ACADEMIC YEAR 2026-2027

SEMESTER WISE CREDIT FOR B.E CSE(AI&ML)

S.No	SEMESTER	CREDIT
1	I	21
2	II	19.5
3	III	25
4	IV	24
5	V	24
6	VI	19
7	VII	18
8	VIII	14
TOTAL CREDITS		**164.5**

SCHEME OF INSTRUCTION & EXAMINATION
Common for B.E CSE(AI) and B.E CSE (AI&ML)
SEMESTER-V

S.No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	PC506CS	Design and Analysis of Algorithms	3	-	-	3	30	70	3	3
2	PC403CS	Computer Organization	3	-	-	3	30	70	3	3
3	PC501AI	Machine Learning	3	-	-	3	30	70	3	3
4	PC502AI	Automata Theory & Compiler Design	3	-	-	3	30	70	3	3
5	PC501DS	Big Data Analytics	3	-	-	3	30	70	3	3
6	PC502CS	Principal of Programming Language	3			3	30	70	3	3
7	PE	Professional Elective-I	3	-	-	3	30	70	3	3
Practical / Laboratory Courses										
8	PC551AI	Machine Learning lab	-	-	2	2	25	50	3	1
9	PC552DS	Big Data Analytics lab	-	-	2	2	25	50	3	1
10	PC552CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
Total			**21**		**6**	**27**	**285**	**640**	30	**24**

Profession Elective – I	
Course Code	**Course Title**
PE622CS	Design Thinking
PE511AI	Data Visualization
PC501CS/PE	Software Engineering
ES---EC	Signals and Systems Essentials
PE512AI	Mobile Application Development

PC506CS	**DESIGN AND ANALYSIS OF ALGORTIHMS**					
Prerequisites	Data Structures		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To review elementary data structures, order notation and algorithm analysis
2. To learn algorithm design strategies such as Divide-and-Conquer
3. To learn algorithm design strategies such as greedy method and dynamic programming
4. To learn algorithm design strategies such as back tracking and branch & bound technique
5. To understand the concepts of NP-hard and NP-complete

Course Outcomes:

CO1: Analyze the time and space complexity of algorithms using asymptotic notations and recurrence relations

CO2: Design algorithms for various computing problems using Divide-and-Conquer technique

CO3: Critically analyze the different algorithm design techniques for optimization problems

CO4: Modify existing algorithms to improve efficiency using state-space search techniques

CO5: Design algorithms for various computing problems and classify them based on computational complexity classes

UNIT-I

Introduction & Elementary Data Structures: Introduction, Fundamentals of algorithm(Line Count, Operation Count), Analysis of algorithms(Best, Average, Worst case), Asymptotic Notations(O,Ω,Θ) Recursive Algorithms, Analysis using Recurrence Relations, Master's Theorem.

Review of elementary data structures–Graphs: BFS, DFS, Bi-Connected Components. Sets: representation, UNION, FIND operations.

UNIT-II

Divide-and-Conquer Method: The general method, Binary search, Finding maximum and minimum, Merge sort, Quick sort.

Brute Force: Knapsack, Traveling salesman problem, Convex-Hull

UNIT-III

Greedy Method: Knapsack problem, Minimum spanning trees, Single source shortest path, Job sequencing with deadlines, Optimal storage on tapes, Optimal merge pattern

Dynamic programming method: All pairs shortest paths, Optimal binary search tress, 0/1 Knapsack problem, Reliability design, Traveling salesman problem

UNIT-IV

Back tracking: N-queens problem, Graph coloring, Hamiltonian cycles

Branch-and-bound: FIFO & LC branch and Bound methods, 0/1 Knapsack problem, Traveling salesperson

UNIT-V

NP-hard and NP-complete problems: Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, **Proofs for NP Complete Problems:** Clique, Vertex Cover.

Parallel Algorithms: Introduction, models for parallel computing

Suggested Readings:

1. Horowitz E, Sahni S, Fundamentals of Computer Algorithms, 2ndEdition, Universities Press, 2007
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012
3. Michael T. Goodrich, Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, John Wiley & Sons, 2002

PC403CS	**COMPUTER ORGANIZATION**					
Prerequisites	-		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives

1. To introduce the fundamental concepts of computer architecture, including functional units, bus structures, and input/output organization.
2. To explain the organization and functioning of memory systems, including RAM, ROM, cache, and virtual memory, and their impact on system performance.
3. To provide a comprehensive understanding of the 8086 microprocessor architecture, instruction set, and assembly-level programming.
4. To develop the ability to interface various I/O devices with the 8086 microprocessor and understand the use of interrupts and peripheral controllers.
5. To expose students to real-time applications of microprocessor-based systems in embedded and IoT development using programmable peripheral interfaces

Course Outcomes:

CO1: Describe the functional units of a computer, types of computers, and various I/O mechanisms including DMA and interrupt-driven I/O.

CO2: Explain the principles of memory organization, including RAM, ROM, cache, and virtual memory concepts.

CO3: Analyze the architecture and internal operations of the 8086 microprocessor and interpret instruction sets and addressing modes.

CO4: Apply assembly-level programming techniques for basic problem-solving using 8086 instructions and subroutine handling.

CO5: Demonstrate how to interface output displays and input devices (e.g., keyboards) with microprocessors using peripheral support chips.

UNIT-I

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Performance, Multiprocessors and Multicomputers, Historical perspective. Input/output Organization: Accessing I/O devices, Interrupts, Processor examples, Direct memory access, parallel interface and serial interface.

UNIT-II

The Memory System: Basic concepts, Semiconductor RAM memories, Read-Only memories, Speed, Size and Cost, Cache memories, Performance considerations, Virtual Memories, Memory management requirements, Secondary Storage.

UNIT-III

8086 Architecture: Introduction to microprocessors and microcontrollers, 8086 Processor Architecture, Internal operations, Instructions and timings. Programming the 8086 Introduction to 8086 instructions, Addressing modes and Programming techniques with Additional instructions.

UNIT-IV

Stacks and subroutines, interfacing peripherals Basic interfacing concepts, interfacing output displays, Interfacing input keyboards. Interrupts 8086 Interrupts, Programmable peripheral interface (Intel 8255A),

UNIT-V

8251, 8253/8254, 8257 8259, real world case study of 8086 with peripheral devices, Application of further to embedded systems development and IOT development

Suggested Readings:

1. Carl Hamacher, Zvonko Vranesic, SafwatZaky, Computer Organization, 5th Edition, McGraw Hill, 2002.

2. Fundamentals Of Microprocessors and Microcontrollers 8/E PB Paperback – 1 January 2012 by Ram B Dhanpat Rai Publishers

3. Microprocessors and Interfacing Paperback – 23 May 2012, by M. Saravanan N. Senthil Kumar , S. Jeevananthan r), S.K.Shah OXFORD PUBLICATIONS

4.. Embedded / Real-Time Systems: Concepts, Design and Programming Black Book, New ed Paperback – 1 January 2003 by Dr. K.V.K Prasad , Dreamtech Publishers

5. VLSI Design Black Book by Dr. K.V.K.K. Prasad , Dr. Kattula Shyamala Dreamtech Publishers

6. 21 IoT Experiments: Learn IoT, the Programmer's way Paperback – 1 January 2017 by Yashavant Kanetkar , Shrirang Korde BPB PUBLICATIONS

PC501AI	**MACHINE LEARNING**					
Prerequisites	Artificial Intelligence		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To learn the basics of machine learning and its main types
2. To understand how to use probability and regression for making predictions.
3. To study the support vector machines and analyze learning with statistics.
4. To explore how to evaluate and improve models using neural networks and optimization
5. To learn how to build models using clustering, active learning, and reinforcement learning

Course Outcomes

On completion of this course, the student will be able to

CO1: Define the basic concepts related to Machine Learning.
CO2: Recognize the underlying mathematical relationships across ML algorithms and their paradigms.
CO3: Determine the various applications of Machine Learning
CO4: Model, design and develop solutions to real world problems using Machine Learning Algorithms
CO5: Design, implement, and evaluate deep neural networks and reinforcement learning agents to solve complex

UNIT-I
Introduction to Machine Learning: Introduction to Machine Learning, Types of Machine Learning,
Concept Learning: Concept Learning Task and General to Specific Ordering of Hypothesis.
Performance Evaluation Metrics: Confusion Matrix, ROC Curve, AUC Curve

UNIT-II
Data Pre-processing Techniques: Loading the data, Missing values, Outlier, Managing categorical data, k- Fold Cross-validation and Data splitting with training and test sets.
Feature Engineering: Data scaling and normalization, whitening, Feature selection and filtering, Visualization of high-dimensional Datasets , PCA kernel PCA.

UNIT-III

Regression Algorithms: Linear Regression, Logistic regression, Lasso and Ridge regression, Error measures with Regression.
Classification Algorithms: Decision Trees, k-Nearest Neighbors (k-NN), Bayes theorem, Naïve Bayes classifiers, Support Vector Machines

UNIT-IV

Ensemble Learning: Introduction to Ensemble Learning with Random Forests, AdaBoost, XGBoost, Light GBM and Voting classifier.
Clustering Algorithms: Basics of Clustering, K-means, Hierarchical Clustering, DBSCAN, Evaluation methods of Clustering

UNIT-V

Neural Networks: Introduction to Neural Networks, Activation functions, multi-layer perceptron, Feed Forward Networks, Backpropagation Algorithm, Introduction to CNN
Reinforcement Learning: overview, State and Action Spaces, The Reward Function, Discounting, Action Selection, Policy, Markov decision processes, Q-learning, uses of Reinforcement learning
Applications of Machine Learning: Text Classification, Image Recognition & Speech Recognition

Suggested Reading:

1. Giuseppe Bonaccorso, "Machine Learning Algorithms", 2nd Edition, Packt, 2018

2. Tom Mitchel "Machine Learning", Tata McGraw Hill, 2017

3. Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing Company, 2020.

4. M N Murty and Ananthanarayana V S, "Machine Learning: Theory and Practice", Universities Press; First Edition, 2024

5. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

6. Uma N. Dulhare , Khaleel Ahmad , Khairol Amali Bin Ahmad , Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications, Scrivener Publishing,Wiley,2020

PC502AI	**AUTOMATA THEORY & COMPILER DESIGN**					
Prerequisites	Discrete Mathematics		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. Provide information to some of the central ideas of theoretical computer science from perspective of formal languages
2. Introduce fundamental concepts of Formal languages, grammars and types of automata
3. Introduce the major concepts of language translation and compiler design.
4. Impart the knowledge of practical skills for constructing a compiler.
5. Introduce various phases of compiler in detail with examples

Course Outcomes:

On completion of this course, the student will be able to:

CO1: Identify and understand the concept of finite automata and their power to recognize regular languages

CO2: Design context-free grammars and pushdown automata for formal languages

CO3: Construct LL and LR parsers and apply syntax-directed translation for semantic analysis

CO4: Generate intermediate code representations and apply three-address code for translation

CO5: Apply code optimization techniques and explain runtime storage organization for code generation

UNIT-I

Introduction to finite automata: The central concepts of automata theory, Structural representation of FA, Types of FA, Conversion of NFA to DFA, NFA with epsilon to NFA without epsilon conversion.
Regular Expression: Introduction to Regular language, Algebraic laws for regular expressions, Conversion of FA to RE, Conversion of RE to FA, Pumping lemma for Regular language

UNIT-II

Grammar: Definition of Grammar, Types of grammars, Derivation, Types of derivations, Derivation tree, Ambiguity, Left recursion elimination.
Push Down Automata: Definition of PDA, Structural representation of PDA, Construction of PDA

UNIT III

Turing Machine: Definition of TM, Structural representation of TM, Construction of TM.
Compiler: Definition of Compiler, Phases of Compiler, Lexical Analysis, Input Buffering.
Syntax Analysis: Types of Parsing, Recursive Descent parsing, Predictive Parsing, Bottom-up Parsing SLR, CLR, and LALR

UNIT-IV

Semantic Analysis: Introduction to Syntax Directed Definition, Syntax Directed Translation, Attributes, Types of Attributes, Bottom-up evaluation of attributes.
Intermediate code generation: Types of Intermediate codes, Types of Three address codes

UNIT-V

Runtime Environment: Storages organization, Storage allocation strategies: Static, Stack, Heap allocations, Activation Record.
Code Optimization: Introduction, Principal sources of optimization, basic block, partition algorithm of basic block, flow graph, techniques of loop optimizations.
Code generation: Issues in code generation, DAG, Simple code generator

Suggested Reading:

1. Compilers: Principles, Techniques and Tools - Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
2. Compiler Construction - Principles and Practice, Kenneth C Louden, Cengage Learning.
3. Modern compiler implementation in C - Andrew W Appel, Revised Edition, Cambridge University Press
4. The Theory and Practice of Compiler writing, J.P. Tremblay and P.G. Sorenson, TMH

PC501DS	**BIG DATA ANALYTICS**					
Prerequisites	Probability & Statistics, Introduction to Data Science		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To understand the fundamentals of Big Data and Hadoop.
2. To understand Apache Hadoop, HDFS concepts and HDFS interfaces for managing Big Data.
3. To understand Map Reduce Jobs, YARN and Spark for Big Data Processing.
4. To explore Hadoop Ecosystem tools such as Pig, Hive and HBase.
5. To learn basic Machine Learning concepts and NoSQL databases for Big Data applications.

Course Outcomes:

On completion of this course, the student will be able to

CO1: Explain the concepts of Big Data and Hadoop

CO2: Use HDFS and Hadoop tools for storing and managing large datasets.

CO3: Apply MapReduce, YARN, and Spark concepts to process and analyze Big Data in distributed environments.

CO4: Apply Pig, Hive, and HBase for Big Data analysis and management

CO5: Understand Machine Learning concepts and NoSQL databases used in Big Data applications

UNIT-I

Introduction to Big Data and Hadoop: Types of Digital Data, Introduction to Big Data, Architecture and Applications, Characteristics of Big Data - 5 V's, Big Data Analytics and types, History and Evolution of Hadoop, Apache Hadoop, Analyzing Data with Hadoop, Hadoop Streaming, Hadoop Ecosystem, IBM Big Data Strategies - Introduction to Info Sphere, Big Insights.

UNIT-II

HDFS (Hadoop Distributed File System): The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Reading data from Hadoop URL, Writing data, Data flow in HDFS, Data Ingestion with Flume and Scoop and Hadoop archives.
Hadoop I/O: Compression, Serialization, Avro and File-Based Data Structures.

UNIT-III

Map Reduce: Concepts of Map Reduce, Anatomy of a Map Reduce Job Run, Hadoop Storage &

Processing Daemons, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features. YARN Architecture, Workflow, Features and Limitations.

Spark: Architecture of Spark, Spark vs Map Reduce, Advantages and Applications

UNIT-IV

Hadoop Ecosystem Tools: Pig, Hive, HBase

Pig: Introduction to PIG, Architecture, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User-Defined Functions, Types of UDF's, Data Processing Operators, Datatypes in Pig, Advantages and Limitations of Pig.

Hive: Architecture, Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, Querying Data and User Defined Functions

HBase: Basics, Concepts, Clients, Examples, Hbase versus RDBMS.

UNIT-V

NoSQL : Introduction to NoSQL, Types, NoSQL vs RDBMS, Characteristics and Applications

Machine Learning and Big Data: Role of Big Data in Machine Learning, Collaborative Filtering, Case Study.

Suggested Reading:

1. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reily Media, 2012
2. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.
3. Vikram Pudi and P. Radha Krishna, "*Big Data Analytics"*, Oxford University Press, 2016.
4. Uma N. Dulhare, Khaleel Ahmad, Khairol Arnali Bin Ahmad, Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications, Scrivener Publishing, Wiley, 2020
5. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley, 2013
6. Michael Berthold, David J. Hand, "Intelligent Data Analytics", Springer, 2007
7. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC Press (2013)

PC502CS	**PRINCIPLES OF PROGRAMMING LANGUAGES**					
Prerequisites	-		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. Describe the fundamental concepts, paradigms, syntax and semantics of programming languages.
2. Analyze language design principles, language evaluation criteria and implementation techniques.
3. Explain concepts of data types, bindings, scope, type checking, expression, and control structures in programming language design.
4. Examine object-oriented, concurrent, logic, and functional programming languages.
5. Compare programming language features and select appropriate paradigms and languages for solving real-world computing problems.

Course Outcomes:

CO1: Explain programming language concepts, paradigms, syntax representation methods and semantic models.
CO2: Analyze data types, bindings, type systems, expressions, and control structures used in programming languages.
CO3: Apply concepts of subprograms, parameter passing mechanisms, scope rules, and abstraction techniques in program design.
CO4: Demonstrate object-oriented, concurrent and logic programming concepts using appropriate languages and tools.
CO5: Evaluate and compare functional and scripting language solutions using languages such as, F#, Python, Clojure & Scala and understand their suitability for different applications.

UNIT-I

Preliminary Concepts: Reasons for studying, concepts of Programming languages, Programming domains, Language Evaluation Criteria. Influence on Language design, Language categories, Programming Paradigms, programming environments. Syntax and Semantics: general problem of describing Syntax and Semantics, formal methods of describing syntax – BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars.

UNIT-II

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names,

Variable, concept of binding, type checking, strong typing, type compatibility, named constants. Expression and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures, Selection, Iteration

UNIT-III

Subprograms: Fundamental of Subprograms, Design issues of subprogram and operations, local referencing environments, parameter passing methods, overloaded subprograms, Generic subprograms, Design issues for functions, user defined overloaded operators, Coroutines.

UNIT-IV

Abstract types: Introduction to Data Abstraction and encapsulation, design issues for Abstract Data types (ADT), language examples, parameterized Abstract data types ADT, object oriented programming in C++, Java, C#, Kotlin & Rust. Concurrency: Introduction to subprogram level concurrence, Semaphores, monitors, message passing, Java threads, C# threads. Exception handling in C++, C#, Python, Java & Rust.
Logic Programming language: Introduction and overview of logic programing, basic elements of prolog, application of logic programming.

UNIT-V

Functional Programming Languages: Introduction, fundamentals of FPL, F#, Scala, **Clojure**, and Python functional programming. Application of functional programming languages and comparison of functional and imperative languages. Scripting Language: pragmatics, key concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope

Suggested Readings:
Suggested Readings:

1) Concepts of Programming languages Robert W. Sebesta 8/e, Pearson Education, 2008 / 11th Edition, 2020
2) Programming Language Design Concepts, D. A. Watt, Wiley Dreamtech, 2007 / Reprint 2015

References:

1) Programming languages 2nd Edition, A. B. Tucker, R. E. Noonan, TMH
2) Programming Kotlin Venkat Subramaniam, Pragprog publications, 2019
3) The Rust Programming language Steve Klabnik, Carol Nichols. Nostarch Press, 2019

4) Functional Python Programming (PythonFP) Steven F. Lott Packtpub publication 3rd Edition, 2024
5) Programming Scala Martin odersky, Lex Spoon, Bill Venners, Oreilly publisher, 3rd Edition 2020
6) **Programming Clojure, Alex Miller, Stuart Holloway, Aaron Bedra, pragprog publication, 3rd Edition, 2018**

PE622CS	**DESIGN THINKING**					
Prerequisites	-		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To familiarize students with design thinking concepts and principles
2. To ensure students can practice the methods, processes and tools of design thinking
3. To ensure students can apply the design thinking approach and have ability to model real world situations
4. To enable students to analyze primary and secondary research in the introduction to design thinking.
5. To learn tools for prototype and testing

Course Outcomes:

CO1. Understand the basics of Design thinking.
CO2. Learn the steps involved in design process
CO3. Understand the different phases of design thinking
CO4. Generate ideas and find solutions
CO5. Use tools for prototype and testing.

UNIT-I

DESIGN THINKING FOR INNOVATION
Introduction to Design Thinking, Understanding the principles of Design Thinking, Business Model Innovation, Challenges Best-Suited for Design Thinking, Product Life Cycle

UNIT-II

PROCESS OF DESIGN
Introduction Design Process, Four Step, Five Step, Twelve Step, Creativity and Innovation in Design Process, Design limitation, Creative Thinking, Lean Canvas Model and other Business Models

UNIT-III

PHASES IN DESIGN THINKING
Understand, Observe, Define, Ideate, Prototype, Test, Reflect Problem Statement, Empathy, The 5 Whys, Stakeholder map, Empathy map, Personas, Peer observation, Trend analysis

UNIT-IV

SOLUTION/IDEA GENERATION

Story Telling, Context mapping, Critical items diagram, Brainstorming, Matrix and Voting methods, Analogies, Benchmarking, Utility maps

UNIT-V

TOOLS AND TECHNIQUES FOR PROTOTYPE AND TEST

Types of Prototype, Exploration Map, Blueprint, MVP, Testing Sheets, Solution Feedback Capturing Tools, Structured Usability Testing, A/B Testing, Design Thinking Applications Case Studies

Suggested Readings:

1. Gavin Ambrose and Paul Harris – Design Thinking, AVA Publishing, 2010
2. David Raizman – History of Modern Design, 2nd edition, Laurence King Publishing Ltd., 2010
3. Michael Lewrick, Patrick Link, Larry Leifer – The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods, Wiley Publishing
4. Design Thinking for Dummies – Wiley
5. Tom Kelley, Jonathan Littman – Ten Faces in Innovation, Currency Books, 2006
6. G. Pahl, W. Beitz, J. Feldhusen, KH Grote – Engineering Design: A Systematic Approach, 3rd edition, Springer, 2007
7. The Field Guide to Human-Centered Design – by Design Kit

PE511AI	**DATA VISUALIZATION**					
Prerequisites	Python or R Programming Descriptive Statistics		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. Understand the principles of data visualization, visual perception, and effective visual communication
2. Apply visualization techniques to analyze, explore, and represent different types of data.
3. Design interactive and meaningful visualizations using appropriate visualization methods and tools.
4. Evaluate visualization designs based on usability, perception, cognition, and user-centered design principles
5. Explore recent trends, advanced visualization techniques, and their applications in data analytics and decision-making.

Course Outcomes:

CO1: Design and create data visualizations

CO2: Conduct exploratory data analysis using visualization.

CO3 Craft visual presentations of data for effective communication.

CO4: Use knowledge of perception and cognition to evaluate visualization design alternatives

CO5: Design and evaluate color palettes for visualization based on principles of perception.

UNIT-I

Introduction of visual perception, visual representation of data, Gestalt principles, information overloads. Creating visual representations, visualization reference model, visual mapping, visual analytics, Design of visualization applications

UNIT-II

Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multi-dimensional data, text and text documents, Levels of Text representations, Vector Space Model, Single document visualization, document collection, Extended Text .

UNIT-III

Visualization of groups, trees, graphs, clusters, networks, Displaying Hierarchical structure, Displaying arbitrary graphs/Networks, other issues software, Metaphorical visualization.

UNIT-IV

Design Effective Visaualization Steps in designing visualization problems in designing Evaluating Visualization: user task used characteristics data Characteristics. visualization Characteristics. Structure for evaluating visualization.

UNIT-V

Recent trends in various perception techniques, Various visualization techniques, data structures used in data visualization

Suggested Reading:

1. Ward, Grinstein Keim, "Interactive Data Visualization: Foundation, Techniques, and Applications', Natick: A K Peters Ltd.
2. E. Tufte, "The Visual Display of Quantitative Information", Graphics Press.
3. Dirken Jos, "Expert Data Visualization", Packt Publishing Ltd.
4. Stephanie Evergreen, "Effective Data Visualization: The right chart for the right data", SAGE publications.

PC501CS/PE	**SOFTWARE ENGINEERING**					
Prerequisites	-		L	T	P	C
			3	-	-	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objective:

1. To introduce the fundamental concepts of software engineering process frameworks, process models, and agile methodologies to select appropriate models for given project scenarios.
2. To illustrate software engineering principles and systematic requirements engineering techniques to gather, analyze, and validate user requirements effectively.
3. To demonstrate analysis and design modelling approaches including data, object-oriented, scenario-based, and behavioural models to transform requirements into design.
4. To describe architectural styles, component-level design, and user interface design principles to develop modular, maintainable, and user-centric software systems.
5. To impart testing strategies, debugging techniques, software metrics, and quality assurance standards like ISO 9000 to ensure delivery of reliable software products.

Course Outcomes:

Student will be able to

CO1. Acquire working knowledge of alternative approaches and techniques for each phase of software development

CO2. Judge an appropriate process model(s) assessing software project attributes and analyze necessary requirements for project development eventually composing SRS

CO3. Acquire skills necessary as an independent or as part of a team for architecting a complete software project by identifying solutions for recurring problems exerting knowledge on patterns

CO4. Concede product quality through testing techniques employing appropriate metrics by understanding the practical challenges associated with the development of a significant software system

CO5. Apply software engineering principles of agility, user interface design, and deployment to develop user-centric software solutions while adapting to changing requirements and ensuring effective product delivery.

UNIT-I

Introduction to Software Engineering: A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Models, Personal and

Team Process Models, Process Technology, Product and Process.
An Agile view of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II
Software Engineering Principles: SE Principles, Communication Principles, Planning Principles, Modeling Principles, Construction Principles, Deployment.
System Engineering: Computer-based Systems, The System Engineering Hierarchy, Business Process Engineering, Product Engineering, System Modeling.
Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use-Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT-III
Building the Analysis Model: Requirements Analysis Modeling Approaches, Data Modeling Concepts, Object-Oriented Analysis, Scenario-based Modeling, Flow-oriented Modeling, Class-based Modeling, Creating a Behavioral Model.
Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design.

UNIT-IV
Creating an Architectural Design: Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design.
Modeling Component-Level Design: Definition of Component, Designing Class-based Components, Conducting Component-level Design, Object Constraint Language, Designing Conventional Components.
Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

UNIT-V
Testing: Strategies: A Strategic Approach to Conventional Software Testing, Test Strategies for O-O Software.
Tactics: Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing, O-O Testing Methods.
Debugging: Debugging Techniques, The Art of Debugging.
Product Metrics: A Framework for Product Metrics, Metrics for each phase of software development.
Software Quality: Definition, Quality Assurance: Basic Elements, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO9000 Quality Standards, SQA Plan.
DevOps: Introduction to DevOps, what is DevOps, Importance and benefits of DevOps, Principles and practices, DevOps lifecycle for business Agility and continuous testing. and

practices, DevOps lifecycle for business Agility and continuous testing.

Suggested Readings:

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw Hill, 2020
2. Ali Behforooz and Frederick J. Hudson, Software Engineering Fundamentals, Oxford University Press, 1996
3. Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2008
4. Deepak Gaikwad, Viral Thakkar, "DevOps Tools from Practitioner's viewpoint", Wiley Publications, 2019.

ES---EC	**SIGNALS AND SYSTEMS ESSENTIALS**					
Prerequisites			L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives

1. Analyze basic concepts related to continuous time signals and systems, mathematical representation of periodic signals.
2. Familiarize with basic operations on signals and mathematical representation of aperiodic signals using Fourier and Laplace transform.
3. Analyze basic concepts related to discrete time signals and systems, mathematical representation discrete time signals.
4. Describe the concept of Z- Transform and its properties and illustrate their applications to analyze systems.
5. Define convolution, correlation operations on continuous and discrete time signals.

Course Outcomes :

1. Define and differentiate types of signals and systems in continuous and discrete time.
2. Apply the properties of Fourier transform for continuous time signals.
3. Relate Laplace transforms to solve differential equations and to determine the response of the Continuous Time Linear Time Invariant Systems to known inputs.
4. Apply Z-transforms for discrete time signals to solve Difference equations.
5. Obtain Linear Convolution and Correlation of discrete time signals with graphical representation.

UNIT I

Definitions and classifications: Classification of signals. Elementary continuous time signals, Basic operations on continuous-time signals. Classification of continuous-time systems: continuous time &discrete time systems, lumpedparameter & distributed –parameter systems, static & dynamic systems, causal & non-causal systems, Time-invariant & time-variant systems, stable & unstable systems.

UNIT II

Representation of Continuous-time signals: Analogy between vectors and signals, orthogonality and completeness. Fourier series Analysis of Continuous-time signals: Fourier series – Existence of Fourier series, Trigonometric and Exponential Fourier series, computational formulae, symmetry conditions, complex Fourier spectrum.

UNIT III

Continuous-time Fourier Transform (FT): The direct and inverse FT, existence of FT, Properties of FT, FT of standard signals, properties of FT, The Frequency Spectrum. Linear Convolution of continuous time signals: Graphical interpretation, properties of convolution, Correlation between continuous-time signals: Auto and Cross correlation, graphical interpretation, properties of correlation.

UNIT IV

Discrete-time signals and systems: Sampling, Classification of discrete-time signals, Basic operations on discrete time signals, Classification of discrete time systems, properties of systems. Linear Convolution of discrete time signals: Graphical interpretation, properties of discrete convolution Fourier analysis of discrete-time signals: Discrete-time Fourier transform (DTFT),properties of DTFT, Transfer function, Discrete Fourier transform properties of DFT.

UNIT V

Z-Transform analysis of signals & systems: The direct Z transform, Region of convergence, Z-plane and S-plane correspondence. Inverse Z transform, Properties of Ztransforms. Solution to linear difference equations, Linear constant coefficient systems, System transfer function.

Suggested Text Books:

1. Alan V O P Penheim, A. S. Wlisky, Signals and Systems, 2nd Edition, Prentice Hall.

2. Rodger E. Ziemer, William H Trenter, D. Ronald Fannin, Signals and Systems, 4th Edition, Pearson 1998.

3. Douglas K. Linder, Introduction to Signals and Systems, McGraw Hill, 1999. Reference Text Books: 1. P. Ramesh babu, R Ananada Natarajan, Signals and Systems, SCITECH, 3rd edition 2009.

PE512AI	**MOBILE APPLICATION DEVELOPMENT**					
Prerequisites	-		L	T	P	C
			3	0	-	-
Evaluation	CIE	30 Marks	SEE		70 arks	

Course Objectives

1. To understand Android platform fundamentals, architecture, and application development tools.
2. To design and develop Android user interfaces using activities, intents, fragments, and UI components.
3. To understand Android data storage techniques using shared preferences, files, and content providers.
4. To develop Android applications using services, multithreading, networking, and telephony features.
5. To build advanced mobile applications using location-based services, Google Maps, and basic iOS development concepts.

Course Outcomes

CO1: Understand the basics of Android devices and Platform.

CO2: Acquire knowledge on basic building blocks of Android programming required for App development.

CO3: Understand persistence Data storage mechanism in Android.

CO4: Understand advanced application concepts like Animations and Google Maps services etc

CO5: Design and develop iOS mobile applications by applying location-based services, maps, navigation, data persistence and user interface design principles.

UNIT-I

Introduction: Introduction to mobile application development, Android platform: Android platform features and architecture, versions, comparison added features in each version, ART (Android Runtime), ADB (Android Debug Bridge).
Development environment/IDE: Android studio and its working environment, gradle build system, emulator setup.
Application anatomy: Application framework basics: resources, layout, values, asset XML representation and generate R.javafile, Android manifest file, creating a simple application

UNIT-II

GUI for Android: Introduction to activities, activities life-cycle, Intent: intent object, intent filters, adding categories, linking activities, Views and View Groups: Basic views, picker views, adapter views, Menu, App Bar etc, basics of screen design; different layouts, App widgets.
Lollipop Material design: new themes, new widgets, Card layouts. RecyclerView
Fragments: Introduction to activities, activities life-cycle

UNIT-III

Different Data persistence schemes: Shared preferences, File Handling, Managing data using Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference.
Content providers: User content provider, Android in build content providers.

UNIT-IV

Services: introduction to services – local service, remote service and binding the service, the communication between service and activity, Intent Service.
Multithreading: Handlers, AsyncTask
Broadcast receivers: Local Broadcast Manager, Dynamic broadcast receiver, System Broadcast. Pending Intent, Notifications
Telephony Manager: Sending SMS and making calls

UNIT-V

Location based services: Displaying Maps, Obtaining the Maps API Key , Displaying the Zoom Control, Changing Views, Navigating to a specific Location Getting the Location that was Touched Geocoding and Reverse Geocoding Getting location data and Monitoring a Location.
Mobile Application Development for iOS:
Introduction, iOS Navigation and Interface Design, Persistent Data in iOS, Tables in iOS: Navigation and Information Display

Suggested Reading:

1. Dawn Griffiths, David Griffiths,"Head First: Android Development" ,OReilly2015,ISBN: 9781449362188
2. J.F.DiMarzio's, "Android 4 Application Development".
3. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.

PC551AI	**MACHINE LEARNING LAB**					
Prerequisites	Probability & Statistics		L	T	P	C
			0	0	2	1
Evaluation	CIE	25 Marks	SEE		50 Marks	

Course Objectives:

1. To provide hands-on experience in implementing machine learning algorithms using Python and relevant libraries.
2. To understand data preprocessing, visualization, and feature engineering techniques.
3. To develop classification and regression models using supervised learning algorithms.
4. To implement clustering, dimensionality reduction, neural networks, and deep learning models.
5. To apply machine learning techniques to real-world datasets and evaluate model performance.

Course Outcomes:

CO1: Implement data preprocessing and exploratory data analysis techniques on datasets.

CO2: Develop and evaluate supervised learning models for classification and regression problems

CO3: Apply unsupervised learning techniques such as clustering and dimensionality reduction

CO4: Build neural network for prediction and classification tasks.

CO5: Build deep learning models for prediction and classification tasks.

List of Experiments to be Performed

1. Data Preprocessing and Exploratory Data Analysis:
   - Handling missing values
   - Data normalization and standardization
   - Data visualization
2. Implementation of Linear Regression for predictive analysis.
3. Implementation of Logistic Regression for classification problems.

4. Implementation of Classification Algorithms:
   - K-Nearest Neighbors (KNN)
   - Decision Tree Classifier
   - Naïve Bayes Classifier
5. Implementation of Support Vector Machine (SVM) for binary and multiclass classification.
6. Dimensionality Reduction using:
   - Principal Component Analysis (PCA)
   - Kernel PCA
7. Implementation of Clustering Algorithms:
   - Centroid based Clustering-K-Means
   - Hierarchical Clustering
8. Implementation of Reinforcement Learning:
   - Markov Decision Process (MDP)
   - Q-Learning
9. Implementation of Artificial Neural Networks:
   - Multilayer Perceptron (MLP)
   - Backpropagation Learning
10. Implementation of Deep Learning Models:
    - Convolutional Neural Network (CNN) for image classification

PC 552DS	BIG DATA ANALYTICS LAB					
Prerequisites	-		L	T	P	C
			0	0	2	1
Evaluation	CIE	25 Marks	SEE		50 Marks	

Course Objectives

1. To provide practical knowledge of Hadoop Distributed File System (HDFS) and Hadoop ecosystem tools.
2. To develop skills in implementing MapReduce applications for big data processing.
3. To enable students to perform data manipulation and analysis using Apache Pig.
4. To introduce PIG Latin and HiveQL to process Big Data
5. To develop the ability to create and use User Defined Functions (UDFs) in Pig and Hive.

Course Outcomes

CO1: Understand Hadoop working environment.
CO2: Develop and execute MapReduce programs for processing and analyzing large datasets.
CO3: Write scripts using PIG to solve real world problems.
CO4: Design and execute HiveQL queries for data storage, retrieval, sorting, grouping, and aggregation.
CO5 : Apply Hadoop ecosystem tools such as HDFS, Pig, and Hive to process Big Data.

List of Programs

1. Understanding and using basic HDFS commands.
2. Word count application using Mapper Reducer on single node cluster.
3. Working with files in Hadoop file system: Reading, Writing and Copying.
4. Write a Pig program to load a dataset and display its contents.
5. Write a Pig program to demonstrate the use of FOREACH and GENERATE operators.
6. Writing User Defined Functions / Eval functions for filtering unwanted data in Pig.
7. Retrieving user login credentials from /etc/passwd using Pig Latin.
8. Working with HiveQL: Create and Load Data into a Hive Table.
9. Demonstrate ORDER BY, SORT BY, GROUP BY, HAVING and Aggregate functions.
10. Writing User Defined Functions in Hive.
11. Write a Apache Spark program to create and display a DataFrame.
12. Write a Java Spark Program to filter records from a DataFrame based on a condition.

PC 552 CS	DESIGN AND ANALYSIS OF ALGORITHMS LAB					
Prerequisites	-		L	T	P	C
			0	0	2	1
Evaluation	CIE	25 Marks	SEE		50 Marks	

Course Objectives:

1. To learn the importance of designing an algorithm in an effective way by considering
2. space and time complexity
3. To learn graph search algorithms.
4. To study network flow and linear programming problems
5. To learn the dynamic programming design techniques.
6. To develop recursive backtracking algorithms.

Course Outcomes:

CO1: Apply graph Search Algorithms.

CO2: Design an algorithm in a effective manner

CO3: Apply iterative and recursive algorithms.

CO4: Implement optimization algorithms for specific applications.

CO5: Design optimization algorithms for specific applications

List of Programs:

1. Print all the nodes reachable from a given starting node in a digraph using BFS Method and Check whether a given graph is connected or not using DFS method.
2. Sort a given set of elements and determine the time required to sort the elements using following algorithms:
    - Merge Sort
    - Quick Sort
3. Implement Knapsack problem using
    - Brute Force Approach
    - Greedy Method
    - Dynamic Programming
4. Find Minimum Cost Spanning Tree of a given undirected graph using
    - Kruskal's algorithm
    - Prim’s algorithm
5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm
6. Implement Travelling Salesperson Problem using
    - Brute Force Approach
    - Dynamic Programming
7. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm
8. Implement the following using BackTracking
    - NQueen's problem , Hamiltonian Cycle, Graph Coloring

SCHEME OF INSTRUCTION & EXAMINATION
Common for B.E CSE(AI) and B.E CSE (AI & ML)
SEMESTER-VI

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs/Wk	
Theory Courses										
1	PC 601 CS	Quantum Computing	3	-	-	3	30	70	3	3
2	PC 602 CS	Nano Technology & Computing	3	-	-	3	30	70	3	3
3	PC 601 AI	Deep Learning	3	-	-	3	30	70	3	3
4	PE	Professional Elective –II	3	-	-	3	30	70	3	3
5	PC 605 CS	Computer Ethics	3	-	-	3	30	70	3	3
Practical/Laboratory Courses										
6	PC 651 AI	Deep Learning Lab	-	-	2	2	25	50	3	1
7	PC 651 CS	Quantum Computing Lab	-	-	2	2	25	50	3	1
8	PW 652 AI	Mini Project	-	-	4	4	25	50	3	2
9	SI 671 AI	Summer Internship*	-	-	-	-	-	-	-	-
Total			**15**		**8**	**23**	**225**	**500**	24	**19**

Profession Elective – II	
Course Code	**Course Title**
PC 601 IT/PE	Embedded Systems
PE741CS/PE	Cloud Computing
PE 622 DS	Data warehousing and Data Mining
PE 622 AI	Speech and Video Processing
PE 623 DS	Forecasting Techniques

Open Elective – I	
Course Code	**Course Title**
OE 601 AI	Fundamentals of Machine Learning(Not for CSE & IT Students & Allied Branches)

PC601CS	QUANTUM COMPUTING					
Prerequisites	Linear Algebra and Computer Science Programming		**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Course Objectives:

1. Understand the mathematical foundations and physical principles that form the basis of quantum computing.
2. Gain knowledge of fundamental computer science concepts, information theory, and quantum mechanics relevant to quantum information processing.
3. Explore quantum architectures, qubits, quantum gates, circuits, and hardware technologies used in quantum computers.
4. Study and analyze major quantum algorithms and their advantages over classical computational methods.
5. Examine the impact of quantum computing on modern cryptographic systems and understand emerging quantum-resistant security techniques.

Course Outcomes:

After successful completion of the course, students will be able to:

1. Apply linear algebra, complex numbers, and quantum physics principles to model and analyze quantum systems.
2. Explain the foundations of computer science, information theory, and quantum mechanics relevant to quantum computing.
3. Analyze quantum architectures, qubits, quantum gates, circuits, and hardware technologies.
4. Evaluate and apply fundamental quantum algorithms to solve computational problems efficiently.
5. Assess the impact of quantum computing on modern cryptography and emerging quantum-safe security solutions.

UNIT – I

Linear Algebra Basics: Basic Algebra, Matrix Math, Vectors and Vector Spaces, Complex Numbers: Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers.

Basic Physics for Quantum Computing: Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement.

UNIT – II

Computer Science for Quantum Computing: Computational Complexity, Coding Theory, Logic Gates, Information Theory: Basic Probability, Set Theory, Information Theory, Quantum Information.

Quantum Theory: Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram, Quantum Entanglement and QKD.

UNIT – III

Quantum Architecture: Qubits, Quantum Gates, quantum logic, Quantum Circuits, The D-Wave Quantum Architecture.
Quantum Hardware: Addressing Decoherence, Topological Quantum Computing, Quantum Essentials, Quantum Networking.

UNIT – IV

Quantum Algorithms: Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein- Vazirani Algorithm, Simon's Algorithm, Shor's Algorithm, Grover's Algorithm.

UNIT – V

Quantum Computing and Cryptography: Current Asymmetric Algorithms, RSA, Diffie-Hellman, Elliptic Curve. The Impact of Quantum Computing on Cryptography: Asymmetric Cryptography, Specific Algorithms and Applications.

SUGGESTED READINGS:

1. Dr. Chuck Easttom, Quantum Computing Fundamentals, Pearson.
2. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press
3. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci
4. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. Basic Concepts. Vol. Basic Tools and Special Topics, World Scientific.
5. Pittenger A. O., An Introduction to Quantum Computing Algorithms.

PC602CS	**NANO TECHNOLOGY**					
Prerequisites	Quantum Mechanics Materials Science		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives

After successful completion of this course, the students will be able to:

1. Understand the limitations of classical CMOS technology, thermodynamic constraints, and the need for emerging nano-computing paradigms.
2. Explain the fundamental principles of quantum mechanics and their significance in nanoscale electronic devices.
3. Analyze the operation and characteristics of nanoscale electronic devices, including Single-Electron Transistors (SETs), Carbon Nanotube FETs (CNTFETs), and Graphene Nanoribbon (GNR)-based circuits.
4. Examine the principles of spintronic devices, memristive systems, and Quantum-dot Cellular Automata (QCA) for next-generation computing applications.
5. Design combinational and sequential logic circuits using Quantum-dot Cellular Automata (QCA) for nano-computing systems.

Course Outcomes

Upon successful completion of this course, students will be able to:

1. Explain the physical and thermodynamic limitations of classical CMOS technology and justify the need for nano-computing technologies.
2. Apply the principles of quantum mechanics to analyze the behavior of nanoscale electronic devices and quantum systems.
3. Analyze the operation, characteristics, and applications of SETs, CNTFETs, Graphene Nanoribbons, and other nanoscale electronic devices.
4. Evaluate the performance and applications of spintronic devices, memristors, and Quantum-dot Cellular Automata in emerging computing architectures.
5. Design QCA-based combinational and sequential logic circuits, including majority gates, memory elements, shift registers, and arithmetic logic units (ALUs), for nano-computing applications.

UNIT-I:

Foundations and Limits of Classical Computing Limits of Silicon CMOS: Sub-threshold leakage, short-channel effects, electromigration and the physical limits of Moore's Law. Thermodynamic Limits: Landauer's principle, power dissipation limits, and irreversible vs. reversible computing.

UNIT-II:

Quantum Mechanics Fundamentals: Wave-particle duality, Schrodinger wave equation, quantum tunnelling and electron confinement. Heisenberg Uncertainty Principle, Quantum Entanglement, Superposition, Spin & Angular Momentum, Perturbation Theory, and Feynman Path Integrals.

UNIT-III:

Nanoscale Electronic Devices Single-Electron Transistors (SET): Coulomb blockade, single-electron tunneling, and SET based logic gates. Carbon Nanotube Electronics: Carbon Nanotube FETs (CNTFETs), ballistic transport, Graphene Nanoribbons (GNR), and CNT-based logic circuits.

UNIT-IV:

Spintronics & Memristors: Spin-transfer torque (STT) devices, giant magnetoresistance (GMR), and memristive systems for neuromorphic nano-computing. Quantum-dot Cellular Automata (QCA) QCA

Basics: QCA cell structure, polarization states, electrostatic interactions, and clocking schemes (Switch, Hold, Release, Relax).

UNIT-V:

QCA Logic Design: Majority gates, invertors, wire crossings (coplanar and multilayer), and full-adder design. Sequential QCA: Memory cell design, shift registers, and QCA-based arithmetic logic units (ALUs).

Reference Books:

1. Nanoelectronics and Information Technology by Rainer Waser.
2. Introduction to Nanocomputing by Sahni, V.
3. Quantum-dot Cellular Automata by Michael Niemier.

PC601AI	**DEEP LEARNING**					
Prerequisites	Machine Learning		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To introduce basic concepts of artificial neural networks and multilayer perceptrons
2. To introduce basic concepts of CNN and VGG
3. . To introduce recurrent neural networks and LSTM's
4. To introduce auto encoders and GAN's
5. To understand the concepts, architectures, and applications of Autoencoders, Generative Adversarial

Course Outcomes:

CO1: Understand the problem of XOR seperability and activation functions in ANN's

CO2: Understand the problem of over fitting, under fitting, Gradient Descent and Stochastic Gradient Descet

CO3: Demonstrate understanding of CNN's and VGG's

CO4: Demonstrate understanding of RNN's and LSTM's

CO5: Use auto encoders and GAN's

UNIT-I

Introduction: Artificial intelligence, Machine learning, and Deep learning - Artificial intelligence -Machine learning – Learning representations from data - The "deep" in deep learning - Understanding how deep learning works, in three figures

Artificial Neural Networks: Introduction, Perceptron, XOR Gate, Perceptron Training Rule, Activation Functions.

Linear Neural Networks: Linear Regression, Implementation of Linear Regression, Softmax Regression, The Image Classification Dataset, Implementation of Softmax Regression

UNIT-II

Multilayer Perceptrons: Multilayer Perceptrons, Implementation of Multilayer Perceptrons, Model Selection, Under fitting and Over fitting, Weight Decay, Dropout, Forward Propagation, Backward Propagation, and Computational Graphs, Numerical Stability and Initialization, Considering the Environment, Predicting House Prices on Kaggle

Optimization Algorithms: Optimization and Deep Learning, Convexity, Gradient Descent, Stochastic Gradient Descent, Mini batch Stochastic Gradient Descent, Momentum, Adagrad, RMS Prop, Ada delta, Adam, Learning Rate Scheduling

UNIT-III

Introduction to Convolutional Neural Networks: Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters

Modern Convolutional Neural Networks: Deep Convolutional Neural Networks (AlexNet), Networks Using Blocks (VGG), Network in Network (NiN), Networks with Parallel Concatenations (GoogLeNet), Batch Normalization, Residual Networks (ResNet), Densely Connected Networks (DenseNet)

UNIT-IV

Recurrent Neural Networks: Sequence Models, Text Preprocessing, Language Models and the Dataset, Recurrent Neural Networks, Implementation of Recurrent Neural Networks from Scratch, Concise Implementation of Recurrent Neural Networks, Back propagation Through Time Modern **Recurrent Neural Networks:** Gated Recurrent Units (GRU),Long Short Term Memory (LSTM), Deep Recurrent Neural Networks, Bidirectional Recurrent Neural Networks, Machine Translation and the Dataset, Encoder-Decoder Architecture, Sequence to Sequence, Beam Search.

UNIT-V

Auto encoders: Types of Auto Encoders and its applications Generative Adversarial Networks: Generative Adversarial Network.

Transformers: The Transformer Architecture, Transformers for Vision (VIT), Bidirectional Encoder Representations from Transformers (BERT) models,

Deep Learning Applications

Suggested Readings:

1. J., & Gibson, A. Deep Learning: A Practitioner's Approach Patterson, O'Reilly Media, Inc 2017.
2. Ian Goodfellow and Yoshua Bengio and Aaron Courville Deep Learning: An MIT Press Book, 2019
3. Simon Haykin, Neural Networks and Learning Machines, 3rd edition, Pearson Prentice Hall
4. Francois Chollet, Deep Learning with Python, Manning Publications, 2018.
5. Uma N Dulhare, Essam Houssain- Deep Learning and Computer Vision: Models and Biomedical Applications:Vol1 & Vol 2 - Springer Publication 2025

PC 605 CS	Computer Ethics					
Prerequisites			L	T	P	C
			3	-	-	3
Evaluation	CIE	30 Marks	SEE		70	

Course Objectives (COBs)

After successful completion of this course, students will be able to:

1. Understand the basic concepts of morality, ethics, and law in computing.
2. Learn professional ethics and ethical responsibilities of computing professionals.
3. Understand privacy, security, anonymity, and ethical issues in the digital world.
4. Gain knowledge of intellectual property rights, cybercrime, and computer ethics.
5. Examine the ethical and social impact of emerging technologies such as AI, IoT, and social media.

Course Outcomes (COs)

After successful completion of this course, the students will be able to:

CO1: Explain the concepts of morality, ethics, law, ethical theories, and their significance in computing and information technology.

CO2: Apply professional codes of conduct and ethical decision-making principles to resolve ethical issues encountered in computing professions.

CO3: Analyze ethical, legal, and social issues related to anonymity, privacy, civil liberties, and data protection in digital environments.

CO4: Analyze intellectual property rights, computer crimes, digital evidence, and ethical responsibilities in computer forensic investigations.

CO5: Evaluate the ethical and societal implications of emerging technologies such as Artificial Intelligence, IoT, online social networks, biometrics, surveillance, and digital media.

UNIT-I:

Morality and Ethics: Moral code of behaviour, Purpose of Law, Penal code, Morality and the law; Ethical theories, Functional definition of ethics, Ethical reasoning and decision making, codes of ethics, technology and values.

UNIT-II:

Ethics and Professions: Professional code of conduct, Professional decision making and ethics, Whistleblowing, harassment and discrimination, ethical and moral implications of profession.

UNIT-III:

Anonymity, Privacy and Civil Liberties: Anonymity in the Internet, its advantages and disadvantages, Types and value of Privacy, Privacy protection and civil liberties, implications for privacy of the Aadhaar database,Workplace privacy and surveillance, Ethics for privacy, Ethical and legal basis for privacy. Case studies of Cambridge Analytica, SECC in India.

UNIT-IV:

Intellectual Property Rights, Computer Crime and Ethics: Computer Products and services, foundations of Intellectual Property Rights (IPR), Ownership, IPR crimes, Protecting computer software under IPR, Transnational issues. Digital evidence, preserving digital evidence, analysis of digital evidence, ethical implications and responsibilities in computer forensic investigations.

UNIT-V:

Social Implications of Information Technology: Advances in Artificial Intelligence, AI and ethics, bias in AI algorithms, Online social networks, Ethical issues in online social networks, security and crimes in online social networks; Internet of Things (IoT) and Location tracking, ethics of tracking, ethics of personalised search/ advertisements in online social networks and

search engines, self-driving vehicles and ethical/legal issues, surveillance and ethical issues, use of biometrics for authentication, ethical implications of biometric technologies, fake news and implications of its spread via social media platforms, digital divide and its implications, sexual predators and pornography on the Internet and its implications.

Text Book:

1. Ethical and Social Issues in the Information Age, Joseph Migga Kizza, 6th ed. 2017 Edition, ISBN-10: 1447149890, ISBN-13: 978-1447149897

Reference Books:

[1] https://theprint.in/politics/exclusive-inside-story-cambridge-analytica-actually india/44012/
[2] https://thewire.in/tech/what-we-know-and-what-we-dont-about-what-cambridge analytica-did-in-india
[3] https://www.huffingtonpost.in/entry/aadhaar-national-social-registry-database-modi in 5e6f4d3cc5b6dda30fcd3462
[4] https://www.huffingtonpost.in/entry/one-year-after-aadhaar-judgement-we-are-still-not listening in 5d8bb628e4b0ac3cdda24659
[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6596313/
[6] https://yjolt.org/sites/default/files/21_yale_j.l._tech._106_0.pdf
[7] https://hbr.org/2019/10/what-do-we-do-about-the-biases-in-ai
[8] https://www.nytimes.com/2019/11/19/technology/artificial-intelligence-bias.html
[9] https://aibusiness.com/three-notable-examples-of-ai-bias/
[10] https://thewire.in/tech/artificial-intelligence-has-a-gender-bias-problem-just-ask-siri
[11] https://www.theguardian.com/books/2019/feb/02/age-of-surveillance-capitalism shoshana-zuboff-review
[12] https://www.nature.com/articles/d41586-018-07135-0
[13] https://www.newyorker.com/science/elements/a-study-on-driverless-car-ethics-offers-a troubling-look-into-our-values
[14] https://towardsdatascience.com/how-ethical-is-facial-recognition-technology 8104db2cb81b
[15] https://www.technologyreview.com/f/615068/facial-recognition-european-union temporary-ban-privacy-ethics-regulation/
[16]https://www.meity.gov.in/writereaddata/files/Information%20Technology%20(Intermediary%20Guielines%20and%20Digital%20Media%20Ethics%20Code)%20Rules%2C%202021%20(updated%2006.04. 2023)-.pdf
[17] https://www.meity.gov.in/content/cyber-laws
[18]https://www.tec.gov.in/pdf/Studypaper/AI%20Policies%20in%20India%20A%20status%20Paper%20 final.pdf

PC 601 IT/PE	**EMBEDDED SYSTEMS**					
Prerequisites	-		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To understand the architecture of 8051 microcontrollers.
2. To understand the various applications of Embedded Systems using the concepts of Interfacing.
3. To familiarize with smart sensors and understand various sensor applications.
4. To learn the concepts of RTOS and the design process using RTOS.
5. To familiarize with the design principles of SOC.

Course Outcomes:

Student will be able to

1. Study and analysis of embedded systems.
2. Design and develop embedded systems (hardware, software and firmware) Analyze, real time systems using RTOS and develop applications.
3. Apply knowledge to interface various sensors and its applications in embedded systems.
4. Understand principles of SOC design
5. Understand principles of Processors

UNIT – I

Embedded Computing: Introduction, Complex Systems and Microprocessor, Embedded System Design Process, Formalisms for System Design, Design Examples. Microprocessors and Microcontrollers: Microprocessors and Microcontrollers, The 8051 Architecture: Introduction, 8051 Microcontroller Hardware, Input/output Ports and Circuits, External Memory. Counter and Timers, Serial data Input/output, Interrupts.

UNIT – II

Programming using 8051. Data Transfer and Logical Instructions. Arithmetic Operations, Decimal Arithmetic. Jump and Call Instructions, Applications: Interfacing with Keyboards, Displays, D/A and A/D Conversions, Multiple Interrupts, Serial Data Communication. Bus protocols: I^2Cbus and CAN bus.

UNIT – III

Smart Sensors Introduction – Primary Sensors – Excitation – Amplification – Filters – Converters –Compensation– Information Coding/Processing - Data Communication – Standards for Smart Sensor Interface–The Automation. Sensors–Applications, Introduction–On-board Automobile Sensors(Automotive Sensors)–Home Appliance Sensors–Aerospace Sensors—Sensors form Manufacturing–Sensors for environmental Monitoring

UNIT – IV

Introduction to Real-Time Operating Systems: Tasks and task states, tasks and data, semaphores, and shared data; message queues, mailboxes and pipes, timer functions, events, memory management,interrupt routines in an RTOS environment. Basic Design Using a Real-Time Operating System: Principles, semaphores and queues, hard real- time scheduling considerations, saving memory and power.

UNIT – V

Introduction to the System Approach System Architecture,Components of the system, Hardware & Software, Processor Architectures, Memory and Addressing. System levels interconnection, An approach for SOC Design, System Architecture and Complexity.

Suggested Readings:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay,—The 8051 Microcontroller and Embedded Systems using Assembly and C ‖, Prentice HallIndia, 2nd Edition.
2. D.Patranabis——Sensors and Transducers‖– PHI Learning Private Limited.
3. Wayne Wolf, "Computers and Components",Elsevier,Second Edition.
4. KennethJ.Ayala,"The8051Microcontroller",Third Edition,,Thomson.
5. DavidE.Simon," An Embedded Software Primer", PearsonEducation

PE742CS	**CLOUD COMPUTING**					
Prerequisites			L	T	P	C
			-	-	-	-
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives

1. Understand cloud computing concepts and architecture
2. Learn cloud service and deployment models
3. Study virtualization and cloud infrastructure
4. Explore cloud security and data protection
5. Analyze cloud platforms and migration strategies

Course Outcomes

After completing this course, the student will be able to:

1. Explain cloud computing characteristics and challenges
2. Apply cloud service and deployment models
3. Use virtualization and cloud management tools
4. Evaluate cloud security and encryption techniques
5. Use online cloud services and plan cloud migration

Unit - I

Introduction - Historical Development, System Models for Distributed and Cloud Computing: Cloud Computing Architecture – The Cloud Reference Model – Cloud Characteristics
Cloud Deployment Models: Public, Private, Community, Hybrid Clouds, Challenges and Risks, Cloud Delivery Models: IaaS, PaaS, SaaS.

Unit - II

Virtual Machines & Cloud Computing Mechanism: Levels of Virtualization, Virtualization Structures/Tools, and Mechanisms.
Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization
Desktop virtualization: Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Cloud Infrastructure Mechanism: Cloud Storage, Cloud Usage Monitor.

Unit - III

State Management Database – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System, State Management Database – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System.

Unit - IV

Cloud Security and Trust Management, Data Security in the Cloud: An Introduction to the Idea of Data Security, The Current State of Data Security in the Cloud,
CryptDB: Onion Encryption layers DET, RND, OPE, JOIN, SEARCH, HOM, and Homomorphic Encryption**.**

Unit - V

Case Studies: Google App Engine (GAE) – GAE Architecture – Functional Modules of GAE –

Amazon Web Services (AWS) – GAE Applications – Cloud Software Environments – Eucalyptus – Open Nebula – Open Stack.

Migrating into a Cloud: Introduction, Challenges while migrating to Cloud, Broad approaches to migrating into the cloud, why migrate – deciding on cloud migration, the Seven-step model of migration into a cloud, Migration Risks and Mitigation

Suggested Reading:

1. Anthony http://T.Velte, Toby J. Velte, Robert Elsenpeter, Cloud computing a practical approach - TATA McGraw-Hill, New Delhi – 2010
2. Thomas Erl, ZaighamMahmood, Ricardo Puttini, Cloud Computing, Concept, Technology and Architecture, Prentice Hall, 2013.
3. K. Chandrasekharan, Essentials of cloud Computing CRC press, 2014
4. Anthony http://T.Velte, Toby J. Velte, Robert Elsenpeter, Cloud computing a practical approach - TATA McGraw-Hill, New Delhi – 2010
5. Raluca Ada Popa, Catherine M.S. Redfield, NickolaiZeldovich, and Hari Balakrishnan, "CryptDB: Protecting Confidentiality with encrypted Query Processing", 23rd ACM Symposium on Operating Systems Principles (SOSP 2011), Cascais, Portugal October 2011.
6. Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing (Principles and Paradigms), Edited by John Wiley & Sons, Inc. 2011

PE622DS	**DATA WAREHOUSING AND DATA MINING**					
Prerequisites	-		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To identify the scope and essentiality of Data warehousing and Data Mining.
2. To analyze the data, data science lifecycle, data collection and cleaning, exploratory data analysis and visualization, statistical inference and prediction, and decision-making algorithms for respective applications.
3. To develop research interest towards advances in data mining
4. To learn classification techniques, including decision trees and Bayesian methods, for predictive modeling
5. To apply clustering, outlier detection and data mining techniques to solve real-world problems and analyze emerging trends

Course Outcomes:

CO1**:** Understand the basic concepts , measures in Data Mining & Data Warehousing

CO2: Understand and Extract knowledge using data preprocessing concepts in data mining

CO3: Extract the hidden pattern using Association Rule mining algorithms

CO4: Classify the data including complex and large datasets using classification algorithms

CO5: Build the model using clustering techniques and solve real world applications using various data mining techniques

UNIT – I

Data warehouse: Introduction to Data warehouse, Difference between operational database systems and data warehouses, Data warehouse Characteristics, Data warehouse Architecture and its Components, Extraction-Transformation-Loading, Logical(Multi-Dimensional), Data Modeling, Schema Design, Star and Snow-Flake Schema, Fact Constellation, Fact Table, Fully Addictive, Semi-Addictive, Non Addictive Measures; Fact-Less-Facts, Dimension Table Characteristics; OLAP Cube, OLAP Operations, OLAP Server Architecture-ROLAP, MOLAP and HOLAP.

UNIT - II

Introduction: Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Integration of a Data Mining System with a Database or Data Warehouse System, Major issues in Data Mining. Data Preprocessing: Need for Preprocessing the Data, Data Cleaning, Data Integration &Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

UNIT – III

Association Rules: Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIOIRI Algorithm, The Partition Algorithms, FP-Growth Algorithms, Compact Representation of Frequent Item Set- Maximal Frequent Item Set, Closed Frequent Item Se

UNIT – IV

Classification: Basic concepts, Classification techniques, Decision Trees-Decision tree Construction, Methods for Expressing attribute test conditions, Measures for Selecting the Best Split, Algorithm for Decision tree Induction ; Naive-Bayes Classifier, Bayesian Belief Networks; K- Nearest neighbor classification-Algorithm and Characteristics. Prediction: Accuracy and Error measures, Evaluating the accuracy of classifier or a predictor, Ensemble methods

UNIT - V

Clustering: Clustering Overview, A Categorization of Major Clustering Methods, Partitioning Clustering-K-Means Algorithm, PAM Algorithm; Hierarchical Clustering-Agglomerative Methods and divisive methods, Outlier Analysis, Data Mining Applications

Suggested Readings:

1. Jiawei Han, Micheline Kamber, Jin Pei, Data Mining: Concepts & Techniques, 3rd Edition. Morgon Koffman,2011
2. Vikram Pudi, P. Radha Krishna, Data Mining, Oxford University Press, 1st Edition, 2009
3. Learning and Big Data: Concepts, Algorithms, Tools and Applications-Uma N. Dulhare Khaleel Ahmad Khairol Amali Bin Ahmad
4. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2008

PE622 AI	**SPEECH AND VIDEO PROCESSING**					
Prerequisites	Digital Signal Processing (DSP) Linear Algebra		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives

1. Knowledge on speech and video processing techniques
2. To acquire knowledge of speech signal analysis, feature extraction, and recognition methods.
3. To learn video representation, compression, and processing techniques for multimedia applications.
4. To understand machine learning and deep learning approaches used in speech and video analytics
5. To develop the ability to design and implement speech and video processing solutions for real-world applications.

Course Outcomes

CO1: Describe the mechanisms of human speech production systems and methods for speech feature extraction

CO2: Understand basic algorithms of speech analysis and speech recognition.

CO3: Explain basic techniques in digital video processing, including imaging characteristics and sensors.

CO4: Apply motion estimation and object tracking algorithms on video sequence.

CO5: Apply Object Tracking & Segmentation

Unit-I

Speech processing concepts: The speech production mechanism, Discrete time speech signals, Pole-Zero modeling of speech, relevant properties of the fast Fourier transform for speech recognition, convolution, linear and nonlinear filter banks, spectral estimation of speech using DFT. Linear Prediction analysis of speech

Unit-II

Speech recognition: Real and Complex Cepstrum, application of cepstral analysis to speech signal, feature extraction for speech, static and dynamic feature for speech recognition,

robustness issues, discrimination in the feature space, feature selection, MFCC, LPCC, Distance measures, vector quantization models. Gaussian Mixture model, HMM.

Unit-III

Basics of Video Processing: Video formation, perception and representation: Principle of color video, video cameras, video display, pinhole model, CAHV model, Camera motion, Shape model, motion model, Scene model, two-dimensional motion models. Three-Dimensional Rigid Motion, Approximation of projective mapping.

Unit-IV

Motion estimation Techniques: Optical flow, motion representation, motion estimation criteria, optimization methods, pixel-based motion estimation, Block matching algorithm, gradient Based, Intensity matching, feature matching, frequency domain motion estimation, Depth from motion. Motion analysis applications: Video Summarization, video surveillance

Unit-V

Object tracking and segmentation: 2D and 3D video tracking, blob tracking, kernel based counter tracking, feature matching, filtering Mosaicing, video segmentation, mean shift based, active shape model, video shot boundary detection. Interframe compression, Motion compensation.

Suggested Reading:

1. Fundamentals of Speech recognition – L. Rabiner and B. Juang, Prentice Hall signal processing series.
2. Digital Video processing, A Murat Tekalp, Prentice Hall.
3. Discrete-time speech signal processing: principles and practice, Thomas F. Quatieri, Coth
4. Video Processing and Communications, Yao Wang, J. Osternann and Qin Zhang, Pearson Education
5. “Speech and Audio Signal Processing”, B.Gold and N. Morgan, Wiley.
6. “Digital image sequence processing, Compression, and analysis”, Todd R. Reed, CRC Press
7. “Handbook of Image and Video processing”, Al Bovik, Academic press, second Edition

PE623DS	**FORECASTING TECHNIQUES**					
Prerequisites	Time Series Analysis Probability & Statistics		L	T	P	C
			3	0	0	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives:

1. To Learn Basics concepts of Time series Analysis and Forecasting
2. To learn about Regression Models based on Time Series
3. To learn Non-Stationary and multivariate time series.
4. To understand ARMA, ARIMA, and Seasonal ARIMA models for analyzing and forecasting time series data.
5. To develop the ability to apply multivariate time series models and advanced forecasting techniques for real-world applications.

Course Outcomes:

CO1:Knowledge of basic concepts in time series analysis and forecasting

CO2:Understanding the use of time series models for forecasting and the limitations of the methods

CO3: Ability to criticize and judge time series regression models.

CO4:Distinguish the ARIMA modelling of stationary and non-stationary time series

CO5:Compare with multivariate times series and other methods of applications

UNIT – I

Introduction of time series analysis: Introduction to Time Series and Forecasting -Different types of data-Internal structures of time series, Models for time series analysis-Autocorrelation and Partial autocorrelation. Examples of Time series Nature and uses of forecasting-Forecasting Process-Data for forecasting-Resources for forecasting

UNIT – II

Statistics background for forecasting: Graphical Displays -Time Series Plots - Plotting Smoothed Data - Numerical Description of Time Series Data Use of Data Transformations and Adjustments-General Approach to Time Series Modeling and Forecasting, Evaluating and Monitoring Forecasting Model Performance.

UNIT – III

Time series regression model: Introduction Least Squares Estimation in Linear Regression Models - Statistical Inference in Linear Regression- Prediction of New Observations - Model Adequacy Checking Variable Selection Methods in Regression Generalized and Weighted Least Squares- Regression Models for General Time Series Data Exponential Smoothing-First order and Second order

UNIT – IV

Autoregressive Integrated Moving Average (arima) models: Autoregressive Moving Average (ARMA) Models - Stationarity and Invertibility of ARMA Models - Checking for Stationarity

using Variogram- Detecting Nonstationarity Autoregressive Integrated Moving Average (ARIMA) Models - Forecasting using ARIMA Seasonal Data - Seasonal ARIMA Models- Forecasting using Seasonal ARIMA Models Introduction Finding the "BEST" Model -Example: Internet Users Data- Model Selection Criteria - Impulse Response Function to Study the Differences in Models Comparing Impulse Response Functions for Competing Models.

UNIT – V

Multivariate time series models and forecasting Multivariate Time Series Models and Forecasting Multivariate Stationary Process- Vector ARIMA Models- Vector AR (VAR) Models - Neural Networks and Forecasting -Spectral Analysis - Bayesian Methods in Forecasting.

Suggested Reading

SUGGESTED READING

1. Time Series Analysis: Forecasting and Control – George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung (5th Edition, Wiley, 2015).
2. Time Series Analysis and Forecasting by Example – Søren Bisgaard and Murat Kulahci (Wiley, 2011).

Reference Books

1. Introduction to Time Series and Forecasting – Peter J. Brockwell & Richard A. Davis (3rd Edition, Springer, 2016).
2. Time Series Analysis and Applications with R – Jonathan D. Cryer & Kung-Sik Chan (Springer).
3. Time Series Analysis by James D. Hamilton (Princeton University Press).

OE 601 AI	FUNDAMENTALS OF MACHINE LEARNING					
Prerequisites	-		L	T	P	C
			3	-	-	3
Evaluation	CIE	30 Marks	SEE		70 Marks	

Course Objectives :

1. To introduce the fundamentals of Machine Learning, Data Science, and learning paradigms.
2. To develop skills in data analysis, preprocessing, visualization, and statistical techniques.
3. To impart knowledge of regression and classification methods for predictive modeling.
4. To enable the application of advanced machine learning algorithms for classification and prediction.
5. To develop practical skills in implementing, evaluating, and selecting machine learning models using Python-based tools.

Course Outcomes:

CO1. Design and implement machine learning solutions of classification, regression problems.

CO2. Evaluate and interpret the results of the machine learning algorithms.

CO3. Evaluate exploratory data analysis and Data preparation and preprocessing on different datasets.

CO4. Calculate Statistical measurements of the given data.

CO5. Analyze and identify the best algorithm matches for a given dataset.

UNIT - I

Introduction: Machine Learning Concepts, Use Machine Learning, and Types of Machine Learning Systems: supervised, unsupervised, semi-supervised, Reinforcement Learning, Batch and Online Learning, Main Challenges of Machine Learning.

UNIT - II

Descriptive Statistics: Data representation, types of data- nominal, ordinal, interval and continuous, central tendency- calculating mean mode median, mean vs median, variability, variance, standard deviation, Mean Absolute Deviation using sample dataset, finding the percentile, interquartile range, Box Plot, Outlier, whisker, calculating correlation, covariance.

Causation, Exploratory data analysis, Data preparation and preprocessing, Data visualization.

UNIT - III

Regression: Introduction to Regression analysis, measure of linear relationship, Regression with stats models, Determining coefficient, meaning and significance of coefficients, coefficient calculation with least square method, Types of regression, Simple Linear Regression, Using Multiple features, Polynomial Regression, Metrics for Regression: MSE, RMSE, MAE.

UNIT-IV

Classification: Classification problem, Probability based approach, Logistic Regression- log-odd, sigmoid transformation, Metrics: Confusion Matrix, Accuracy, Error Rate, Precision, Recall, ROC curve, Fl score, and introduction to gradient descent.

UNIT - V

Non Parametric & SVM classification: About Non parametric classification, Decision Trees: Entropy, Gain ratio, Information Gain, Splitting criteria, Ensemble Method: Introduction to Random Forest, Accuracy measure & performance Instance based learning- Introduction, KNN algorithm, Distance measures, model building, locally weighted regression, radial basis functions, SVM classifier, hyper-plane, slack variables, geometric transformation kernel trick, kernel transformation.

Suggested Readings:

1. Booz, Allen, Hamilton, The Field Guide to Data Science

2. AurélienGéron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly Media, 2017-03-10

3. Peter Harrington, Machine Learning in Action, Manning Publications

4. Python For Data Analysis by wes McKinny 2nd edition, O'REILLY publications.

5. Jason Brownlee data analysis for machine learning.

MC 601 CS	COMPUTER ETHICS					
Prerequisites			L	T	P	C
			3	-	-	-
Evaluation	CIE	50 Marks	SEE		-	

Course Objectives (COBs)

6. Understand the basic concepts of morality, ethics, and law in computing.
7. Learn professional ethics and ethical responsibilities of computing professionals.
8. Understand privacy, security, anonymity, and ethical issues in the digital world.
9. Gain knowledge of intellectual property rights, cybercrime, and computer ethics.
10. Examine the ethical and social impact of emerging technologies such as AI, IoT, and social media.

Course Outcomes (COs)

After successful completion of this course, the students will be able to:

CO1: Explain the concepts of morality, ethics, law, ethical theories, and their significance in computing and information technology.

CO2: Apply professional codes of conduct and ethical decision-making principles to resolve ethical issues encountered in computing professions.

CO3: Analyze ethical, legal, and social issues related to anonymity, privacy, civil liberties, and data protection in digital environments.

CO4: Analyze intellectual property rights, computer crimes, digital evidence, and ethical responsibilities in computer forensic investigations.

CO5: Evaluate the ethical and societal implications of emerging technologies such as Artificial Intelligence, IoT, online social networks, biometrics, surveillance, and digital media.

UNIT-I:

Morality and Ethics: Moral code of behaviour, Purpose of Law, Penal code, Morality and the law; Ethical theories, Functional definition of ethics, Ethical reasoning and decision making, codes of ethics, technology and values.

UNIT-II:

Ethics and Professions: Professional code of conduct, Professional decision making and ethics, Whistleblowing, harassment and discrimination, ethical and moral implications of profession.

UNIT-III:

Anonymity, Privacy and Civil Liberties: Anonymity in the Internet, its advantages and disadvantages, Types and value of Privacy, Privacy protection and civil liberties, implications for privacy of the Aadhaar database [4],Workplace privacy and surveillance, Ethics for privacy, Ethical and legal basis for privacy. Case studies of Cambridge Analytica [1] [2], SECC in India [3].

UNIT-IV:

Intellectual Property Rights, Computer Crime and Ethics: Computer Products and services, foundations of Intellectual Property Rights (IPR), Ownership, IPR crimes, Protecting computer software under IPR, Transnational issues. Digital evidence, preserving digital evidence, analysis of digital evidence, ethical implications and responsibilities in computer forensic investigations.

UNIT-V:

Social Implications of Information Technology: Advances in Artificial Intelligence, AI and ethics, bias in AI algorithms, Online social networks, Ethical issues in online social networks, security and crimes in online social networks; Internet of Things (IoT) and Location tracking, ethics of tracking, ethics of personalised search/ advertisements in online social networks and search engines, self-driving vehicles and ethical/legal issues, surveillance and ethical issues, use of biometrics for authentication, ethical implications of biometric technologies, fake news and implications of its spread via social media platforms, digital divide and its implications, sexual predators and pornography on the Internet and its implications.

Suggested Readings:

2. Ethical and Social Issues in the Information Age, Joseph Migga Kizza, 6th ed. 2017 Edition, ISBN-10: 1447149890, ISBN-13: 978-1447149897

Reference Books:

[1] https://theprint.in/politics/exclusive-inside-story-cambridge-analytica-actually india/44012/

[2] https://thewire.in/tech/what-we-know-and-what-we-dont-about-what-cambridge analytica-did-in-india

[3] https://www.huffingtonpost.in/entry/aadhaar-national-social-registry-database-modi in 5e6f4d3cc5b6dda30fcd3462

[4] https://www.huffingtonpost.in/entry/one-year-after-aadhaar-judgement-we-are-still-not listening in 5d8bb628e4b0ac3cdda24659

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6596313/

[6] https://yjolt.org/sites/default/files/21_yale_j.l._tech._106_0.pdf

[7] https://hbr.org/2019/10/what-do-we-do-about-the-biases-in-ai

[8] https://www.nytimes.com/2019/11/19/technology/artificial-intelligence-bias.html

[9] https://aibusiness.com/three-notable-examples-of-ai-bias/

[10] https://thewire.in/tech/artificial-intelligence-has-a-gender-bias-problem-just-ask-siri

[11] https://www.theguardian.com/books/2019/feb/02/age-of-surveillance-capitalism shoshana-zuboff-review

[12] https://www.nature.com/articles/d41586-018-07135-0

[13] https://www.newyorker.com/science/elements/a-study-on-driverless-car-ethics-offers-a troubling-look-into-our-values

[14] https://towardsdatascience.com/how-ethical-is-facial-recognition-technology 8104db2cb81b

[15] https://www.technologyreview.com/f/615068/facial-recognition-european-union temporary-ban-privacy-ethics-regulation/

[16]https://www.meity.gov.in/writereaddata/files/Information%20Technology%20(Intermediary%20Guielines%20and%20Digital%20Media%20Ethics%20Code)%20Rules%2C%202021%20(updated%2006.04. 2023)-.pdf

[17] https://www.meity.gov.in/content/cyber-laws

[18]https://www.tec.gov.in/pdf/Studypaper/AI%20Policies%20in%20India%20A%20status%20Paper%20 final.pdf

PC651 AI	**DEEP LEARNING LAB**					
Prerequisites			L	T	P	C
			2	0	0	1
Evaluation	CIE	25 Marks	SEE		50 Marks	

Course Objectives:

1. The objectives of this course are
2. Understand basic concepts of Deep learning and their applications.
3. Evaluating Deep learning methods, models and algorithms.
4. Analyzing CNN, RNN, Transformers and GAN along with their applications.
5. To design sequence learning and natural language processing models using LSTM, Encoder–Decoder, and BERT architectures.

Course Outcomes:

On Successful completion of the course, students will be able to

CO1 Evaluate the performance of various optimization techniques used in deep learning.

CO2 Analyze various Autoencoders and Regularization Techniques.

CO3 Design and Develop various Convolution Neural Networks architectures.

CO4 Analyze various RNNs and Encoder Decoder Models.

CO5 Understand the importance of Transformers and GANS to develop real-time applications.

List of Experiments:

1. Implementation of Perceptron for Boolean operations.
2. Implementation of Classification and regression problems with Multilayer Perceptron.
3. Compare the Performance of various Optimization techniques of Momentum Based GD, Stochastic GD, and Adam.
4. Implementation of Denoising auto encoders.
5. Implementation of Regularization through L2, Dropout, and Early stopping Techniques and compare their Performance.
6. Implementation of VGG16, ResNet50 and compare their performance.
7. Implementation of LSTM for next word prediction.
8. Implementation of Encoder Decoder Models for Language translation.
9. Implementation of pretrained BERT Models.
10. Implementation of GANs for generating fake images.

PC651CS	**QUANTUM COMPUTING LAB**					
Prerequisites			L	T	P	C
			2	0	0	1
Evaluation	CIE	25 Marks	SEE		50 Marks	

Course Objectives:

1. To familiarize students with the principles of quantum computation and quantum information processing.
2. To provide hands-on experience in designing and simulating quantum circuits using IBM Quantum Composer.
3. To develop skills in implementing quantum algorithms using the Qiskit programming framework.
4. To enable students to analyze quantum gate operations, measurement processes, and quantum circuit behavior.
5. To expose students to practical applications of quantum computing in cryptography, search, optimization, and communication.

Course Outcomes:

CO1 Demonstrate the operation of single-qubit and multi-qubit quantum gates using quantum circuit simulators.

CO2 Design and simulate quantum circuits and interpret measurement results using IBM Quantum Composer.

CO3 Develop quantum programs using Python and Qiskit for implementing fundamentalquantum operations.

CO4 Implement and analyze quantum algorithms such as Deutsch's, Deutsch–Jozsa, Grover's, and Shor's algorithms.

CO5 Evaluate the performance and applications of quantum computing techniques in solving computational problems.

LIST OF EXPERIMENTS:

1. Single Qubit Gate Simulation - Quantum Composer

2. Multiple Qubit Gate Simulation - Quantum Composer

3. Composing Simple Quantum Circuits with Q-Gates and Measuring Output into Classical Bits

4. IBM Qiskit Platform Introduction

5. Implementation of Shor's Algorithm

6. Implementation of Grover's Algorithm

7. Implementation of Deutsch's Algorithm

8. Implementation of Deutsch-Jozsa Algorithm

Note: Using Python and Qiskit

PW652 AI	**MINI PROJECT**					
Prerequisites	Software Engineering Principles A Core Programming Language		L	T	P	C
			0	0	2	1
Evaluation	CIE	25 Marks	SEE 50		50 Marks	

Course Objectives:

1 To help students identify and understand real-world problems in their field.

2 To encourage application of technical knowledge and skills learned in previous courses.

3 To improve students' ability to work in teams and manage projects effectively.

4 To enhance communication skills through technical writing and presentations

5 To promote awareness of current technologies, industry practices, and research trends.

Course Outcomes:

On completion of this course, the student will be able to

CO1 Identify a relevant problem and define a clear project objective..
CO2 Apply technical and analytical skills to design and develop a solution.
CO3 Work effectively as a team to plan, implement, and manage the project..
CO4 Communicate ideas clearly through oral presentations and written reports.
CO5 Demonstrate awareness of industry standards and modern tools used in their domain

The department may initiate the project allotment procedure at the end of Semester V and finalize it within the first two weeks of Semester VI.

The department shall appoint a Project Coordinator to oversee the following activities:

- Collection of project topics and descriptions from faculty members. Problems may also be solicited from industries.
- Formation of student groups, with a maximum of three (3) students per group.
- Allotment of project guides to the student groups.

The objective of the Mini Project is to develop solutions to realistic problems by applying the knowledge and skills acquired through various courses, emerging technologies, and current industry practices. This enables students to understand contemporary problems in their respective domains and explore suitable methodologies for solving them.

To create awareness of current problems and solution techniques, the first two (2) weeks of Semester VI shall be devoted to special lectures delivered by faculty members, research scholars, postgraduate students of the department, and invited experts from industries and R&D organizations.

Upon completion of these seminars, each group shall prepare and submit a formal project proposal based on their own ideas or as suggested by the project guide.

A seminar schedule shall be prepared by the Project Coordinator for all students from the fifth week until the last week of the semester, and the schedule shall be strictly followed.

Each project group shall be required to:

1. Submit a one-page synopsis prior to the seminar for display on the department notice board.
2. Deliver a 30-minute presentation followed by a 10-minute discussion session.
3. Submit a technical report on the seminar presentation.

At least two faculty members shall evaluate the Mini Project for the award of sessional marks. The evaluation shall be based on the performance of students in all three components mentioned above.

The seminar presentation shall include the following aspects of the project:

- Problem definition and specification.
- Literature survey.
- Overview of available techniques for solving the identified problem.
- Planning of the work and preparation of activity (bar) charts.
- Oral presentation and technical report writing.

SI671AI	**SUMMER INTERNSHIP**					
Prerequisites			L	T	P	C
			-	-	-	
Evaluation	SEE	-	CIE		-	

Course Objectives:

1 To expose students to real-time industrial environments and practices.

2 To help students understand how classroom knowledge is applied in the industry

3 To encourage professional interactions and collaboration with industry experts

4 To develop problem-solving and project execution skills in real-world scenarios.

5 To improve technical reporting and presentation abilities.

Course Outcomes:

On completion of this course, the student will be able to

CO1 Gain hands-on experience by working on real industry problems
CO2 Apply academic knowledge to practical and professional situations.
CO3 Interact and work effectively with professionals in a team environment.
CO4 Prepare a clear and concise technical report based on the work done.
CO5 Deliver a structured seminar presentation summarizing the internship experience

Summer Internship is introduced as a part of the curriculum to encourage students to work on problems of interest to industries. A batch of three (3) students shall be attached to a mentor from a Computer Industry, Software Company, or R&D Organization for a period of **8 weeks**. The internship shall be carried out during the summer vacation after the completion of the **III Year** of the course.

A faculty coordinator shall be assigned to each group of three students to monitor their progress and interact with the industry coordinator. Upon completion of the internship, each student shall submit a brief technical report on the work carried out and deliver a seminar presentation organized by the department.

The award of sessional marks shall be based on the performance of the students, as evaluated by a committee constituted by the department. One faculty member shall coordinate the overall activities related to the Industry Attachment Program.

Students are required to undergo a **Summer Internship of Six (6) Weeks duration** at the end of **Semester VI**, and the corresponding credits shall be awarded after evaluation in **Semester VII**.