BACHELOR’S DEGREE PROGRAMME

B.Tech.

Computer Science and Engineering with Specialization in Data Science

Academic Curricula

2024-2028

SCHOOLOFCOMPUTERENGINEERING

KALINGAINSTITUTEOF INDUSTRIAL TECHNOLOGY
BHUBANESWAR – 751024

ODISHA,INDIA

Programme Specific Outcome (PSO)

PSO I: Apply computing theory, languages and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analysis.

PSO II: Apply the principles and techniques of database design, administration, and implementation to enhance data collection capabilities and decision-support systems. Ability to critique the role of information and analytics in supporting business processes and functions.

PSO III: Invent and use appropriate models of data analysis, assess the quality of input, derive insight from results, and investigate potential issues. Also to organize big data sets into meaningful structures, incorporating data profiling and quality standards.

Guideline and Notes to obtain the Specialization

A student has to follow the B.Tech Computer Science curricula. To get the specialization the student has to take the following as the professional electives in the respective semester from the basket.

PE: Professional Elective				
PE	**CourseCode**	**Course Title**	**Pre-requisites**	**Credits**
PE I		Any one Subject from PE- I Basket of CSE Syllabus.		3
PE II	CS30035	Bigdata Using SCALA	-	3
PE III	CS30030	Reinforcement Learning		3
PE IV	CS40001	Deep Learning Techniques		3
PE V	CS40020	Data Modeling and Visualization		3

Course Title	**Big Data using SCALA**
Course Code (Credit)	**CS30035 (L-T-P-Cr: 3-0-0-3)**
Prerequisites	**Background in Python programming, basic Linux commands, statistics and machine learning**

Course Objectives:

- To understand the fundamental concepts of the Big Data ecosystem and its technologies.
- To develop a solid foundation in Scala programming to solve Big Data problems.
- To implement data processing pipelines using Apache Spark and Scala.
- To apply advanced data analysis techniques on Big Data using Scala.
- To design and implement strategies for handling and managing large-scale data in a distributed environment.
- To develop and deploy real-world Big Data applications using Scala.

Course Contents:

UNIT I:

Introduction

Big Data systems, Spark overview, looking under the hood: Scala, Scala and Functional Programming, Scala; Important concepts. Parallel Processing and Mutable State, More Advanced Scala Concepts (REPL, substitution, type inference, lazy functions, lists and streams, generics and variance); Dealing with Exceptional Conditions.

UNIT II:

Collections

Lazy Lists, Managing State, Types, Functional composition and for comprehensions, recursion, Week, Syntax, Type Declarations, Functions, methods & operators, Specifications & Unit Tests.

UNIT III:

Implicits

Serialization/de-serialization, Parallel Processing and Futures, Monoids, functors, and monads, Syntactic sugar, Repositories, Enumerated Types; Actors, Pattern matching, Tour of the API, Parsing and DSLs.

UNIT IV:

Spark

Spark details, GraphX, Mllib, Spark Streaming and Spark SQL, Play/Activator, Numerical Computing, Projects and other topics not already covered.

CourseOutcomes:

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

CO1: Describe the components of the Big Data ecosystem, the role of Scala within this ecosystem, and identify key technologies and tools used for Big Data processing.

CO2: Write efficient and robust Scala code, leveraging both functional and object-oriented programming paradigms to manage and process large datasets.

CO3: Create, optimize, and deploy Spark applications using Scala to perform data processing tasks such as data transformation, aggregation, and analysis on large-scale datasets.

CO4: Utilize Scala libraries and Spark MLlib to perform machine learning tasks, statistical analysis, and data mining on large datasets, generating actionable insights.

CO5: Apply best practices for data storage, retrieval, and management in distributed systems, ensuring efficient and reliable Big Data workflows.

CO6: Design, develop, and implement end-to-end Big Data applications, utilizing Scala and related technologies to solve complex real-world problems, and evaluate their performance through case studies and practical projects.

Textbooks:

1. Martin Odersky, Lex Spoon, Bill Venners, and Frank Sommers, Programming in Scala, Artima Inc, 2021
2. Paul Chiusano and Runar Bjarnason, Functional Programming in Scala, Manning Publications, 2014

ReferenceBooks:

1. Md. Rezaul Karim, Scala and Spark for Big Data Analytics, Packt Publishing, 2017
2. Irfan Elahi, Scala Programming for Big Data Analytics, APress, 2019

Course Title	**Reinforcement Learning**
Course Code (Credit)	**CS30030 (L-T-P-Cr: 3-0-0-3)**
Prerequisites:	**Probability and Linear Algebra (Basics), Programming Knowledge (preferably Python), Data Structures and Algorithms**

Course Objectives

- To understand the principles of reinforcement learning and its capabilities.
- To understand the basic concepts of reinforcement learning, including agents, environments, states, actions, rewards, and policies.
- Learn about various RL algorithms such as Q-learning, SARSA (State-Action-Reward-State-Action), Deep Q-Networks (DQN), Policy Gradients, Actor-Critic methods, and their respective advantages and disadvantages.
- Develop the skills to implement RL algorithms from scratch using programming languages like Python, and libraries such as TensorFlow, PyTorch, and OpenAI Gym.
- Evaluate the performance of RL models, understand concepts of exploration vs. exploitation, and improve models through techniques like hyperparameter tuning, reward shaping, and handling environment variability.

Course Contents:

UNIT I

Introduction: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope, An Extended Example: Tic-Tac-Toe,

Multi-armed Bandits: K-armed Bandit Problem, Action-value Methods, the 10-armed Testbed, Incremental Implementation, tracking a Nonstationary Problem, Optimistic Initial Values, Upper-Confidence-Bound Action Selection, Gradient Bandit Algorithms

Finite Markov Decision Processes:
The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Unified Notation for Episodic and Continuing Tasks, Policies and Value Functions, Optimal Policies and Optimal Value Functions, Optimality and Approximation

UNIT II

Dynamic Programming: Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration, Efficiency of Dynamic

Monte Carlo Methods: Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation.

Temporal-Difference Learning: TD Prediction, Advantages of TD Prediction Methods, Optimality of TD, Sarsa: On-policy TD Control, Q-learning: Off-policy TD Control, Expected Sarsa, Maximization Bias and Double Learning, Games, After states, and Other Special Cases.

UNIT III

n-step Bootstrapping: n-step TD Prediction, n-step Sarsa, n-step Off-policy Learning, Off-policy Learning Without Importance Sampling, The n-step Tree Backup Algorithm

Planning and Learning with Tabular Methods: Models and Planning, Dyna: Integrated Planning, Acting, and Learning, When the Model Is Wrong, Prioritized Sweeping, Prioritized Sweeping, expected vs. Sample Updates, Trajectory Sampling, Real-time Dynamic Programming, Planning at Decision Time, Heuristic Search, Rollout Algorithms, Monte Carlo Tree Search

UNIT IV

On-policy Prediction with Approximation: Value-function Approximation, The Prediction Objective, Stochastic-gradient and Semi-Gradient Methods, Linear Methods, Feature Construction for Linear Methods, Polynomials, Fourier Basis, Coarse Coding, Tile Coding, Radial Basis Functions, Selecting Step-Size Parameters Manually, Nonlinear Function Approximation: Artificial Neural Networks, Least-Squares TD

On-policy Control with Approximation: Episodic Semi-Gradient Control, Semi-gradient n-step Sarsa, Average Reward: A New Problem Setting for Continuing Tasks, Deprecating the Discounted Setting, Differential Semi-gradient n-step Sarsa

UNIT V

Policy Gradient Methods: Policy Approximation and its Advantages, The Policy Gradient Theorem, REINFORCE: Monte Carlo Policy Gradient, REINFORCE with Baseline, Actor–Critic Methods, Policy Gradient for Continuing Problems, Policy Parameterization for Continuous Actions

Course Outcomes:

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

CO1: Demonstrate a thorough understanding of the theoretical underpinnings of reinforcement learning.
CO2: Implement key reinforcement learning algorithms, such as Q-learning, Deep Q-Networks (DQN), Policy Gradients, and Actor-Critic methods.
CO3: Design, develop, and train RL models using popular frameworks and tools (e.g., Tensor Flow, PyTorch, Open AI Gym).
CO4: Apply reinforcement learning to solve complex, real-world problems in various domains, such as robotics, game playing, autonomous vehicles, and recommendation systems.
CO5: Evaluate the performance of RL models using appropriate metrics and methodologies, and enhance their effectiveness through techniques like reward shaping, hyper parameter tuning, and handling stochastic environments.
CO6: Recognize and address ethical, practical, and safety considerations in the deployment of RL systems, including the potential for unintended consequences and the importance of robust and interpretable models.

Textbooks:

Richard S. Sutton and Andrew G. Barto; Reinforcement Learning: An Introduction; 2nd Edition, MIT Press, 2020, Available online:

Reference Books:
1. Andrew G. Barto and Sridhar Mahadevan; Recent Advances in Hierarchical Reinforcement Learning; Discrete Event Dynamic Systems, vol. 13, pp. 341–379, 2003.

2. Vincent Francois-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare and Joelle Pineau; An Introduction to Deep Reinforcement Learning; ArXiv ePrint, 2018.

3. Kaiqing Zhang, Zhuoran Yang, Tamer Başar; Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms; ArXiv ePrint, 2021.

Course Title	**Deep Learning and Techniques**
Course Code (Credit)	**CS40001 (L-T-P-Cr: 3-0-0-3)**
Prerequisites	**Probability, Linear Algebra**

Course Objectives

- To understand the principles of deep learning and its capabilities.
- To understand different types of neural networks.
- Tomaster the concepts of forward propagation and backward propagation in deep neural networks (DNN)
- Toget introduced to modelling and performance improvement techniques in deep learning
- To comprehend hyper parameter tuning and model interpretability
- To learn about dropout and early stopping techniques and their implementation.
- To grasp the fundamentals of convolution neural networks (CNNs) and recurrent neural networks (RNNs).
- To acquire practical skills to design, implement, and train deep learning systems.

Course Contents:

UNIT I

Basics of artificial neural networks (ANN): Artificial neurons, Computational models of neurons, Structure of neural networks, Functional units of ANN for pattern recognition tasks.

Feed forward neural networks: Pattern classification using perception, Multilayer feed forward neural networks (MLFFNNs), Back propagation learning, Empirical risk minimization, Regularization, Auto encoders

UNIT II

Deep neural networks (DNNs): Difficulty of training DNNs, Greedy layer wise training, Optimization for training DNNs, Newer optimization methods for neural networks (AdaGrad, RMSProp, Adam), Second order methods for training, Regularization methods (dropout, drop connect, batch normalization)

UNIT III

Convolution neural networks (CNNs): Introduction to CNNs – convolution, pooling, Deep CNNs, Different deep CNN architectures – LeNet, AlexNet, VGG, PlacesNet, training a CNNs: weights initialization, batch normalization, hyper parameter optimization, Understanding and visualizing CNNs.

UNIT IV

Recurrent neural networks (RNNs): Sequence modeling using RNNs, Back propagation through time, Long Short Term Memory (LSTM), Bidirectional LSTMs, Bidirectional RNNs, Gated RNN Architecture

UNIT V

Generative models: Restrictive Boltzmann Machines (RBMs), Stacking RBMs, Belief nets, Learning sigmoid belief nets, Deep belief nets.

Course Outcomes:

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

CO1: Acquire fundamental understanding of the core concepts in neural networks and deep learning

CO2: Implement most popular deep learning programming frameworks based on Python

CO3: Explore the mathematics behind the functioning of artificial neural networks

CO4: Analyze the given dataset for designing a neural network based solution

CO5: Design and implement deep learning models for signal/image processing applications.

CO6: Develop new architectures to solve practical real-world problems such as computer vision and natural language processing

Textbooks:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep learning, In preparation for MIT Press, Available online: http://www.deeplearningbook.org

ReferenceBooks:

1. S. Haykin, Neural Networks and Learning Machines, Prentice Hall of India, 2010

2. Satish Kumar, Neural Networks - A Class Room Approach, Second Edition, Tata McGraw-Hill, 2013

3. B. Yegnanarayana, Artificial Neural Networks, Prentice- Hall of India, 1999

4. C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Course Title	**Data Modelling and Visualization**
Course Code (Credit)	**CS40020 (L-T-P-Cr: 3-0-0-3)**
Prerequisites	**Basic understanding of statistics and bivariate linear regression. Some experience in the use of a statistical software package, preferable basic data management tasks in R**

Course Objectives

- Gain a comprehensive understanding of the fundamental principles and theories of data visualization, including visual perception, data types, and the importance of clarity and accuracy in visual representations.
- Learn to use a variety of data visualization tools and software, such as Tableau, Power BI, D3.js, Matplotlib, and Plotly, to create compelling and interactive visualizations.

- Develop the ability to design effective and aesthetically pleasing visualizations by choosing appropriate chart types, color schemes, and layouts based on the nature of the data and the intended audience.
- Understand how to apply visualization techniques to explore and analyze data, uncover patterns, trends, and insights, and effectively communicate findings to stakeholders.
- Acquire skills in data preprocessing, including cleaning, transforming, and aggregating data, to ensure it is in a suitable format for visualization.
- Learn to critically evaluate and critique visualizations, both their own and others', to identify strengths and weaknesses and make informed improvements.

Course Contents:

UNIT I

Introduction: Purpose, Scope, Intended readers, Content preview, Communication style. Different Roles of Tables and Graphs Quantities and categories, Choosing the best medium of communication, Tables defined, when to use tables, Graphs defined, A brief history of graphs, When to use graphs

UNIT II

Visual Perception and Graphical Communication: Mechanics of sight, Attributes of pre-attentive processing, Applying visual attributes to design, Gestalt principles of visual perception

Fundamental Variations of Graphs: Encoding data in graphs, Relationships in graphs, Graph design solutions

General Design for Communication: Highlight, Organize, Integrate tables, graphs, and text

UNIT III

General Graph Design: Maintain visual correspondence to quantity, Avoid 3D Component Level Graph Design Primary data component design, Secondary data component design, Non-data component design.

UNIT IV

Displaying Many Variables at Once: Combining multiple units of measure, Combining graphs in a series of small multiples, Other arrangements of multi-graph series, Correlation Analysis

Course Outcomes:

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

CO1: Demonstrate proficiency in using various data visualization tools and software such as Tableau, Power BI, D3.js, Matplotlib, and Plotly to create effective and interactive visualizations.

CO2: Apply principles of visual storytelling to communicate data-driven insights clearly and persuasively to diverse audiences, tailoring visualizations to the needs of specific stakeholders.

CO3: Design aesthetically pleasing and functional visualizations by selecting appropriate chart types, color schemes, and layouts, and critically evaluate visualizations for clarity, accuracy, and effectiveness.

CO4: Perform data preprocessing tasks, including cleaning, transforming, and aggregating data, to prepare datasets for visualization and ensure they accurately represent the underlying information.

CO5: Conduct exploratory data analysis using visualization techniques to identify patterns, trends, and anomalies in data, thereby deriving meaningful insights for decision-making.

CO6: Demonstrate an understanding of key principles and theories of data visualization, including visual perception, cognitive load, and data ethics, and apply these principles to create impactful visualizations.

Textbooks:

Stephen Few, Show Me the Numbers: Designing Tables and Graphs to Enlighten. (Second Edition), Analytics Press, 2012.

Reference Books:

1. Stephen Few, Now You See It: Simple Visualization Techniques for Quantitative Analysis, Analytics Press; First Edition, 2009
2. Tamara Munzner, Visualization: Analysis and Design, CRC Press, 2014
3. Edward Tufte, The Visual Display of Quantitative Information, Graphics Press, 2001