Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati

Department of Artificial Intelligence and Data Science

S.Y. B. Tech Syllabus 2024-25 (As per NEP 2020)

Syllabus: HONOR w. e. f. AY: 2024- 2025
SEMESTER-III

Honor in Artificial Intelligence and Data Science

SEM	Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks								Credits		
			T H	PR	TUT	Activity	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
III	AI23281	Introduction to Computational Intelligence	2	2	-	10	20	50	20	20		120	2	1		3

Dept. Autonomy Coordinator
Mrs. R. S. Naik

Dept. Academic Coordinator
Mr. P. N. Shendage

HOD, AI&DS
Dr. C. S. Kulkarni

Dean Autonomy
Dr. C. B. Nayak

Dean Academics
Dr. S. M. Bhosle

Principal
Dr. S. B. Lande

VPKBIET BARAMATI
Affiliated to S.P.P.U. Pune
Id No./PU/PU/Engg/153/2005
PUNE (MS.)

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

BUCKET OF Honor SUBJECT

Honor SUBJECT (only for students having CGPA >= 7.5)
AI23281: Introduction to Computational Intelligence

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(Autonomous Institute)

AI23281- Introduction to Computational Intelligence

Teaching Scheme: **Theory: 2 Hours/Week** **Practical: 2 Hour/Week**	**Credits** **03**	**Examination Scheme:** **Activity: 10 Marks** **ISE: 20 Marks** **ESE: 50 Marks** **Term Work: 20 Marks** **Practical: 20 Marks**

Prerequisites: Python Programming

Course Objectives:

- To provide a strong foundation on fundamental concepts in Computational Intelligence.
- To enable Problem-solving through various searching techniques.
- To apply these techniques in applications which involve perception, reasoning and learning.
- To apply Computational Intelligence techniques for information retrieval.
- To apply Computational Intelligence techniques primarily for machine learning.

Course Outcomes (COs): The students will be able to learn:

CO1: Provide a basic exposition to the goals and methods of Computational Intelligence.
CO2: Study of the design of intelligent computational techniques.
CO3: Apply the Intelligent techniques for problem solving.
CO4: Improve problem solving skills using the acquired knowledge in the areas of, reasoning, natural language understanding, computer vision, automatic programming and machine learning.
CO5: Use supervised/ unsupervised/ reinforcement learning method.

Course Contents

Unit I Introduction to Computational Intelligence (06 Hours)

History, Biological neurons & Artificial models, intelligence machine, man-machine interaction, data mining for IoT, Relation between AI, ML, DL, data science and CI. Types of data analytics – predictive, prescriptive, descriptive, and diagnostic, Data analytics rule. Introduction of R language.

Unit II Fuzzy Logic (6 Hours)

Introduction to Fuzzy Set: Introduction, definition, membership Function, Fuzzy operator, Fuzzy Set Characteristics, Fuzziness and Probability.
Fuzzy Logic and Reasoning--Fuzzy Logic: Linguistics Variables and Hedges, Fuzzy Rules.
Fuzzy Inferencing: neuro inferencing Fuzzification, Defuzzification
Fuzzy logic Controllers: Fuzzy logic Controllers, Fuzzy logic Controller Types

Unit III Classification (6 Hours)

Forms of Learning , Supervised Learning , Learning Decision Trees , Regression and Classification with Linear Models , Artificial Neural Networks , Nonparametric Models , Support Vector Machines, Statistical Learning , Learning with Complete Data , Learning with Hidden Variables- The EM Algorithm , Reinforcement Learning.

Unit IV Probabilistic Models (6 Hours)

Probability basics, Bayes Rule and its Applications, Bayesian Networks, Exact and Approximate Inference in Bayesian Networks.

Unit V Artificial Intelligence(6 Hours)

Introduction to Artificial Intelligence, Search, Heuristic Search, A* algorithm, Game Playing, Alpha-Beta Pruning, Expert systems, Inference, Rules, Forward Chaining and Backward Chaining, Genetic Algorithms.

Unit VI Knowledge Representation & Reasoning (6 Hours)

Proposition Logic — First Order Predicate Logic , Unification , Forward Chaining, Backward Chaining , Resolution , Knowledge Representation , Ontological Engineering , Categories and Objects , Events , Mental Events and Mental Objects — Reasoning Systems for Categories, Reasoning with Default Information, Prolog Programming.

Text Books:

1. Andreis P. Engelbrecht, "Computational Intelligence an introduction", 2nd edition, Wiley publication
2. Nazmul Siddique, Hojjat Adeli, "Computational Intelligence, Synergies of Fuzzy logic, Neural Networks and Evolutionary computing", Wiley publication
3. Stuart Russell, Peter Norvig, —Artificial Intelligence: A Modern Approac, Third Edition, Pearson Education / Prentice Hall of India, 2010.
4. Elaine Rich and Kevin Knight, —Artificial Intelligence, Third Edition, Tata McGrawHill, 2010.

Reference Books:

1. James M. Keller, Derong Liu, David B. Fogel, "Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation", John Wiley & Sons, 2016.
2. Mitchell Melanie, "An Introduction to Genetic Algorithms", The MIT Press Cambridge, Massachusetts, MIT Press paperback edition, 1998.
3. Dan W.Patterson, —Introduction to Artificial Intelligence and Expert Systems‖, PHI, 2006.
4. Nils J. Nilsson, —Artificial Intelligence: A new Synthesis‖, Harcourt Asia Pvt. Ltd., 2000.

E-Resources:

1. **https://nptel.ac.in/courses/106102220**
2. **https://onlinecourses.nptel.ac.in/noc23_cs87/preview**
3. **https://onlinecourses.nptel.ac.in/noc22_ee21/preview**

List of Assignments

1. Write a program to implement the K-means clustering algorithm in Python.
2. Implement a program to perform feature selection using a genetic algorithm.
3. Implement a fuzzy clustering algorithm (e.g. Fuzzy C-means) to cluster a dataset of points into different groups.
4. Implement a fuzzy logic controller for a temperature control system using Python. The controller should be able to adjust the temperature set point based on the input temperature and the desired comfort level.
5. Write a python program to simulate a coin flip: Write a Python program that simulates flipping a fair coin 100 times and counts the number of heads and tails.
6. Write a Python program that generates random numbers: 100 random numbers between 1 and 10 and calculates the probability of each number occurring.
7. Implement SVM classification algorithm on any dataset.
8. Implement A* algorithm using python.
9. Implement Bayesian classification algorithm on any dataset.
10. Apply linear regression model techniques to predict the data on any dataset.

Syllabus Honor w. e. f. AY: 2024- 2025
SEMESTER-IV

Honor in Artificial Intelligence and Data Science

SEM	Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks							Credits			
			TH	PR	TUT	Activity	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
IV	AI23291	Soft Computing: Intelligent Problem-Solving Techniques	2	2	-	10	20	50	20	20		120	2	1		3

Dept. Autonomy Coordinator
Mrs. R. S. Naik

Dept. Academic Coordinator
Mr. P. N. Shendage

HOD, AI&DS
Dr. C. S. Kulkarni

Dean Autonomy
Dr. C. B. Nayak

Dean Academics
Dr. S. M. Bhosle

Principal
Dr. S. B. Lande

VPKBIET BARAMATI PUNE (MS.)

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

BUCKET OF Honor SUBJECT

Honor SUBJECT (only for students having CGPA >= 7.5)
AI23291-: Soft Computing: Intelligent Problem-Solving Techniques

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati
(Autonomous Institute)

AI23291- Soft Computing: Intelligent Problem-Solving Techniques

Teaching Scheme:	Credits	Examination Scheme:
Theory: 2 Hours/Week	**03**	**Activity: 10 Marks**
Practical: 2 Hour/Week		**ISE: 20 Marks**
		ESE: 50 Marks
		Term Work: 20 Marks
		Practical: 20 Marks

Prerequisites: Python Programming

Course Objectives:

- To explain the concept of soft computing techniques with their applications in product design. Manufacturing and operations with case studies
- To expose students to the concept of fuzzy logic and their applications in mechanical system.
- To familiarize the deep learning model development using artificial neural network.
- To introduce the concept of genetic algorithm and various advanced algorithms.
- To apply Markov models for system and process modeling and optimization

Course Outcomes (COs): The students will be able to learn:

1. Apply soft fuzzy logic approach for solving different problems in absence of sufficient data and using expert judgments.
2. Develop deep learning model using artificial neural network.
3. Apply genetic algorithms other random search procedures useful while seeking global optimum in self learning situations.
4. Apply reinforcement and deep learning models for different data sets and optimize the system performance

Course Contents

Unit I Fuzzy Logic (7 Hours)

Fuzzy Systems Fuzzy set theory: Fuzzy sets, Operations, Membership Functions, Fuzzy relations and their composition, Measures, Rules, Propositions, Implications, and inferences, Defuzzification techniques, Logic controller design, Some applications of fuzzy logic.

Unit II Artificial Neural Network (7 Hours)

Artificial Neural Network (ANN) Neuron, Nerve structure and synapse, Biological and artificial neurons, Architectures – single layer and multilayer feed forward networks, recurrent networks. Back propagation algorithm, Working principle, Types of ANN, Activation functions – linear, Sigmoid, Tanh, supervised and unsupervised learning, Training techniques for ANNs, Applications, advantages, and limitations.

Unit III Genetic Algorithms(7)

Basic Genetics, Concepts, Working Principle, Creation of Offspring, Encoding, Fitness Function, Selection Functions, Genetic Operators-Reproduction, Crossover, Mutation; Genetic Modeling, Advantages, limitations and applications, Comparison between GA and traditional algorithms

Text Books:

1. Neural Networks: A Comprehensive Foundation by S. Haykin, Pearson.
2. Fuzzy Logic with Engineering Application by T. J. Ross, John Wiley and Sons.
3. Evolutionary Computation by D.B. Fogel, IEEE Press.

4. D. K. Pratihar, Soft Computing, Narosa Publishing House, 2008.
5. An Introduction to Genetic Algorithm Melanic Mitchell (MIT Press).

Reference books:

1. Evolutionary Algorithm for Solving Multi-objective, Optimization Problems (2nd Edition), Collelo, Lament, Veldhnizer (Springer).
2. Fuzzy Logic with Engineering Applications Timothy J. Ross (Wiley).
3. Neural Networks and Learning Machines Simon Haykin (PHI).
4. Sivanandam, Deepa, Principles of Soft Computing, Wiley.
5. Jang J.S.R, Sun C.T. and Mizutani E, "Neuro-Fuzzy and soft computing", Prentice Hall. 6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill

E-Resources:

1. **https://nptel.ac.in/courses/106102220**
2. **https://onlinecourses.nptel.ac.in/noc23_cs87/preview**
3. **https://onlinecourses.nptel.ac.in/noc22_ee21/preview**

List of Assignments

1. Design triangular, trapezoidal, Gaussian, and sigmoid membership functions for a given problem (e.g., temperature classification: Cold, Warm, Hot). Plot these membership functions and analyze their behavior. **Tools:** Python libraries like `skfuzzy`, MATLAB.
2. Implement a simple neural network to solve the non-linear XOR problem. **Task:** Use an MLP with at least one hidden layer to classify the XOR dateset. Train the network using back-propagation. Compare the results with and without activation functions (e.g., ReLU, Sigmoid). **Tools:** Python libraries - NumPy, TensorFlow/Keras
3. Build a neural network to classify handwritten digits. **Task:** Load the MNIST dataset. Design and train a feedforward neural network with multiple hidden layers. Experiment with different optimizers (e.g., SGD, Adam). Visualize training loss and accuracy over epochs. **Tools:** `Keras`, `TensorFlow`, `Matplotlib`
4. Optimize the selection of items to maximize value while respecting weight constraints. **Task:** Implement a genetic algorithm to solve the **0/1 knapsack problem**. Define chromosomes as binary strings where `1` indicates selecting an item. Apply selection, crossover, and mutation operations. Compare results with a greedy algorithm. **Tools:** Python libraries: NumPy, Matplotlib.
5. Implement **Q-Learning**, a model-free RL algorithm. **Task:** Create a 4x4 grid world with obstacles and a goal state. Implement Q-Learning to learn the optimal policy. Visualize the Q-table, optimal actions, and rewards after convergence. **Tools:** Python libraries: NumPy, Matplotlib.