 : Use HIVE data warehouse and NoSQL databases such as HBase.
CO 5 : Analyse core, stream and graph data analytics using appropriate tools.

Textbooks:

1. Big Data and Analytics, Seema Acharya, Subhashini Chellappan, Wiley, 2nd Edition, 2019.
2. HADOOP: The definitive Guide, Tom White, O Reilly, 2012.

References:

1. Professional NoSQL, Shashank Tiwari, WROX, 2011.
2. Big Data Black Book, DreamTech, 2016.
3. Learning Spark: Lightning-fast data analysis by Holden Karau, Andy Konwinski, Patrick Wendell & Matei Zaharia, O'reilly, 1st Edition, 2015.
4. Practical Graph Analytics with Apache Giraph by R. Shaposhnik, C. Martella, D. Logothetis, Apress, 1st Edition, 2015.

Course Code:

VIRTUAL REALITY AND AUGMENTED REALITY

(Professional Elective III)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To study about Fundamental Concept and Components of Virtual Reality
2. Understanding how to do Modeling, model management.
3. Develop an understanding of the basic concepts of augmented reality.
4. To learn human sense devices that support augmented reality.

Unit I - Introduction to Virtual Reality

Introduction, Perception of Reality, Goal of virtual reality, The three I's of virtual reality, commercial VR technology and the five classic components of a VR System. Applications: Medical Applications of VR, VR in Education, VR in Manufacturing, VR in Robotics.

Unit II - Input Output Devices

Trackers, Navigation, and Gesture Interfaces: Three-dimensional position trackers, navigation and manipulation interfaces and gesture interfaces. Graphics displays, sound displays and haptic feedback.

Unit III - Modeling

Geometric modeling: Virtual Object Space, Object Visual appearance, kinematics modeling: Homogeneous Transformation Matrices, Object Position, Transformation Invariants, Object Hierarchies, Viewing the Three-Dimensional World, physical modeling: Collision Detection, Surface Deformation, Force Computation, Force Smoothing and Mapping, Haptic Texturing, behavior modeling, model management: Level-of-Detail Management, Cell Segmentation.

Unit IV - Introduction to Augmented Reality

Computer Generated Worlds, Augmented reality, Goal of AR, Understanding Virtual Space, Defining Visual Space and Content, Defining Position and Orientation in Three dimensions, Navigation, Relationship between augmented reality and other technologies, Augmented Reality concepts, two-step process of augmented reality applications, Concepts related to augmented reality, ingredients of an augmented reality Experience, AR Applications: Soil sampling, Health care, Navigation assistance, Information Control and Big Data Visualization.

Unit V - Human Senses and Devices

The Mechanics of Feeling: The Science of Feeling, Anatomy and Composition of the Skin, Tactile and Force Feedback Devices: Haptic Illusions, Tactile Feedback Devices, Force Feedback Devices, Devices to Enable Navigation and Interaction: 2D Versus 3D Interaction and Navigation, The Importance of a Manual Interface, Gaming and Entertainment Interfaces, Navigating with Your Mind.

Course Outcomes: At the end of the course, the student should be able to:

CO 1	:	Understand the fundamentals of Virtual Reality.
CO 2	:	Understand the components of VR.
CO 3	:	Exhibit proficiency with Modeling, types, Model management.
CO 4	:	Understand and analyze concepts of augmented reality.
CO 5	:	Learn human sense devices that support augmented reality.

Textbooks:

1. Virtual Reality Technology, Gregory C. Burdea and Philippe Coiffet, John Wiley and Sons, Inc (Wiley Inter Science), Second Edition, 2006.
2. Understanding Augmented Reality, Concepts and Applications, Alan B. Craig, Morgan Kaufmann, 2013.

References:

1. Developing Virtual Reality Applications, Foundations of Effective Design, Alan Craig, William Sherman and Jeffrey Will, Morgan Kaufmann, 2009.
2. Virtual Reality Systems, John Vince, Pearson Education Asia, 2007.
3. Augmented reality for developers: Build practical augmented reality applications with Unity, ARCore, ARKit, and Vuforia, Linowes, J., & Babilinski, K. (2017).
4. Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability), Steve Aukstakalnis, 1st Edition.

Course Code:

SCALABLE ARCHITECTURES FOR MACHINE LEARNING APPLICATIONS

(Professional Elective – III)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Students will be able to learn application and building Scalable Machine Learning, Hadoop, SMACK Stack and also Message Services.
2. Students will be able to select the appropriate architecture for enterprise architectures based on the size, scale and applications used in the enterprise.

Unit I – Introduction

Introduction to Scalable Machine Learning, Some Machine Learning Background. Algorithms for Large scale Learning, Overview of Hadoop and Current Big Data Systems.

Unit II – Apache Spark for ML

How Programming for Data Flow Differs, Apache Spark Engine, Working with Vectors and Matrices in Spark, Brief tour of Spark ML, beyond parallelization, Practical Big Data.

Unit III – SMACK Stack

Anatomy of Fast Data Applications, SMACK Stack – Functional Decomposition, Message Backbone- Understanding messaging requirements, Data ingestion, Fast data& low latency, Message Delivery Semantics, Distributing Messages, The Model: Akka, Storage: Apache Cassandra.

Unit IV – Compute Engines

Micro Batch Processing, One-at-a time Processing, Choice of processing engine, Storage as the Fast Data Borders, The message backbone as Transition Point, The Manager: Apache Mesos, The Broker: Apache Kafka.

Unit V – Serving and Substrate

Sharing stateful streaming state, Data Driven Micro-services, State and Micro-services.

Deployment environments for Fast Data Applications, Application containerization, resource scheduling, Kubernetes, Cloud Deployments, Improving SMACK: Fast Data Patterns, Data Pipelines.

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

CO 1: Understand the basic concepts of Scalable Machine Learning and working of Big data Systems
CO 2 : Use Apache Spark for Machine Learning
CO 3 : Use Apache Akka and Cassandra in SMACK stack
CO 4 : Use Apache Mesos and Apache Kafka for cluster management and event streaming
CO 5 : Improve SMACK through Fast Data Patterns and Data Pipelines

Textbooks:

1. Big Data SMACK: A Guide to Apache Spark, Mesos, Akka, Cassandra, and Kafka, Raul Estrada and Isaac Ruiz, Apress, 2016
2. Large Scale Machine Learning with Python, Bastiaan Sjardin, Luca Massaron, Alberto Boschetti, Packt Publishing, 2016.

References:

1. Designing Fast Data Application Architectures by Gerard Maas, Stavros Kontopoulos, Sean Glover, Publisher: O'Reilly Media, Inc., June 2018
2. Spark- The definitive Guide by Bill Chambers & Matei Zaharia, O'Reilly Media, Inc., June 2019
3. Link - https://learning.oreilly.com/videos/scalable-machine-learning

Course Code:

AI AND GAMES

(Professional Elective IV)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To acquire knowledge on basic concepts, goals, importance, and challenges of artificial intelligence and games. To analyze the nature of various advanced search strategies in AI.
2. To learn and develop theoretical foundations on AI methods and able to analyze & identify applications, strengths and limitations of different methods To resolve Complex Decision-Making rules and their applications in AI.
3. To gain knowledge and demonstrate designing of game and AI algorithm considering different ways of using AI in games & AI methods, and type of game.
4. To understand the importance of content generation and acquiring the concepts to generate content using different methods. To acquire ability to analyze, interpret, and evaluate content & content generators.
5. To learn modeling of players using different approaches and play game by specifying inputs for the model.

Unit I -Introduction

A Brief History of Artificial Intelligence and Games, Academia, Industry, The Gap, Games for Artificial Intelligence, Games Are Hard and Interesting Problems, Human-Computer Interaction, Games Are Popular, There Are Challenges for All AI Areas, Games Best Realize Long-Term Goals of AI, Why Artificial Intelligence for Games, AI Plays and Improves Your Game, More Content, Better Content, Player Experience and Behavioral Data Analytics.

Unit II - AI Methods

General Notes, Representation, Utility, Learning = Maximize Utility (Representation), Ad-Hoc Behavior Authoring, Finite State Machines, Behavior Trees, Utility-Based AI, Tree Search, Uninformed Search, Best-First Search, Mini-max, Monte Carlo Tree Search, Evolutionary Computation, Local Search. Evolutionary Algorithms, Supervised Learning, Artificial Neural Networks, Support Vector Machines, Decision Tree Learning, Reinforcement Learning, Core Concepts and a High-Level Taxonomy, Q-Learning, Unsupervised Learning, Clustering, Frequent Pattern Mining, Notable Hybrid Algorithms, Neuro-evolution, TD Learning with ANN Function Approximators.

Unit III - Playing Games

Why Use AI to Play Games? Playing to Win in the Player, Role, Playing to Win in a Non-player Role, Playing for Experience in the Player Role, Playing for Experience in a Non-player Role, Summary of AI Game-Playing Goals and Roles, Game Design and AI Design Considerations, Characteristics of Games, Characteristics of AI Algorithm Design, How Can AI Play Games? Planning-Based Approaches, Reinforcement Learning, Supervised Learning, Chimeric Game Players, Which Games Can AI Play? Board Games, Card Games, Classic Arcade Games, Strategy Games, Racing Games, Shooters and Other First-Person Games, Serious Games, Interactive Fiction.

Unit IV - Generating Content

Why Generate Content, Taxonomy, Taxonomy for Content, Taxonomy for Methods, Taxonomy of Roles, How Could We Generate Content? Search-Based Methods, SolverBased Method Grammar-Based Methods, Cellular Automata, Noise and Fractals, Machine Learning, Roles of PCG in Games, Autonomous, Experience-Driven, Experience-Agnostic, What Could Be Generated? Levels and Maps, Visuals, Audio, Narrative, Rules and Mechanics, Games. Evaluating Content Generators, Why Is It Difficult? Function vs Aesthetics, How Can We Evaluate a Generator.

Unit V - Modeling Players

What Player Modeling Is and What It Is Not, Why Model Players? A High-Level Taxonomy of Approaches, Model-Based (Top-Down) Approaches, Model-Free (Bottom-Up) Approaches, Hybrids, What Is the Model"s Input Like? Gameplay, Objective, Game Context, Player Profile, Linked Data, What Is the Model"s Output Like? Modeling Behavior, Modeling Experience, No Output, How Can We Model Players? Supervised Learning, Reinforcement Learning, Unsupervised Learning, What Can We Model? Player Behavior, Player Experience.

Course Outcomes: After completion of course, students will be able to

CO 1 : Acquire basic knowledge on history, fundamentals, goals, and challenges of Artificial Intelligence and games.
CO 2 : Learn core concepts and differentiate AI methods and able to analyze & identify applications, strengths and limitations of different methods.
CO 3 : Design of game and AI algorithm considering different ways of using AI in games and AI methods, and type of game.
CO 4 : Generate content using different methods and based on content taxonomy. Analyze, Interpret, and evaluate content & content generators.
CO 5 : Model players using top down, bottom up, and hybrid approaches.

Textbooks:

1. Artificial Intelligence and Games by Georgios N. Yannakakis and Julian Togelius, Springer, 1st Edition, 2018.
2. Artificial Intelligence: Structures and Strategies for complex problem solving, G. Luger, Fourth Edition, Pearson Education, 2002.

References:

1. Artificial Intelligence for Games, Ian Millington and John Funge, Elsevier, Second Edition. 2009.
2. Programming Game AI by Example - Mat Buckland, First Edition, Word Ware Game Developers Library, Jones and Bartlett Publishers, Inc, 2004.

Course Code:

COMPUTATIONAL INTELLIGENCE

(Professional Elective IV)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	End Examination Marks	:	60 Marks
Credits	:	3	End Exam Duration	:	3 Hours

Course Objectives:

1. To learn about the soft computing and study rough sets.
2. To learn the concepts of Associative Memories in ANN.
3. To understand Fuzzy Logic system for intelligent system design.
4. To study and explore basics of Genetic Algorithms
5. To analyze the effect of hybrid-system design.

Unit I – Introduction to Soft Computing and Rough Sets

Introduction, Hard vs Soft Computing, Real-Life situations characterized by soft qualities, Soft Computing Techniques. Rough sets – Its Introduction, Indiscernibility, Set Approximations, Properties of Rough Sets, Rough Membership, Reducts, Applications.

Unit II – Associative Memories in ANN

Associative memory, Hetero-correlators, HAM, BAM, multiple training encoding strategy, Exponential BAM, Real coded pattern pairs, Applications.

Unit III - Fuzzy Logic and Genetic Algorithms

Fuzzy set theory, crisp sets, fuzzy sets, crisp relations, Fuzzy relations, Fuzzy systems, Crisp logic, predicate logic, fuzzy logic, fuzzy based systems, Defuzzification methods, applications.

Unit IV – Genetic Algorithm

Fundamentals of GA – creation of off springs – encoding – fitness function reproduction – crossover- insertion and deletion- mutation- bitwise operators – applications.

Unit V - Hybrid Systems

Hybrid systems – Neuro Fuzzy – Neuro Genetic – fuzzy Genetic hybrids- GA based weight determination and applications- fuzzy BPN.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Appreciate the soft computing paradigm and understand the rough sets.

CO 2 : Understand the concepts of Associative Memories in ANN.

CO 3 : Understand Fuzzy Logic system for intelligent system design.

CO 4 : Study and explore basics of Genetic Algorithms

CO 5 : Analyze the effect of hybrid-system design.

Textbooks:

1. Introduction to Soft Computing-Neuro-Fuzzy and Genetic Algorithms, Samir Roy and Udit Chakraborty, Pearson, First Edition, 2013.
2. Neural Networks, Fuzzy Logic and Genetic Algorithms, Rajasekaran.S, Vijayalakshmi Pai.G.A, PHI, 2011.

References:

1. Fuzzy sets and Fuzzy Logic Theory and Applications, George J. Klir, Bo Yuan, Prentice – Hall of India Pvt. Ltd, 2000.
2. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning And Machine Intelligence by Jang and Sun and Mizutani, Pearson India, 1st Edition, 2015.

Course Code:

ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING
(Professional Elective IV)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To know the characteristics, practical issues and common architectures of Artificial Neural Networks (ANNs)
2. To learn the training of supervised ANNs
3. To learn the training of unsupervised ANNs and problems with gradients and how to handle them using deep learning
4. Necessary knowledge to design CNN, AE and GAN networks
5. Necessary knowledge to design RNNs and LSTM networks

Unit I – Introduction to Neural Networks and Learning Processes

What is a Neural Network, Human Brain, Models of a Neuron, Network Architectures, Feedback, Knowledge Representation, Artificial Intelligence and Neural Networks.

Basic Learning Rules: Error-Correction Learning, Memory-based Learning, Hebbian Learning, Competitive Learning, Boltzmann Learning. Learning Paradigms- Learning with a teacher, learning without a teacher, Learning tasks, Memory, Adaptation.

Unit II – Supervised ANNs: Single Layer and Multilayer Perceptrons

Introduction, Adaptive Filtering problem, Unconstrainted optimization techniques, Least-Mean-Squares Algorithm, Learning Curves, Perceptron, Perceptron Convergence Theorem. Back-Propagation Algorithm, XOR Problem, Heuristics for making the Back-Propagation perform better, Output Representation and Decision Rule, Feature Detection, Back-Propagation and Differentiation, Generalization, Approximation of functions, Virtues and Limitations of Back-Propagation Learning, Accelerated Convergence of Back-Propagation Learning.

Unit III – Unsupervised ANNs and Introduction to Deep Learning

Introduction, Kohonen Self-Organizing Feature Maps – Theory, Architecture, Flowchart, Training Algorithm. Learning Vector Quantization - Theory, Architecture, Flowchart, Training Algorithm.

Revisiting Machine Learning pipeline, Challenges with ML algorithms, Motivation towards Deep Learning, Learning Hierarchical Representations, Essential differences between traditional machine learning and deep learning, why deep learning was not took off in 1986, List of Deep Learning Networks.

Unit IV – Convolutional Neural Networks, Deep Autoencoders and GANs

Introduction, The Basic Structure of a Convolutional network, Training a convolutional network, Introduction to Autoencoders in deep learning, structure of a deep autoencoder and its training, Variational Autoencoder. Introduction to Generative Adversarial Network, structure of GAN and its training.

Unit V – Recurrent Neural Networks and LSTM

Introduction, analyzing variable length inputs, Tackling seq2seq with Neural N-Grams, Word2Vec, A Case for Stateful Deep Learning models, RNNs: The Architecture of Recurrent Neural Networks, The Challenges of Training Recurrent Networks, Long Short-Term Memory (LSTM).

Course Outcomes: At the end of the course, the student should be able to:

CO 1 : Appreciate ANN and various learning rules in ANN
CO 2 : Train binary and multiclass ANN models
CO 3 : Train unsupervised ANNs and understand the transition from ANN to deep neural networks (DNNs).
CO 4 : Train CNN, AE and GAN models
CO 5 : Train RNN, understand the challenges in training RNN and train LSTM

Textbooks:

1. Neural Networks: A Comprehensive Foundation, Simon Haykin, Third Edition, Pearson, 2011.
2. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, Nicholas LoCascio and Nikhil Buduma, O'Reilly Media Inc, June 2017.

References:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville – (2017, MIT).
2. Neural Networks and Deep Learning A Textbook, Charu C. Aggarwal, Springer, 2018.
3. Neural Networks for Pattern Recognition, Christopher Bishop, Clarendon Press, Oxford 1995.

Course Code:

NATURAL LANGUAGE AND SPEECH PROCESSING LAB
(Professional Core)

Instruction	:	4 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	End Exam Duration	:	3 Hours

Course Objectives:

1. Knowledge on basic Language processing features, design an innovative application using NLP components.
2. To learn and implement the speech processing concepts.
3. To enable the students to solve real-world problems using speech processing concepts.

LIST OF EXPERIMENTS:

NLP PROBLEMS

1. Write a python program to perform the following tasks on Reuters dataset from NLTK.
   (a) Split the text document content by Whitespace.
   (b) Use Regular Expressions for cleaning the text.
   (c) Convert the text from uppercase to lowercase.
   (d) Count the number of distinct words and characters.
2. Write a python program to perform stop words removal on Reuters dataset from NLTK using standard stop words list and standard procedure.
3. Write a python program to explore PoS tagging module of NLTK on Reuters dataset.
4. Compare Stanford PoS tagging and the NLTK PoS tagging and draw the observations.
5. Write a python program to explore WordNet and SentiWordNet modules from NLTK.
6. Write a python program to generate unigrams, bigrams and trigrams on Reuters dataset from NLTK.
7. Write a python program to obtain Bag of Words from three simple text documents using scikit learn CountVectorizer.
8. Write a python program to find the most similar words taken from Brown corpus and Movie Reviews dataset using Word2Vec genism model and plot the most similar words using t-SNE.
9. Write a python program to perform TF-IDF on four simple text documents.
10. Write a python program to classify the text using HMM
11. Write a python program to classify the text with RNN.
12. Write a python program to predict the next word in a sentence using RNN.
13. Write a python program to analyze tweets and classifying them into disaster and non-disaster ones with GloVe and LSTM to extract useful information during crises.
14. Write a python program to perform sentiment analysis on Amazon product reviews using WordNet, SentiWordNet of NLTK and LSTM.

SPEECH PROCESSING PROBLEMS

15. Write a python program to generate speech spectrum based on sampling frequency, depth and Language.
16. Write a python program to extract words, syllables and phones from the audio file.
17. Write a program to visualize the Autocorrelation plot (amplitude (vs) time graph) by analyzing the nature of uttered vowels (a, e, i).

Course Outcomes: After completion of course, students will be able to:

CO1: Understand and implement proper experimental methodology for training and evaluating empirical NLP systems.
CO2: Design, implement, and analyze NLP algorithms and classify the text.

CO3: Implement deep learning's RNN for text classification and next text sequence prediction and implement LSTM for text classification.
CO4: Implement basic speech processing modules to extract the textual counterparts.
CO5: Implement the speech signal spectrum and plot the nature of speech signal

References:

1. Steven Bird, Ewan Klein and Edward Loper, —Natural Language Processing with Python, First Edition, O 'Reilly Media, 2009.
2. Daniel Jurafsky, James H. Martin—Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2013.
3. Breck Baldwin, —Language Processing with Java and Ling Pipe Cookbook, Atlantic Publisher, 2015.

Course Code:

IMAGE PROCESSING AND COMPUTER VISION LAB

(Professional Core)

Instruction	:	4 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Learn various python packages for real-world image data processing.
2. Understand digital image processing pipeline.
3. Learn and understand the programs that uses image enhancement, restoration and segmentations methods.
4. Learn and understand the programs that uses to implement motion detection, face detection in computer vision applications.
5. Learn and understand the programs that makes use of object tracking in virtual environments.

LIST OF EXPERIMENTS:

1. Write a python program to find the histogram of color images.
2. Write a python program to find the transformation of the following
   a) Discrete cosine transform b) Wavelet transform
3. Write a python program to detect edge using the following
   a) Canny edge method b) Parametric edge detection
4. Write a python program image restoration using Weiner filtering
5. Write a python program to find the threshold value of an image by Otsu's method.
6. Write a python program for curve detection using the Hough transform.
7. Write a python program to segment an image using the watershed segmentation method.
8. Write a python program for multiple optical surface segmentation for a given image.
9. Implementation of Quadtree region identification for an image.
10. Implementation of a python program region graph construction from Skeleton of an object.
11. Get the awareness on Keras and Tensorflow python modules.
12. Implementation of K -means cluster method using TensorFlow.
13. Implementation of SVM learning and classification using TensorFlow.
14. Write a python program and find the motion detection and tracking of an object using Lucas kanade tracking method
15. Implement the following
    a) Line labeling b) Iterative method finding the closest reciprocal point.
16. Implement any of the following using TensorFlow /OpenCV.
    a) Automatic Number Plate recognition
    b) Fingerprint Recognition
    c) Face Recognition
17. Write a python program to instantiate a small convnet and visualize the convnet.
18. Write a python program to extract features with a pre-trained CNN.
19. Write a python program to build CNN to distinguish between Horses and Humans
20. Write a python program to utilize (i) dropout and (ii) weight regularization to prevent overfitting in neural networks.
21. Classifying Hand written digits using tensorflow.js
22. Hands-on object detection using YOLOv4 in python.

Course Outcomes: At the end of the course, the student should be able to:

CO1	:	Pre-process the real-world images.
CO2	:	Apply image filtering methods.
CO3	:	Develop the programs that uses image enhancement, restoration and segmentations methods
CO4	:	Evaluate and implement motion detection, face detection in computer vision applications
CO5	:	Develop the programs to implement object tracking in virtual environments

References:

1. Pattern classification. Richard Duda. Hart and David strok John Wiley publishers.
2. Image Processing. Analysis and Machine Vision, Milan Sonka, Vaciav Hlavac and Roger Boie,Second Edition, Thomson learning, 2001.
3. OpenCV-Python Tutorials Documentation (Release 1), Alexander Mordvintsev & Abid K, 2017.
4. Deep Learning for Computer Vision with Python - Practitioner Bundle, Dr. Adrian Rosebrock, published by PYIMAGESEARCH, 1st Edition, 2017.