INSTITUTE OF ENGINEERING AND TECHNOLOGY
DEEN DAYAL UPADHYAYA GORAKHPUR UNIVERSITY
GORAKHPUR

COURSE STRUCTURE
&
SYLLABUS

FOR

B.TECH.
in

Artificial Intelligence and Machine Learning

Based
on
AICTE MODEL CURRICULUM

[Effective from the Session: 2025-26]

Curriculum for Bachelor of Technology

Artificial Intelligence and Machine Learning

Course Structure & Semester-wise Credit Distribution

Credit Distribution (Course Type) in B. Tech. (Artificial Intelligence and Machine Learning) Program		
S. No.	**Course Type**	**Credit**
1	BSC	14
2	ESC	15
3	HSMC	10
4	PCC	74
5	PEC	15
6	Project work, Seminar and internship in industry or elsewhere	16
7	AEC offered by University	08
8	SEC offered by University	09
9	OEC	12
10	**Mandatory Non-Credit Courses** [ACITE-Mandatory Non-Credit (NC) Courses]	(02) **Non-Credit**
Total Credit		**173**

Abbreviation:

BSC: Basic Science Course
ESC: Engineering Science Course
HMSC: Humanities & Social Science including Management Courses
PCC: Professional Core Course
PEC: Professional Elective Course
OEC: Open Elective Course

Basic Science Courses (BSC)			
S. No.	**Course Code**	**Course Title**	**Credits**
1.	ECHE101	Engineering Chemistry	3+0
2.	ECHE151	Engineering Chemistry Lab	0+1
3.	EPHY101	Engineering Physics	3+0
4.	EPHY151	Engineering Physics Lab	0+1
5.	EMAT103	Mathematics for Data Science	3+0
6.	EMAT104	Mathematics for Machine Learning	3+0
		Total	**14**
Engineering Science Course (ESC)			
1.	ECE101	Basic Electronics Engineering	3+0
2.	ECE151	Basic Electronics Engineering Lab	0+1
3.	ECE102	Basic Electrical Engineering	3+0
4.	ECE152	Basic Electrical Engineering Lab	0+1
6.	CSE101	Programming for Problem Solving	3+0
7.	CSE151	Programming for Problem Solving Lab	0+1
8.	ME103	Fundamental of Mechanical Engineering and Mechatronics	3+0
		Total	**15**
Humanities and Social Sciences including Management Courses (HSMC)			
1.	HSM101	Professional Communication	3+0
2.	HSM301	Organization Behavior	3+0
3.	HUV102	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	3+0
4.	HSM151	Professional Communication Lab	0+1
		Total	**10**

Professional Core Course (PCC)			
S. No.	**Course Code**	**Course Title**	**Credit**
1.	AIML102	Conversational AI	3+0
2.	AIML201	Artificial Intelligence	3+0
3.	AIML251	Artificial Intelligence Lab	0+1
4.	AIML202	Data Structure	4+0
5.	AIML252	Data Structure Lab	0+1
6.	AIML203	Programming in Python	2+0
7.	AIML253	Python Programming Lab	0+1
8.	AIML204	Data Analytics using R Programming	3+0
9.	AIML254	R-Programming Lab	0+1
10.	AIML205	Mathematical Logic	3+0
11.	AIML206	Soft Computing	3+0
12.	AIML256	Soft Computing Lab	0+1
13.	AIML207	Operating Systems	4+0
14.	AIML257	Operating System Lab	0+1
15.	AIML208	AI Ethics and Society	2+0
16.	AIML209	Machine Learning	4+0
17.	AIML259	Machine Learning Lab	0+1
18.	AIML210	Theory of Automata and Formal Languages	4+0
19.	AIML301	Reinforcement Learning	3+0
20.	AIML302	Database Management System	3+0
21.	AIML352	Database Management System Lab	0+1
22.	AIML303	Design and Analysis of Algorithms	4+0
23.	AIML353	Design and Analysis of Algorithm Lab	0+1
24.	AIML304	Data Mining and Data Warehousing	3+0
25.	AIML305	Natural Language Processing	3+0
26.	AIML306	Distributed System	3+0
27.	AIML307	Neural Network & Deep Learning	3+0
28.	AIML357	Neural Network & Deep Learning Lab	0+1
29.	AIML308	Cloud Computing	3+0
30.	AIML358	Cloud Computing Lab	0+1
31.	AIML309	Image Processing	3+0
Total			**74**

Project Work, Seminar and Internship/Industrial Training in Industry or elsewhere			
S. No.	**Course Code**	**Course Title**	**Credit**
1.	AIML MP1	AIML Mini Project I	**0+1**
2.	AIML MP2	AIML Mini Project II	**0+2**
3.	AIML IND401	Industrial Training	**0+2**
4.	AIML S401	Seminar	**0+2**
5.	AIML P402	AIML Major Project I	**0+3**
6.	AIML P403	AIML Major Project II	**0+6**
Total			**16**

Professional Elective Course (PEC)			
S. No.	**Semester**	**PEC-Elective**	**Credit**
1	V	Elective-I	3+0
2	VI	Elective-II	3+0
3	VII	Elective-I	3+0
4	VII	Elective-I	3+0
5	VIII	Elective-II	3+0
Total			**15**
Students have to select unique course(s) either from Elective-I or from Elective-II in the semester. Repetition of courses is not allowed.			

INSTITUTE OF ENGINEERING AND TECHNOLOGY
Deen Dayal Upadhyaya Gorakhpur University, Gorakhpur

Professional Elective Courses (from SWAYAM)
Discipline: Artificial Intelligence and Machine Learning, **Duration:** 12 Weeks, **Credit:** 3
Elective-1 (Odd Semester)

S. No.	Course Code	Course Id	Course Name	SME Name	Institute	Link
1.	EAIML 101	noc23- cs92	Artificial Intelligence: Search Methods For Problem Solving	Prof. Deepak Khemani	IITM	https://onlinecourses.nptel.ac.in/noc23_cs92/preview
2.	EAIML 102	noc23- cs84	Statistical Learning For Reliability Analysis	Prof. Monalisa Sarma	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs84/preview
3.	EAIML 103	noc23- cs126	Deep Learning for Computer Vision	Prof. Vineeth N Balasubramanian	IITH	https://onlinecourses.nptel.ac.in/noc23_cs126/preview
4.	EAIML 104	noc23-cs92	Artificial Intelligence : Search Methods For Problem Solving	Prof. Deepak Khemani	IITM	https://onlinecourses. nptel.ac.in/noc23_cs9 2/preview
5.	EAIML 105	noc23-ge40	Fundamentals Of Artificial Intelligence	Prof. Shyamanta M. Hazarika	IITG	https://onlinecourses. nptel.ac.in/noc23_ge4 0/preview
6.	EAIML 106	noc23-cs86	Machine Learning For Earth System Sciences	Prof. Adway Mitra	IITKGP	https://onlinecourses. nptel.ac.in/noc23_cs8 6/preview
7.	EAIML 107	noc23-ee100	Applied Linear Algebra For Signal Processing, Data Analytics And Machine Learning	Prof. Aditya K. Jagannatham	IITK	https://onlinecurses. nptel.ac.in/noc23_ee1 00/preview
8.	EAIML 108	noc23-ee99	Applied Optimization For Wireless, Machine Learning, Big Data	Prof. Aditya K. Jagannatham	IITK	https://onlinecourses. nptel.ac.in/noc23_ee9 9/preview
9.	EAIML 109	noc23-ma93	Essential Mathematics For Machine Learning	Prof. S. K. Gupta & Prof. Sanjeev Kumar	IITR	https://onlinecourses. nptel.ac.in/noc23_ma 93/preview
10.	EAIML 110	noc23-ge44	Bandit Algorithm (Online Machine Learning)	Prof. Manjesh Hanawa	IITB	https://onlinecourses. nptel.ac.in/noc23_ge4 4/preview

Note: In case of changes in the above subjects (**Elective-1 (Even Semester) (from SWAYAM)),** Dean /Director/ HoD will issue separate list of subjects.

Professional Elective Courses (from Department of CSE)
Elective-1 (Odd Semester)

S. No.	Course Code	Course Name	Credit
1.	EAIML116	Computer Vision	3+0
2.	EAIML117	Cognitive and Affective Systems	3+0
3.	EAIML118	Computational Intelligence	3+0
4.	EAIML119	High Performance Computing	3+0
5.	EAIML120	Parallel and Distributed Computing	3+0
6.	EAIML121	Data-Driven Reliability Analysis with AI and ML	3+0
7.	EAIML122	Applied Machine Learning for Engineering and Science	3+0
8.	EAIML123	Pervasive Computing	3+0
9.	EAIML124	Generative Models	3+0
10.	EAIML125	Randomized Algorithms	3+0

Note:

- Students may take subjects from Elective-1 (from SWAYAM) and/or Elective-1(from Department of CSE)
- Execution of the courses offered by the department depends on the availability of faculty persons.

Elective-2 (Even Semester) (from SWAYAM)
Discipline: Artificial Intelligence and Machine Learning, **Duration:** 12 Weeks, **Credit:** 3

S. No.	Course Code	Course Id	Course Name	SME Name	Institute	Link
1.	EAIML 201	noc25-cs69	The Joy of Computing using Python	Prof. Sudarshan Iyengar	IIT Ropar	https://onlinecourses.nptel.ac.in/noc25_cs69/preview
2.	EAIML 202	noc25-cs61	Quantum Algorithms and Cryptography	Prof. Shweta Agrawal	IIT Madras	https://onlinecourses.nptel.ac.in/noc25_cs61/preview
3.	EAIML 203	noc25-cs63	Secure Computation: Part I	Prof. Ashish Choudhury	IIIT Bangalore	https://onlinecourses.nptel.ac.in/noc25_cs63/preview
4.	EAIML 204	noc25-cs43	Introduction To Industry 4.0 And Industrial Internet Of Things	Prof. Sudip Misra	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc25_cs43/preview
5.	EAIML 205	noc25-ma27	High Performance Scientific Computing	Prof. Shivasubramanian Gopalakrishnan, Prof. Om Jadhav, Prof. S Vamshi Krishna, Prof. Ashish Kuvelkar	IIT Bombay	https://onlinecourses.nptel.ac.in/noc25_ma27/preview
6.	EAIML 206	noc23-cs31	Foundations of Cryptography	Prof. Ashish Choudhury	IIIT Bangalore	https://onlinecourses.nptel.ac.in/noc25_cs31/preview
7.	EAIML 207	noc25-ee13	Computer Vision And Image Processing - Fundamentals And Applications	Prof. M. K. Bhuyan	IIT Guwahati	https://onlinecourses.nptel.ac.in/noc25_ee13/preview
8.	EAIML 208	noc25-cs09	Business Intelligence & Analytics	Prof. Saji K Mathew	IIT Madras	https://onlinecourses.nptel.ac.in/noc25_cs09/preview
9.	EAIML 209	noc25-cs06	Algorithms in Computational Biology and Sequence Analysis	By Prof. Chirag Jain	IISc Bangalore	https://onlinecourses.nptel.ac.in/noc25_cs06/preview
10.	EAIML 210	noc25-cs38	Human Computer Interaction (In English)	Prof. Rajiv Ratn Shah	IIIT Delhi	https://onlinecourses.nptel.ac.in/noc25_cs38/preview

Note: In case of changes in the above subjects (**Elective-2 (Even Semester) (from SWAYAM)),** Dean /Director/ HoD will issue separate list of subjects.

Professional Elective Courses (from Department of CSE)
Elective-2 (Even Semester)

S. No.	Course Code	Course Name	Credit
1.	EAIML214	Computational Science and High-Performance Computing	3+0
2.	EAIML215	Confidential Computing with AI and ML	3+0
3.	EAIML216	Service Oriented Architecture	3+0
4.	EAIML217	Introduction to Internet of Things	3+0
5.	EAIML218	Quantum Computing	3+0
6.	EAIML219	Distributed Algorithms	3+0
7.	EAIML220	Block chain Architecture Design	3+0
8.	EAIML221	Big Data Analytics using Python	3+0
9.	EAIML222	Machine Learning for Computational Biology	3+0
10.	EAIML223	Software Defined Network	3+0

Note:

- Students may take subjects from Elective-2 (from SWAYAM) and/or Elective-2(from Department of CSE)
- Execution of the courses offered by the department depends on the availability of faculty persons.

ACITE- Mandatory Non-Credit (NC) Courses

S. No.	Course Code	Course Title	Credits
1.	-----------------------	Induction Program (3-weeks)	----------------
2.	ENV201	Environment & Ecology	2+0

AICTE Mandatory 3-Week Student Induction Program

The Essence and Details of Induction program can also be understood from the 'Detailed Guide on Student Induction program', as available on AICTE Portal.
(Link: https://www.aicte-india.org/sites/default/files/Model_Curriculum/UG-1/ug-vol1.pdf).

Induction Program (mandatory)	Three-week duration
Induction Program for students to be offered right at the start of the first year.	Physical activity
	Creative Arts
	Universal Human Values-I
	Literary
	Proficiency Modules
	Lectures by Eminent People
	Visits to local Areas
	Familiarization to Dept./Branch & Innovations

AEC and SEC Offered by the University for Implementation of NEP2020 (University Mandatory Course)

University offers a pool of courses for AEC and SEC to implement NEP2020. Student has to select one course under SEC category in first, second and third semester (repetition of courses are not allowed). In the same context, student has to select one course under AEC category in first, second, third and fourth semester (repetition of courses are not allowed).

As per NEP2020, year wise credit requirements for the award of "Certificate in Artificial Intelligence and Machine Learning", "Diploma in Artificial Intelligence and Machine Learning", "Bachelor of Vocation (B. Voc.) in Artificial Intelligence and Machine Learning" and "B. Tech. in Artificial Intelligence and Machine Learning" are given below:

First Year: After earning 46 credits in first year (*23 credits in first semester and 23 credits in second semester*), student will be eligible for **"Certificate in Artificial Intelligence and Machine Learning"**.

Second Year: After earning 92 credits *(46 credits from first year and 46 credits in second year (24 credits in third semester and 22 credits in fourth semester)),* student will be eligible for the award of "**Diploma in Artificial Intelligence and Machine Learning"**.

Third Year: After earning 142 credits *(46 credits from first year and 46 credits in second year and 50 credits in third year (25 credits in fifth semester and 25 credits in sixth semester)),* student will be eligible for the award of "**Bachelor of Vocation (B. Voc.) in Artificial Intelligence and Machine Learning"**.

Fourth Year: After earning 173 credits *(46 credits from first year and 46 credits in second year ,50 credits in third year and 31 credit in fourth year (16 credits in seventh semester and 15 credits in eighth semester)),* student will be eligible for the degree of **"B. Tech. in Artificial Intelligence and Machine Learning"**.

Table1: Year wise credit requirement for award of Certificate, Diploma, B. Voc., and B. Tech. Degree in Department of Artificial Intelligence and Machine Learning under the light of NEP2020.

After Year	Credit Requirement	Eligibility of
1st	46	"Certificate in Artificial Intelligence and Machine Learning"
2nd	92	"Diploma in Artificial Intelligence and Machine Learning"
3rd	142	"Bachelor of Vocation (B. Voc.) in Artificial Intelligence and Machine Learning"
4th	173	"B. Tech. in Artificial Intelligence and Machine Learning"

Engineering Open Elective Courses (from Department)

S. No.	Department	Course Code	Course Title	Credits	Category
1.	CSE	CSEOE01	Web Technology	3+0	Engineering Open Elective Course-1
	AIML	AIMLOE01	Machine Learning for Engineering and science applications		
2.	CSE	CSEOE02	Web Application Development using Python	3+0	Engineering Open Elective Course-2
	AIML	AIMLOE02	Machine Learning for Soil and Crop Management		
3.	CSE	CSEOE03	Front End Technologies	3+0	Engineering Open Elective Course-3
	AIML	AIMLOE03	Artificial Intelligence: Knowledge Representation and Reasoning		
4.	CSE	CSEOE04	Back-End Technologies	3+0	Engineering Open Elective Course-4
	AIML	AIMLOE04	Deep Learning for Natural Language Processing		

Note:

- Students of the department may choose Engineering Open Elective Courses offered by the department of Computer Science and Engineering or by other departments also.
- Execution of the courses offered by the department depends on the availability of faculty persons.

Open Elective I (from SWAYAM)

S. No.	Course code	Course Id	Course Name	SME Name	Institute	Link
1.	AIMLOE05	noc25-ag01	Basics of Crop Breeding and Plant Biotechnology	Prof. Joydeep Banerjee	IIT Kharagpur	https://onlinecourses.nptel.ac.in/noc25_ag01/preview
2.	AIMLOE06	noc25-ag02	Cooling Technology: Why and How utilized in Food Processing and allied Industries	Prof. Tridib Kumar Goswami	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
3.	AIMLOE07	noc25-ag06	Machine Learning for Soil and Crop Management	Prof. Somsubhra Chakraborty	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
4.	AIMLOE08	noc25-ag08	Novel Technologies for Food Processing and Shelf Life Extension	Prof. Hari Niwas Mishra	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
5.	AIMLOE09	noc25-ag09	Soil and Water Conservation Engineering	Prof. Rajendra Singh	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
6.	AIMLOE010	noc25-ag10	Soil Science and Technology	Prof. Somsubhra Chakraborty	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
7.	AIMLOE11	noc25-ag11	Traction Engineering	Prof. Hifjur Raheman	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview
8.	AIMLOE12	noc25-mg17	Data Analysis and Decision Making - I	Prof. Raghu Nandan Sengupta	IIT Kanpur	https://onlinecourses.nptel.ac.in//preview
9.	AIMLOE13	noc25-mg06	AI in Marketing	Prof. Zillur Rahman	IIT Roorkee	https://onlinecourses.nptel.ac.in//preview
10.	AIMLOE14	noc25-ge24	Non-conventional energy Resources	Prof. Prathap Haridoss	IIT Madras	https://onlinecourses.nptel.ac.in//preview
11.	AIMLOE15	noc25-mg66	Business Forecasting	Prof. Pankaj Dutta	IIT Bombay	https://onlinecourses.nptel.ac.in//preview
12.	AIMLOE16	noc25-mg38	Leadership and Team Effectiveness	Prof. Santosh Rangnekar	IIT Roorkee	https://onlinecourses.nptel.ac.in//preview
13.	AIMLOE17	noc25-mg08	Artificial Intelligence (AI) for Investments	Prof. Abhinava Tripathi	IIT Kanpur	https://onlinecourses.nptel.ac.in//preview
14.	AIMLOE18	noc25-hs68	Psychology Of Stress, Health And Well-Being	Prof. Dilwar Hussain	IIT Guwahati	https://onlinecourses.nptel.ac.in//preview
15.	AIMLOE19	noc25-hs67	Psychology of Personality and Individual Differences: Theory and Applications	Prof. Dilwar Hussain	IIT Guwahati	https://onlinecourses.nptel.ac.in//preview
16.	AIMLOE20	noc25-hs72	Soft Skill Development	Prof. Priyadarshi Patnaik Prof. V. N. Giri Prof. D. Suar	IIT Kharagpur	https://onlinecourses.nptel.ac.in//preview

Open Elective II (from SWAYAM)

S. No.	Course code	Course Id	Course Name	SME Name	Institute	Link
1.	AIMLOE21	noc23-ch44	Polymers: Concepts, Properties, Uses And Sustainability	Prof. Abhijit P Deshpande	IITM	https://onlinecourses.nptel.ac.in/noc23_ch44/preview
2.	AIMLOE22	noc23-ec02	Economics	Economics Of Banking And Finance Markets	Prof. Sukumar Vellakkal	https://onlinecourses.nptel.ac.in/noc23_ec02/preview
3.	AIMLOE23	noc23-ec06	Principles Of Economics	Prof. Sabuj Kumar Mandal	IITM	https://onlinecourses.nptel.ac.in/noc23_ec06/preview
4.	AIMLOE24	noc23-ge41	Solar Energy Engineering And Technology	Prof. Pankaj Kalita	IITG	https://onlinecourses.nptel.ac.in/noc23_ge41/preview
5.	AIMLOE25	noc23-hs130	Ecology And Society	Prof. Ngamjahao Kipgen	IITG	https://onlinecourses.nptel.ac.in/noc23_hs130/preview
6.	AIMLOE26	noc23-hs131	Science, Technology And Society	Prof. Sambit Mallick	IITG	https://onlinecourses.nptel.ac.in/noc23_hs131/preview
7.	AIMLOE27	noc23-hs143	Educational Leadership	Prof. Atasi Mohanty	IITKGP	https://onlinecourses.nptel.ac.in/noc23_hs143/preview
8.	AIMLOE28	noc23-hs145	Soft Skills	Prof. Binod Mishra	IITR	https://onlinecourses.nptel.ac.in/noc23_hs145/preview
9.	AIMLOE29	noc23-hs151	Public Speaking	Prof. Binod Mishra	IITR	https://onlinecourses.nptel.ac.in/noc23_hs151/preview
10.	AIMLOE30	noc23-hs155	Environmental Science	Prof. Samik Chowdhury & Prof. Sudha Goel	IITKGP	https://onlinecourses.nptel.ac.in/noc23_hs155/preview
11.	AIMLOE31	noc23-hs90	Consumer Psychology	Prof. Naveen Kashyap	IITG	https://onlinecourses.nptel.ac.in/noc23_hs90/preview
12.	AIMLOE32	noc23-lw02	Right To Information And Good Governance	Prof. Sairam Bhat	National Law School of India University	https://onlinecourses.nptel.ac.in/noc23_lw02/preview
13.	AIMLOE33	noc23-lw03	Constitution Of India And Environmental Governance: Administrative And Adjudicatory Process	Prof. Sairam Bhat Prof. M. K. Ramesh	National Law School of India University	https://onlinecourses.nptel.ac.in/noc23_lw03/preview
14.	AIMLOE34	noc23-ma72	Non-Parametric Statistical Inference	Prof. Niladri Chatterjee	IITD	https://onlinecourses.nptel.ac.in/noc23_ma76/preview
15.	AIMLOE35	noc23-mg122	Introduction To Marketing Essentials	Prof. Zillur Rahman	IITR	https://onlinecourses.nptel.ac.in/noc23_mg122/preview
16.	AIMLOE36	noc23-te11	Science Of Clothing Comfort	Prof. Apurba Das	IITD	https://onlinecourses.nptel.ac.in/noc23_te11/preview

Credit Distribution (Semester wise) (Minimum Common Programme)

S.No.	Semester Number	Credit
1	Semester I	23
2	Semester II	23
3	Semester III	24
4	Semester IV	22
5	Semester V	25
6	Semester VI	25
7	Semester VII	16
8	Semester VIII	15
Total Credit		**173**

Note: Students, who have joined the job/internship/ summer training, can earn credits from MOOCS courses in the semesters.

Course Structure

Branch/Course: Artificial Intelligence and Machine Learning

Semester-I (First Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Basic Science Course	EPHY101	Engineering Physics	3+0
2.	Basic Science Course	EPHY151	Engineering Physics Lab	0+1
3.	Basic Science Course	EMAT103	Mathematics for Data Science	3+0
4.	Engineering Science Course	CSE101	Programming for Problem Solving	3+0
5.	Engineering Science Course	CSE151	Programming for Problem Solving Lab	0+1
6.	Engineering Science Course	ECE102	Basic Electrical Engineering	3+0
7.	Engineering Science Course	ECE152	Basic Electrical Engineering Lab	0+1
8.	Engineering Science Course	ME103	Fundamental of Mechanical Engineering and Mechatronics	3+0
9.	SEC Course	SEC-x	SEC Course	03
10.	AEC Course	AEC-x	AEC Course	02
11.	Induction Program	-----------------------		------------
Total Credit				**23**

Semester-II (First Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Basic Science Course	ECHE101	Engineering Chemistry	3+0
2.	Basic Science Course	ECHE151	Engineering Chemistry Lab	0+1
3.	Basic Science Course	EMAT104	Mathematics for Machine Learning	3+0
4.	Humanities and Social Sciences including Management Course	HSM101	Professional Communication	3+0
5.	Humanities and Social Sciences including Management Course	HSM151	Professional Communication Lab	0+1
6.	Engineering Science Course	ECE101	Basic Electronics Engineering	3+0
7.	Engineering Science Course	ECE151	Basic Electronics Engineering Lab	0+1
8.	Professional Core Course	AIML102	Conversational AI	3+0
9.	SEC Course	SEC-x	SEC Course	03
10.	AEC Course	AEC-x	AEC Course	02
Total Credit				**23**

Branch/Course: Artificial Intelligence and Machine Learning

Semester-III (Second Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	AIML201	Artificial Intelligence	3+0
2.	Professional Core Course	AIML202	Data Structure	4+0
3.	Professional Core Course	AIML203	Programming in Python	2+0
4.	Professional Core Course	AIML204	Data Analytics using R Programming	3+0
5.	Professional Core Course	AIML205	Mathematical Logic	3+0
6.	Professional Core Course	AIML251	Artificial Intelligence Lab	0+1
7.	Professional Core Course	AIML252	Data Structure Lab	0+1
8.	Professional Core Course	AIML253	Python Programming Lab	0+1
9.	Professional Core Course	AIML254	R-Programming Lab	0+1
10.	SEC Course	SEC-x	SEC Course	03
11.	AEC Course	AEC-x	AEC Course	02
Total Credit				**24**

Semester-IV (Second Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	AIML206	Soft Computing	3+0
2.	Professional Core Course	AIML207	Operating Systems	4+0
3.	Professional Core Course	AIML208	AI Ethics and Society	2+0
4.	Professional Core Course	AIML209	Machine Learning	4+0
5.	Professional Core Course	AIML259	Machine Learning Lab	0+1
6.	Professional Core Course	AIML210	Theory of Automata and Formal Languages	4+0
7.	Professional Core Course	AIML256	Soft Computing Lab	0+1
8.	Professional Core Course	AIML257	Operating System Lab	0+1
9.	AEC Course	AEC-x	AEC Course	02
Total Credit				**22**

Branch/Course: Artificial Intelligence and Machine Learning

Semester-V (Third Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	AIML301	Reinforcement Learning	3+0
2.	Professional Core Course	AIML302	Database Management System	3+0
3.	Professional Core Course	AIML303	Design and Analysis of Algorithms	4+0
4.	Professional Core Course	AIML304	Data Mining and Data Warehousing	3+0
5.	Professional Core Course	AIML305	Natural Language Processing	3+0
6.	Professional Elective Courses		**Elective-I** (from SWAYAM/ Department of AIML)	3+0
7.	Humanities & Social Science including Management Courses	HSM301	Organization Behavior	3+0
8.	Professional Core Course	AIML352	Database Management System Lab	0+1
9.	Professional Core Course	AIML353	Design and Analysis of Algorithm Lab	0+1
10.	Project work, Seminar and internship in industry or elsewhere	AIMLMP1	AIML Mini Project – I	0+1
		Total Credit		**25**

Semester-VI (Third Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	AIML306	Distributed System	3+0
2.	Professional Core Course	AIML307	Neural Network & Deep Learning	3+0
3.	Professional Core Course	AIML308	Cloud Computing	3+0
4.	Professional Core Course	AIML309	Image Processing	3+0
5.	Professional Elective Course		**Elective-II** (from SWAYAM/ Department of AIML)	3+0
6.	Open Elective Course		**Open Elective-I (from Department)** **or** **Open Elective-I (from SWAYAM)**	3+0
7.	Humanities & Social Science including Management Courses	HUV102	Universal Human Values-II: Understanding Harmony and Ethical Human Conduct	3+0
8.	Professional Core Course	AIML357	Neural Network & Deep Learning Lab	0+1
9.	Professional Core Course	AIML358	Cloud Computing Lab	0+1
10.	Project work, Seminar and internship in industry or elsewhere	AIMLMP2	AIML Mini Project-II	0+2
		Total Credit		**25**

Branch/Course: Artificial Intelligence and Machine Learning

Semester-VII (Fourth Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Elective Course		**Elective-I** (from SWAYAM/ Department of AIML)	3+0
2.	Professional Elective Course		**Elective-I** (from SWAYAM/ Department of AIML)	3+0
3.	Open Elective Courses		**Open Elective-II (from Department)** **or** **Open Elective-II (from SWAYAM)**	3+0
4.	Project work, Seminar and internship in industry or elsewhere	AIML S401	Seminar	0+2
5.	Project work, Seminar and internship in industry or elsewhere	AIML IND401	Industrial Training	0+2
6.	Project work, Seminar and internship in industry or elsewhere	AIML P402	AIML Major Project-I	0+3
		Total Credit		**16**

Semester-VIII (Fourth Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Elective Courses		**Elective-II** (from SWAYAM/ Department of AIML)	3+0
2.	Open Elective Courses		**Open Elective-III (from Department)** **or** **Open Elective-I (from SWAYAM)**	3+0
3.	Open Elective Courses		**Open Elective-IV (from Department)** **or** **Open Elective-I (from SWAYAM)**	3+0
4.	Project work, Seminar and internship in industry or elsewhere	AIML P403	AIML Major Project-II	0+6
		Total Credit		**15**

Mathematics for Data Science (EMAT103)

Credit: 3+0

Syllabus

Course Objective: Students will be able to understand the basic mathematics for understanding the data science, and to use integration in the calculation of area, volume, mass, and center of gravity and to apply multivariable calculus to study the nature of multivariable functions Students will finally understand the concept of Differential equation and its application.

1. Basic of Artificial Intelligence and Data Science, Set Theory, Number system, Sets and their operations, Relations and functions - Relations and their types, Functions and their types. Function of One variable, Function of two variables, Graphs and Tangents.

2. Calculus of functions of single variable: Review of limits, continuity, and differentiability. Mean value theorems: Rolle's Theorem, Lagrange's theorem, Cauchy's theorem, Taylor's theorem with remainders, indeterminate forms, curvature, curve tracing. Fundamental theorem of Integral calculus, mean value theorems of integral calculus, evaluation of definite integrals and various applications, Improper integrals: Beta and Gamma functions, differentiation under integral sign.

3. Calculus of Functions of Several Variables: Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, Tangent plane and normal line. Total differentiation, chain rules, Taylor's formula, maxima and minima, Lagrange's method of undetermined multipliers. Double and triple integrals, Jacobian, change of order of integration, change of variables, application to area, volumes, Mass, Centre of gravity.

4. Differential equation and its modelling with curve fitting: Modelling with Differential Equations, Direction Fields and Euler's Method, Linear and Bernoulli's differential equations, Nonlinear differential equations, Polar curves, angle between the radius vector and the tangent, angle between two curves. Pedal equations. Curvature and Radius of curvature - Cartesian, Parametric, Polar and Pedal forms.

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

CO1: To analyze the nature (convergence or divergence) of a sequence or series.
CO2: To apply mean value theorems in the study of motion of an object.
CO3: To use integration in the calculation of area, volume, mass, and center of gravity.
CO4: To apply multivariable calculus to study the nature of multivariable functions.

References

1. Kreyszig, E., Advanced Engineering Mathematics, John Wiley & Sons
Reference Books:
1. Piskunov, N., Differential and Integral calculus, Mir publishers Moscow (Vol. 1, Vol. 2)

Mathematics for Machine Learning (EMAT104)

Credit: 3+0

Course Objective: Students will be able to learn the fundamentals of mathematical concepts used in the area of machine Learning. Students will learn the concept of probability, statistics and linear algebra for machine learning.

Syllabus

1. Introduction to Machine Learning, Introduction to linear equations, Intercepts and slopes, System of equations, Exponentials, radicals and logarithms, Polynomials, Polynomial operations, Factorizations, Introduction to quadratic equations, Quadratic Functions, Maxima and Minima.

2. Probability Basic rules and axioms events, sample space, dependent and independent events, conditional probability, Random variables- continuous and discrete, expectation, variance, distributions- joint and conditional, Bayes' Theorem, Popular distributions-binomial, Bernoulli, poisson, exponential, Gaussian.

3. Statistics Fundamentals of Data: Collection, Summarization, and Visualization; Sampling and Sampling Distributions, Central Limit Theorem; Methods of Estimation, Unbiased estimators; Confidence Interval Estimation: Z-interval, t-interval; Hypothesis Testing, Types of Errors, Rejection Region Approach and p-value Approach.

4. Linear Algebra: Vector space, subspaces, linear dependence and independence of vectors, matrices, projection matrix, orthogonal matrix, idempotent matrix, partition matrix and their properties, quadratic forms, systems of linear equations and solutions; Gaussian elimination, eigenvalues and eigenvectors, determinant, rank, nullity, projections, LU decomposition, singular value decomposition.

Course Outcomes (CO): At the end of this course students will be able to:

CO1: To develop simple algorithms for arithmetic and logical problems. **CO2:** To translate the algorithms to programs & execution (in C language). **CO3:** To implement conditional branching, iteration and recursion.
CO4: To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
CO5: To use arrays, pointers and structures to develop algorithms and programs.

References

1. Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.
2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong,2020, Mathematics For Machine Learning.MIT-Press.

Conversational AI (AIML102)

Credit: 3+0

Syllabus

Course Objective: Students will be able to Students will understand the concepts of chatbot designing and able to build their own chatbots. Students will be able to deploy chatbot for its practical use.

1: Conversational Design Process: Introduction to virtual assistant/chatbot, use cases, what is conversational design, conversational design process, designing conversational flows, writing the script, designing your conversations, Introduction to Dialog Flow, Setting up Dialog Flow.

2: Building blocks of Interaction models: Agents, types of Intents, creating Intents, training phrases, Entities, configuring rich responses, small talk and salutations, Configuring and testing Intents on Google Assistant, Working on Connected Flows.

3: Linear and Non-linear dialogue: Actions & Parameters, understanding slot filling, context, extended Lead Generation, linear dialogue, nonlinear Dialogue, webhook, Fulfilment.

4: Fulfilment: Fulfilment using webhook, basic setup of webhook code, Extracting parameter values and structuring responses, fulfilment using cloud function

5: Deployment: Introduction to Heroku, Deploying to Heroku, Deploying on Alexa, Re-training , Validation & Testing.

Course Outcomes:
CO1: Students will understand the concepts of chatbot designing.
CO2: Students will be able to build their own chatbots.
CO3: Students will be able to deploy chatbot for its practical use.

References

1. Hands-on chatbot with Google Dialogflow, Loonycorn, O'Reilly, Packt publishing.
2. Hands-on chatbots and conversational UI development, Srini Janarthanam, Packt publishing.

Artificial Intelligence (AIML201)

Credit: 3+0

Course Objective: This course sheds light on the fundamentals of Artificial Intelligence and its applications in various areas. The student will learn to apply knowledge representation techniques and problem-solving strategies to common AI applications.

Syllabus

1. **Introduction:** Definition – Future of Artificial Intelligence – Characteristics of Intelligent Agents– Typical Intelligent Agents – Problem Solving Approach to Typical AI problems.

2. **Problem Solving Methods:** Problem solving Methods – Search Strategies- Uninformed – Informed – Heuristics – Local Search Algorithms and Optimization Problems – Searching with Partial Observations – Constraint Satisfaction Problems – Constraint Propagation – Backtracking Search – Game Playing – Optimal Decisions in Games – Alpha – Beta Pruning – Stochastic Games.

3. **Knowledge Representation:** First Order Predicate Logic – Prolog Programming – 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.

4. **Software Agents & Applications:** Architecture for Intelligent Agents – Agent communication – Negotiation and Bargaining – Argumentation among Agents – Trust and Reputation in Multi- agent systems. **Applications:** AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing – Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving.

Course Outcome (CO): At the end of course, the student will be able

CO1: To understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents.
CO2: Understand search techniques and gaming theory.
CO3: The student will learn to apply knowledge representation techniques and problem-solving strategies to common AI applications.
CO4: Students should be aware of techniques used for classification and clustering.

References

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach||, Prentice Hall, Third Edition, 2009.
2. Bratko, "Prolog: Programming for Artificial Intelligence", Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.
3. M. Tim Jones, —Artificial Intelligence: A Systems Approach, Jones and Bartlett Publishers, Inc. First Edition, 2008
4. Nils J. Nilsson, —The Quest for Artificial Intelligence, Cambridge University Press, 2009.
5. William F. Clocksin and Christopher S. Mellish, Programming in Prolog: Using the ISO

Standard, Fifth Edition, Springer, 2003.Gerhard Weiss, ―Multi Agent Systems, Second Edition, MIT Press, 2013.

6. David L. Poole and Alan K. Mackworth, ―Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

Artificial Intelligence Lab (AIML251)

Credit: 0+1

1. Write a program to conduct uninformed and informed search.
2. Write a program to conduct game search.
3. Write a program to construct a Bayesian network from given data.
4. Write a program to infer from the Bayesian network.
5. Write a program to run value and policy iteration in a grid world.
6. Write a program to do reinforcement learning in a grid world
7. Implementation of toy problems
8. Developing agent programs for real world problems
9. Implementation of constraint satisfaction problems
10. Implementation and Analysis of DFS and BFS for an application
11. 11. Developing Best first search and A* Algorithm for real world problems
12. Implementation of minimax algorithm for an application
13. Implementation of unification and resolution for real world problems.
14. Implementation of knowledge representation schemes - use cases
15. Implementation of uncertain methods for an application
16. Implementation of block world problem

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

DATA STRUCTURE (AIML202)

Credit: 4+0

Course Objective: To provide the knowledge of basic data structures and its implementations in context of writing efficient programs.

Syllabus

1. **Introduction:** Basic Terminology, Elementary Data Organization, Built in Data Types in C. Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big Oh, Big Theta and Big Omega, Time-Space trade-off. Abstract Data Types (ADT) Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Derivation of Index Formulae for 1-D,2-D,3-D and n-D Array Application of arrays, Sparse Matrices and their representations. Linked lists: Array Implementation and Pointer Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition Subtraction & Multiplications of Single variable & two variables Polynomial.

2. **Stacks and Queues:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers, and Hanoi towers. Tradeoffs between iteration and recursion. Queues: Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

3. **Searching & Sorting:** Concept of Searching, Sequential search, Index Sequential Search, Binary Search. Concept of Hashing & Collision resolution Techniques used in Hashing. Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort and Radix Sort.

4. **Trees & Graphs:** Basic terminology used with Tree, Binary Trees, Binary Tree Representation: Array Representation and Pointer (Linked List) Representation, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree. Extended Binary Trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search. Threaded Binary trees, Traversing Threaded Binary trees. Huffman coding using Binary Tree. Concept & Basic Operations for AVL Tree, B Tree & Binary Heaps. Graphs: Terminology used with Graph, Data Structure for Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.
CO2: Discuss the computational efficiency of the sorting and searching algorithms.
CO3: Implementation of Trees and Graphs and perform various operations on these data structures.
CO4: Understanding the concept of recursion, application of recursion and its implementation and removal of recursion.

References

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India.
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
4. Thareja, "Data Structure Using C" Oxford Higher Education.
5. AK Sharma, "Data Structure Using C", Pearson Education India.
6. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
7. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.
8. P. S. Deshpandey, "C and Data structure", Wiley Dreamtech Publication.
9. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education.
10. Berztiss, AT: Data structures, Theory and Practice, Academic Press.
11. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with Applications", McGraw Hill.
12. Adam Drozdek "Data Structures and Algorithms in Java", Cengage Learning.

Data Structure Lab (AIML252)

Credit: 0+1

1. Write C Programs to illustrate the concept of the following:
2. Sorting Algorithms-Non-Recursive.
3. Sorting Algorithms-Recursive.
4. Searching Algorithms.
5. Implementation of Stack using Array.
6. Implementation of Queue using Array.
7. Implementation of Circular Queue using Array.
8. Implementation of Stack using Linked List.
9. Implementation of Queue using Linked List.
10. Implementation of Circular Queue using Linked List.
11. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
12. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Programming in Python (AIML203)

Credit: 2+0

Course Objectives: This course enables students to solve problems in Python.

Syllabus

1. **Programming Basics and Decision Making :** Key features and applications of Python, Python Editors and Compilers (Interpreters), Using different offline and online Python IDE, Interacting with Python programs, Data types: Numeric, Boolean, Strings, Lists, Sets, Tuples, Dictionary; Variables: Declaration and initialization; Other concepts: Operators, Expressions, Indentation, Comments, Casting, Simple Statements: Taking inputs from user, Displaying outputs; Conditional statements: If...Else. Introduction to Linux: Understanding Linux, Installation of Python on Linux

2. **Control Flow and Other Programming Concepts** For Loops, While Loops, Break, Continue; Array**:** Looping Array elements, Array methods; Functions: Local and Global Variables, Built-in functions, User defined functions, Declaration of a function, Defining the function, Calling of the function, Functions with arguments, Recursion.

3. **OOP and File Handling:** Classes and objects, attributes and methods, constructors and destructors, inheritance, polymorphism, Exception Handling**:** Try...Except; Management of text files: Type of files, various file operations on text files, creating a text file, opening a file, closing a file, reading a text file, writing into a text file, copying a file to another file.

4. **Advance Concepts:** Problem solving**-** Use of Python to solve real time problems, How Python helps to research problems, Creating various types of graphs corresponding to any data to show different kinds of results and analysis; Data Analysis**:** Understanding problems of data science and machine learning, Creating codes in Python for various data analysis problems, Other advance programs.

Course Outcomes:

CO1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
CO2: Demonstrate proficiency in handling Strings and File Systems.
CO3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
CO4: Interpret the concepts of Object-Oriented Programming as used in Python.

References

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.
2. Think Python, Allen Downey, Green Tea Press.
3. Introduction to Python, Kenneth A. Lambert, Cengage
4. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
5. Learning Python, Mark Lutz, O'Really
6. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist '', 2nd edition, Updated for Python 3, Shroff/O 'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
7. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated

for Python 3.2, Network Theory Ltd., 2011.
8. John V Guttag, ―Introduction to Computation and Programming Using Python ‟, Revised and expanded Edition, MIT Press, 2013.

Python Programming Lab (AIML253)

Credit: 0+1

1. Krishna and his five friends have decided to go for an industrial visit by sharing the expenses of the fuel equally. Write a Python program to calculate the amount (in Rs) each of them needs to put in for the complete journey. The program should also display **True**, if the amount to be paid by each person is divisible by 3, otherwise it should display **False**.
   Hint: Use the relational operators in the print statement.
   Assumptions: Assume that mileage of the vehicle, amount per litre of fuel and distance for one way are given.

2. Write a python program to find the best of two test average marks out of three test's marks accepted from the user.

3. Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

4. Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display a suitable error message if the condition for input value is not followed.

5. Develop a Python program to convert binary to decimal, octal to hexadecimal using functions.

6. Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

7. Write a Python program to implement insertion sort and merge sort using lists.

8. Write a program to convert roman numbers into integer values using dictionaries.

9. Write a function called is phonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

10. Develop a Python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com).

11. Write a Python program to accept a file name from the user and perform the following operations
    a. Display the first N line of the file.
    b. Find the frequency of occurrence of the word accepted from the user in the file.

12. Write a Python program to create a ZIP file of a particular folder which contains several files inside it.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Data Analytics using R Programming (AIML204)

Credit: 3+0

Course Objective: The objective of this course is to teach students how to think logically about data and analyze it. The course stresses on analytics of data and reasoning and describes different ways in which problems could be solved.

Syllabus

1. Introduction to Data Analytics: Sources and nature of data, classification of data, characteristics of data, introduction to Big Data platform, need of data Analytics, evolution of analytic scalability, analytic process and tools, analysis vs reporting, modern Data analytic tools, applications of data analytics. Data Analytics Lifecycle: Need, key roles for successful analytic projects, various phases of data Analytics lifecycle – discovery, data preparation, model planning, model building, communicating Results, operationalization.

2. Data Analysis: Regression modeling, multivariate analysis, Bayesian modeling, inference and Bayesian networks, support vector and kernel methods, analysis of time series: linear systems Analysis & nonlinear dynamics, rule induction, neural networks: learning and generalisation, Competitive learning, principal component analysis and neural networks, fuzzy logic: extracting Fuzzy models from data, fuzzy decision trees, stochastic search methods.

3. Mining Data Streams: Introduction to streams concepts, stream data model and architecture, stream Computing, sampling data in a stream, filtering streams, counting distinct elements in a stream, Estimating moments, counting oneness in a window, decaying window, Real-time Analytics Platform (RTAP) applications, Case studies – real time sentiment analysis, stock market predictions.

4. Frequent Item sets and Clustering: Mining frequent item sets, market-based modelling, Apriori Algorithm, handling large data sets in main memory, limited pass algorithm, counting frequent item Sets in a stream, clustering techniques: hierarchical, K-means, clustering high dimensional data, CLIQUE and ProCLUS, frequent pattern-based clustering methods, clustering in non-Euclidean Space, clustering for streams and parallelism.

References and References:

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer
2. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press.
3. Bill Franks, Taming the Big Data Tidal wave: Finding Opportunities in Huge Data Streams with Advanced
Analytics, John Wiley & Sons.
4. John Garrett, Data Analytics for IT Networks: Developing Innovative Use Cases, Pearson Education
5. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley

R Programming Lab (AIML254)

Credit: 0+1

1. Download and install R-Programming environment and install basic packages using install Packages() Command in R Programming.
2. Write a R program to take input from the user and display the values. Also print the version of R installation.
3. Learn all the basics of R-Programming (Data types, Variables, Operators etc.)
4. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 61.
5. Create a function to print squares of numbers in sequence.
6. Write a program to find list of even numbers from 1 to n using R-Loops.
7. Write a R program to create three vectors a, b, c with 3 integers. Combine the three vectors to become a 3 x 3 matrix where each column represents a vector. Print the content of the matrix.
8. Write a R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array.
9. Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.
10. Write a program to join columns and rows in a data frame using cbind() and rbind() in R.
11. Implement different String Manipulation functions in R.
12. Implement different data structures in R (Vectors, Lists, Data Frames)
13. Write a program to read a csv file and analyze the data in the file in R.
14. Create pie chart and bar chart using R.
15. Write a R program to create a simple bar plot on four subject marks.
16. Create a data set and do statistical analysis on the data using R.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Mathematical Logic (AIML205)

Credit: 3+0

Course Objectives: Objective of this course is to develop a strong foundation in logic, master proof systems and logical calculations, and analyses axiomatic theories and computability.

Syllabus

1: Propositional Logic: Syntax, Unique parsing, Semantics, Equivalences, Consequences, Calculations, Informal proofs. Normal Forms and Resolution: Clauses, CNF and DNF representations, Adequacy of Calculations, SAT, Resolution refutation, Adequacy of resolution.

2: Proof Systems: Axiomatic system PC, Adequacy of PC, Analytic tableau PT, Adequacy of PT, Compactness of PL. First Order Logic: Syntax of FL, Scope and binding, Substitutions, Semantics of FL, Quantifier laws, Equivalences, Consequences.

3: Normal Forms in FL: Calculations, Informal proofs, Prenex forms, Skolem forms, Herbrand Theorem, Skolem-Lowenheim theorem, Resolution in FL.

4: Proof Systems for FL: Axiomatic system FC, Analytic tableau FT, Adequacy of FC and FT, Compactness in FL, Axiomatic Theories: Undecidabilty of FL, Godel incompleteness theorems.

Course Outcomes: On completion of the course the student will be able to:

CO1: Understand propositional logic, including syntax, semantics, equivalences, and logical consequences.
CO2: Master normal forms (CNF, DNF) and resolution methods for satisfiability, refutation, and logical Calculations.
CO3: Learn first-order logic (FOL), including syntax, semantics, scope, binding, quantifier laws, and logical Transformations.
CO4: Explore proof systems, including axiomatic systems and analytic tableaux, and understand their Adequacy, compactness, and limitations in both propositional and first-order logic.

References:
1. A.Singh, Logics for Computer Science, PHI, 2004.
2. A singh, and C.Goswami,Fundamentals of Logic, ICPR New Delhi, 1998.

Soft Computing (AIML206)

Credit: 3+0

Course Objective: This course enables students to adopt nature-inspired computing models to solve specific problems.

Syllabus

1. Introduction: What is computational intelligence?- Biological basis for neural networks- Biological versus Artificial neural networks- Biological basis for evolutionary computation- Behavioral motivations for fuzzy logic, Myths about computational intelligence- Computational intelligence application areas, Evolutionary computation, computational intelligence-Adoption, Types, self-organization and evolution, Historical views of computational intelligence, Computational intelligence and Soft computing versus Artificial intelligence and Hard computing.

2. Genetic Algorithm, Genetic Representations, Initial Population, Fitness Function, Selection and Reproduction, Genetic Operators(Selection, Crossover, Mutation), Artificial Immune Systems, Other Algorithms Harmony Search, Honey-Bee Optimization, Memetic Algorithms, Co-evolution, Multi Objective Optimization, Artificial Life, Constraint Handling, Collective Systems Collective Behaviour and Swarm Intelligence, Particle Swarm Optimization and Ant Colony Optimization, Artificial evolution of Competing Systems, Artificial Evolution of cooperation and competition. Recent topics from research papers.

3. Fuzzy Logic-I (Introduction): Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp relations, Fuzzy to Crisp conversion. Fuzzy Logic –II (Fuzzy Membership, Rules): Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzy fications & Defuzzificataions, Fuzzy Controller, Industrial applications

4. Neural Network Concepts and Paradigms- What Neural Networks are? Why they are useful, Neural network components and terminology- Topologies - Adaptation, Comparing neural networks and other information Processing methods- Stochastic- Kalman filters , Perceptron Network, Adaline Network, Madaline Network, Back Propagation Network, Linear and Nonlinear regression - Correlation - Bayes classification -Vector quantization - Radial basis functions Preprocessing - Post processing.

References

1. Dario Floreano, Claudio Mattiussi, “Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies”, MIT Press, 2008.
2. Eberhart, E. and Y. Shi., “Coputational Intelligence: Concepts and Implementations”, Morgan Kauffmann, San Diego, 2007.
3. D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning”, AddisonWesley, 1989
4. R. C. Ebelhart et al., “Swarm Intelligence”, Morgan Kaufmann, 2001. • M. Dorigo and T. Stutzle, “Ant Colony Optimization”, A Bradford Book, 2004.
5. Leandro Nunes De Castro, Fernando Jose Von Zuben, “Recent Developments in Biologically Inspired Computing”, Idea Group Publishing, 2005.

Soft Computing Lab (AIML256)

Credit: 0+1

1. Implement Perceptron Network
2. Implement Adaline Network
3. Implement Madaline Network for XOR Function
4. Implement Back Propagation Network for XOR Function using Bipolar Inputs and Binary Targets.
5. Implement Kohonen Self-Organizing Feature Map
6. Genetic Algorithm Implementation to Maximize F(X1,X2)=4X1+3X2
7. Genetic Algorithm Implementation to maximize $F(X)=X^2$
8. Genetic Algorithm Implementation to minimize $F(X)=X^2$
9. Implementation of Traveling Salesman Problem.
10. Genetic Algorithm Implementation to Minimize F(X1,X2)=4X1+3X2

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Operating Systems (AIML207)

Credit: 4+0

Course Objective: Student will learn about processes and processor management, concurrency and synchronization, memory management schemes, and secondary storage management, security and protection.

Syllabus

1. **Introduction:** Operating system and functions, Classification of Operating systems- Batch, Interactive, Time-sharing, Real-Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.

2. **Concurrent Processes:** Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

3. **CPU Scheduling & Disk Scheduling:** Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock. Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling.

4. **Memory Management:** Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

Course Outcome (CO): At the end of course, the student will be able to:

CO1: Understand the structure and functions of OS.
CO2: Learn about Processes, Threads and Scheduling algorithms.
CO3: Understand the principles of concurrency and Deadlocks.
CO4: Learn various memory management schemes.

References

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley.
2. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education.
3. Harvey M Dietel, "An Introduction to Operating System", Pearson Education.
4. D M Dhamdhere, "Operating Systems: A Concept based Approach", 2nd Edition, TMH.
5. William Stallings, "Operating Systems: Internals and Design Principles", 6th Edition, Pearson

Operating System Lab (AIML257)

Credit: 0+1

1. Study of hardware and software requirements of different operating systems (UNIX, LINUX, WINDOWS XP/ WINDOWS 7/8/10)
2. Execute various UNIX system calls for
   - ***a.*** Process management
   - ***b.*** File management
   - ***c.*** Input/output Systems calls
3. Implement CPU Scheduling Policies:
   - ***a.*** SJF
   - ***b.*** Priority
   - ***c.*** FCFS
   - ***d.*** Multi-level Queue
4. Implement file storage allocation technique:
   - ***a.*** Contiguous (using array)
   - ***b.*** Linked –list (using linked-list)
   - ***c.*** Indirect allocation (indexing)
5. Implementation of contiguous allocation techniques:
   - ***a.*** Worst-Fit
   - ***b.*** Best- Fit
   - ***c.*** First- Fit
6. Calculation of external and internal fragmentation
   - ***a.*** Free space list of blocks from system
   - ***b.*** List process file from the system
7. Implementation of compaction for the continually changing memory layout and calculate total movement of data.
8. Implementation of resource allocation graph RAG).
9. Implementation of Banker's algorithm.
10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
11. Implement the solution for Bounded Buffer (producer-consumer) problem using inter process communication techniques-Semaphores.
12. Implement the solutions for Readers-Writer's problem using inter process communication technique -Semaphore.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

AI Ethics and Society (AIML208)

Credit: 2+0

Course Objective:

To introduce students to the ethical principles and societal impact of AI technologies. To analyze bias, fairness, and privacy issues in AI models and data-driven decision-making. To explore AI's role in society, including its impact on employment, governance, and cultural aspects.

Syllabus

1. Foundations of AI Ethics

Introduction to AI and Ethics: Definition and scope of AI ethics, historical context of AI development, importance of ethical considerations in AI, ethical challenges in machine learning and automation, AI's impact on decision-making and human autonomy. Ethical Theories and AI: Utilitarianism, deontology, virtue ethics, fairness, accountability, and transparency (FAT) in AI, principles of responsible AI development, ethical dilemmas in AI applications, trade-offs between privacy and utility.

2. Bias, Fairness, and Privacy in AI

Bias and Fairness in AI: Sources of bias in AI models, algorithmic fairness and discrimination, mitigating bias in data and model training, fairness-aware machine learning techniques, real-world examples of biased AI decisions. Privacy and Surveillance: Data privacy in AI systems, ethical concerns in data collection and processing, GDPR and AI regulations, AI-driven surveillance and its societal implications, privacy-preserving AI techniques (differential privacy, federated learning).

3. Societal Impact and AI Governance

AI and Society: AI's impact on employment and the future of work, AI in healthcare, finance, and education, social and cultural implications of AI-driven automation, ethical concerns in AI-generated content and misinformation, AI and sustainability. AI Governance and Regulation: Global AI policies and frameworks, role of government and industry in AI regulation, AI ethics guidelines by organizations (EU, IEEE, UNESCO), case studies of AI governance failures, ethical AI deployment in businesses and public services.

4. Ethical AI Design and the Future of AI

Ethical AI Development and Human-Cantered AI: Explainable AI (XAI) and interpretability, human-in-the-loop AI systems, designing AI for social good, ethical AI design principles, AI for accessibility and inclusion. The Future of AI and Ethical Challenges: Emerging ethical challenges in AI (AGI, autonomous weapons, deepfake technology), ethical concerns in AI-driven decision-making, philosophical perspectives on AI consciousness, ensuring ethical AI development for future generations.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: To introduce students to the ethical principles and societal impact of AI technologies.
CO2: To analyse bias, fairness, and privacy issues in AI models and data-driven decision-making.
CO3: To explore AI's role in society, including its impact on employment, governance, and cultural aspects.
CO4: To understand AI policies, regulations, and global ethical frameworks for responsible AI development.

References

1. Mark Coeckelbergh – *AI Ethics*, MIT Press, 2020.
2. Nick Bostrom – *Superintelligence: Paths, Dangers, Strategies*, Oxford University Press, 2014.
3. Virginia Eubanks – *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*, St. Martin's Press, 2018.
4. Kate Crawford – *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*, Yale University.

Machine Learning (AIML209)

Credit: 4+0

Course Objectives: This course provides an advanced level of understanding to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications

Syllabus

1. **Introduction:** Learning, Types of Learning, well defined learning problems, Designing a Learning System, History of ML, Introduction of Machine Learning Approaches – (Artificial Neural Network, Clustering, Reinforcement Learning, Decision Tree Learning, Bayesian networks, Support Vector Machine, Genetic Algorithm), Issues in Machine Learning and Data Science Vs Machine Learning.

2. **Regression and Decision Tree Learning:** Linear Regression and Logistic Regression BAYESIAN LEARNING - Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm. SUPPORT VECTOR MACHINE: Introduction, Types of support vector kernel – (Linear kernel, polynomial kernel, and Gaussian kernel), Hyperplane – (Decision surface), Properties of SVM, and Issues in SVM. DECISION TREE LEARNING: Decision tree learning algorithm, Inductive bias, Inductive inference with decision trees, Entropy and information theory, Information gain, ID-3 Algorithm, Issues in Decision tree learning. INSTANCE-BASED LEARNING – k-Nearest Neighbor Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning.

3. **Artificial Neural Networks:** Perceptron's, Multilayer perceptron, Gradient descent and the Delta rule, Multilayer networks, Derivation of Backpropagation Algorithm, Generalization, Unsupervised Learning – SOM Algorithm and its variant; DEEP LEARNING - Introduction, concept of convolutional neural network , Types of layers – (Convolutional Layers , Activation function , pooling , fully connected) , Concept of Convolution (1D and 2D) layers, Training of network, Case study of CNN for e.g. on Diabetic Retinopathy, Building a smart speaker, Self-deriving car etc.

4. **Reinforcement Learning:** Introduction to Reinforcement Learning, Learning Task, Example of Reinforcement Learning in Practice, Learning Models for Reinforcement – (Markov Decision process, Q Learning - Q Learning function, Q Learning Algorithm), Application of Reinforcement Learning, Introduction to Deep Q Learning. GENETIC ALGORITHMS: Introduction, Components, GA cycle of reproduction, Crossover, Mutation, Genetic Programming, Models of Evolution and Learning, Applications.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Understand the need for machine learning for various problem solving.
CO2: Understand a wide variety of learning algorithms and how to evaluate models generated from data.
CO3: Understand the latest trends in machine learning.
CO4: Design appropriate machine learning algorithms and apply the algorithms to a real-world problem.

References

1. Tom M. Mitchell, —Machine Learning, McGraw-Hill Education (India) Private Limited, 2013.
2. Ethem Alpaydin, —Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press 2004.
3. Stephen Marsland, —Machine Learning: An Algorithmic Perspective, CRC Press, 2009.
4. Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag.

Machine Learning Lab (AIML259)

Credit: 0+1

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in python classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
11. Carry out the performance analysis of classification algorithms on a specific dataset.

Note: The instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Theory of Automata and Formal Languages (AIML210)

Credit: 4+0

Course Objective: This course focuses on the basic theory of Computer Science and formal methods of computation like automata theory, formal languages, grammars and Turing Machines. The objective of this course is to explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

Syllabus

1. **Basic Concepts and Automata Theory:** Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata, Myhill-Nerode Theorem, Simulation of DFA and NFA.

2. **Regular Expressions and Languages:** Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem, Algebraic Method Using Arden's Theorem, Regular and Non-Regular Languages- Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties, Finite Automata and Regular Languages, Regular Languages and Computers, Simulation of Transition Graph and Regular language.

3. **Regular and Non-Regular Grammars:** Context Free Grammar (CFG)-Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars-Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms- Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Chomsky Hierarchy, Programming problems based on the properties of CFGs.

4. **Push Down Automata and Properties of Context Free Languages:** Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA, Deterministic Pushdown Automata (DPDA) and Deterministic Context free Languages (DCFL), Pushdown Automata for Context Free Languages, Context Free grammars for Pushdown Automata, two stack Pushdown Automata, Pumping Lemma for CFL, Closure properties of CFL, Decision Problems of CFL, Programming problems based on the properties of CFLs. Turing Machines and Recursive Function Theory: Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Turing Machine as Computer of Integer Functions, Universal Turing machine, Linear Bounded Automata, Church's Thesis, Recursive and Recursively Enumerable language, Halting Problem, Post's Correspondence Problem, Introduction to Recursive Function Theory.

Course Outcome (CO): At the end of course, the student will be able to
CO1: Analyse and design finite automata, pushdown automata, Turing machines, formal languages.
CO2: Analyse and design, Turing machines, formal languages, and grammars.
CO3: Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.

CO4: Prove the basic results of the Theory of Computation.

References

1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
2. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill
3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI
4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International.

Reinforcement Learning (AIML301)

Credit: 3+0

Course Objective:

To understand the fundamental concepts and principles of reinforcement learning. To explore model free and model-based RL techniques, including Monte Carlo and temporal difference methods. To analyse deep reinforcement learning algorithms and their applications.

Syllabus

1. Fundamentals of Reinforcement Learning: Introduction to Reinforcement Learning (RL): Basic concepts, differences between supervised, unsupervised, and reinforcement learning, agent-environment interaction, Markov Decision Processes (MDPs), reward functions, policy vs. value function, exploration vs. exploitation. Dynamic Programming for RL: Bellman equations, policy evaluation, policy iteration, value iteration, discounted and undiscounted rewards, optimality equations.

2 Model-Free RL and Approximate Solutions: Monte Carlo and Temporal Difference Learning: Monte Carlo methods, first-visit and every-visit MC, TD(0) learning, n-step TD, SARSA, Q-learning, Expected SARSA, trade-offs between MC and TD methods. Function Approximation in RL: Need for function approximation, linear function approximation, deep Q-networks (DQN), policy gradient methods, actor-critic methods, in deep RL.

3. Advanced Reinforcement Learning Algorithms: Policy Optimization Methods: REINFORCE algorithm, trust region policy optimization (TRPO), proximal policy optimization (PPO), soft actor-critic (SAC), deep deterministic policy gradient (DDPG). Multi-Agent and Hierarchical RL: Cooperative and competitive multi-agent RL, self-play and adversarial training, hierarchical RL, options framework and game AI.

4. Applications and Future Trends in RL: Real-World Applications of RL: RL in robotics, self-driving cars, recommendation systems, healthcare, trading and finance, energy optimization, automated control systems. Ethical Considerations and Challenges: Challenges in reward shaping, sample efficiency, safety constraints in RL, ethical concerns in AI decision-making, generalization in RL models, ongoing research trends and future directions in reinforcement learning.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the mathematical foundation and principles of reinforcement learning.
CO2: Implement fundamental RL algorithms like Q-learning, SARSA, and temporal difference learning.
CO3: Apply deep reinforcement learning methods for decision-making and control problems.
CO4: Design and optimize policy-based RL algorithms for real-world applications.

References

1. Richard S. Sutton, Andrew G. Barto – *Reinforcement Learning: An Introduction*, MIT Press, 2018.
2. David Silver's RL Course Notes – Available online, University College London (UCL).
3. Francois-Lavet, Dumoulin-Thurston, Holland, et al. – *An Introduction to Deep Reinforcement Learning*, Foundations and Trends in Machine Learning, 2018.
4. Csaba Szepesvári – *Algorithms for Reinforcement Learning*, Morgan & Claypool Publishers, 2010.
5. Nicolas Heess, Timothy P. Lillicrap, Tom Erez, Yuval Tassa, David Silver, Martin Riedmiller – *Deep Reinforcement Learning with Continuous Control*, 2015.

Database Management System (AIML302)

Credit: 3+0

Course Objectives:

To explain basic database concepts, applications, data models, schemas and instances. To demonstrate the use of constraints and relational algebra operations. To emphasize the importance of normalization in databases.

Syllabus

1. **Introduction:** Overview, Database System vs. File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence and Database Language and Interfaces, Data Definitions Language, DML, Overall Database Structure. Data Modelling Using the Entity Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of an ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.

2. **Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus. Introduction on SQL: Characteristics of SQL, Advantage of SQL. SQl Data Type and Literals. Types of SQL Commands. SQL Operators and Their Procedure. Tables, Views and Indexes. Queries and Subqueries. Aggregate Functions. Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL.

3. **Database Design & Normalization:** Functional dependencies, normal forms, first, second, 8 third normal forms, BCNF, inclusion dependence, lossless join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

4. **Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling. Distributed Database: Distributed Data Storage, Concurrency Control, Directory System.Concurrency Control Techniques: Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction, Case Study of Oracle.

Course Outcome (CO): At the end of course, the student will be able to
CO1: Apply knowledge of databases for real life applications.
CO2: Apply query processing techniques to automate the real time problems of databases.
CO3: Identify and solve the redundancy problem in database tables using normalization.
CO4: Understand the concepts of transactions, their processing so they will be familiar with broad range of database management issues including data integrity, security and recovery.

References

1. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill.
2. Date C J, "An Introduction to Database Systems", Addision Wesley.
3. Elmasri, Navathe, "Fundamentals of Database Systems", Addision Wesley.
4. O'Neil, Databases, Elsevier Pub.
5. Ramakrishnan"Database Management Systems",McGraw Hill.
6. Leon & Leon,"Database Management Systems", Vikas Publishing House.
7. Bipin C. Desai, "An Introduction to Database Systems", Galgotia Publications.
8. Majumdar & Bhattacharya, "Database Management System", TMH.

Database Management Systems Lab (AIML352)

Credit: 0+1

1. Installing oracle/ MYSQL
2. Creating Entity-Relationship Diagram using case tools.
3. Writing SQL statements Using ORACLE /MYSQL:
    a. Writing basic SQL SELECT statements.
    b. Restricting and sorting data.
    c. Displaying data from multiple tables.
    d. Aggregating data using group functions.
    e. Manipulating data.
    f. Creating and managing tables.
4. Implementing Normalization
5. Creating cursor
6. Creating procedure and functions
7. Creating packages and triggers
8. Design and implementation of payroll processing system
9. Design and implementation of Library Information System
10. Design and implementation of Student Information System
11. Automatic Backup of Files and Recovery of Files
12. Sample of Mini project (Design & Development of Data and Application) for following:
    a. Inventory Control System.
    b. Material Requirement Processing.
    c. Hospital Management System.
    d. Railway Reservation System.
    e. Personal Information System.
    f. Web Based User Identification System.
    g. Timetable Management System.
    h. Hotel Management System

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner

Design and Analysis of Algorithms (AIML303)

Credit: 4+0

Course Objective: The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Syllabus

1. Introduction: Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time.

2. Advanced Data Structures: Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps, Tries, Skip List.

3. Divide and Conquer with Examples Such as Sorting, Matrix Multiplication, Convex Hull and Searching. Greedy Methods with Examples Such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths - Dijkstra's and Bellman Ford Algorithms.

4. Dynamic Programming with Examples Such as Knapsack. All Pair Shortest Paths – Warshal's and Floyd's Algorithms, Resource Allocation Problem. Backtracking, Branch and Bound with Examples Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets.

Course Outcome (CO): At the end of course, the student will be able to
CO1: Design new algorithms, prove them correct, and analyse their asymptotic and absolute runtime and memory demands.
CO2: Find an algorithm to solve the problem (create) and prove that the algorithm solves the problem correctly (validate).
CO3: Understand the mathematical criterion for deciding whether an algorithm is efficient, and know many practically important problems that do not admit any efficient algorithms.
CO4: Apply classical sorting, searching, optimization and graph algorithms.

References

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.
2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",

3. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
4. LEE "Design & Analysis of Algorithms (POD)", McGraw Hill
5. Richard E.Neapolitan "Foundations of Algorithms" Jones & Bartlett Learning
6. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
7. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples,Second Edition, Wiley, 2006.
8. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
9. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
10. Harsh Bhasin,"Algorithm Design and Analysis",First Edition,Oxford University Press.
11. Gilles Brassard and Paul Bratley,Algorithmics:Theory and Practice,Prentice Hall,1995.

Design and Analysis of Algorithm Lab (AIML353)

Credit: 0+1

1. Program for Recursive Binary & Linear Search.
2. Program for Heap Sort.
3. Program for Merge Sort.
4. Program for Selection Sort.
5. Program for Insertion Sort.
6. Program for Quick Sort.
7. Knapsack Problem using Greedy Solution
8. Perform Travelling Salesman Problem
9. Find Minimum Spanning Tree using Kruskal's Algorithm
10. Implement N Queen Problem using Backtracking
11. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide and- conquer method works along with its time complexity analysis: worst case, average case and best case.
12. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide and- conquer method works along with its time complexity analysis: worst case, average case and best case.
13. Implement, the 0/1 Knapsack problem using
    a. Dynamic Programming method
    b. Greedy method
14. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
15. Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.
16. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
17. Write programs to
    a. Implement All-Pairs Shortest Paths problem using Floyd's algorithm.
    b. Implement Travelling Sales Person problem using Dynamic programming.
18. Design and implement to find a subset of a given set S = {Sl, S2,. ,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.
19. Design and implement to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner. It is also suggested that open source tools should be preferred to conduct the lab (C, C++ etc).

Data Mining and Data Warehousing (AIML304)

Credit: 3+0

Course Objective: The course content enables students to: Analyze the difference between On Line Transaction Processing and On-Line analytical processing. To Create Multidimensional schemas suitable for data warehousing. To Understand various data mining functionalities. To Understand in detail about data mining algorithms.

Syllabus

1) **Data Warehousing and Business Analysis:** Data warehousing Components –Building a Data warehouse –Data Warehouse Architecture – DBMS Schemas for Decision Support – Data Extraction, Cleanup, and Transformation Tools –Metadata – reporting – Query tools and Applications – Online Analytical Processing (OLAP) – OLAP and Multidimensional Data Analysis.

2) **Data Mining:** Data Mining Functionalities – Data Preprocessing – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization and Concept Hierarchy Generation- Architecture of Typical Data Mining Systems- Classification of Data Mining Systems. **Association Rule Mining: -** Efficient and Scalable Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

3) **Classification and Prediction:** Issues Regarding Classification and Prediction – Classification by Decision Tree Introduction – Bayesian Classification – Rule Based Classification – Classification by Backpropagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction – Accuracy and Error Measures – Evaluating the Accuracy of a Classifier or Predictor – Ensemble Methods – Model Section.

4) **Cluster Analysis, Mining Object, Spatial, Multimedia, Text and Web Data:** Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis. Mining Object, Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects – Spatial Data Mining – Multimedia Data Mining – Text Mining – Mining the World Wide Web.

Course Outcome (CO): At the end of course, the student will be able to

CO1: To understand the principles of Data Warehousing and Data Mining.
CO2: To be familiar with the Data warehouse architecture and its Implementation.
CO3: To know the Architecture of a Data Mining system.
CO4: To understand the various Data preprocessing Methods.

References

1. Jiawei Han, Micheline Kamber and Jian Pei "Data Mining Concepts and Techniques", Third Edition, Elsevier, 2011.
2. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint 2007.
3. K.P. Soman, Shyam Diwakar and V. Ajay "Insight into Data mining Theory and Practice", Easter Economy Edition, Prentice Hall of India, 2006.
4. G. K. Gupta "Introduction to Data Mining with Case Studies", Easter Economy Edition.

Natural Language Processing (AIML305)

Credit: 3+0

Course Objective: To understand the advanced concepts of Natural Language Processing and to apply the various concepts of NLP in other application areas.

Syllabus

1. **Introduction:** Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance, WORD LEVEL ANALYSIS : Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

2. **Syntactic Analysis:** Context Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures.

3. **Semantics and Pragmatics:** Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

4. **Basic Concepts of Speech Processing:** Speech Fundamentals: Articulatory Phonetics – Production and Classification of Speech Sounds; Acoustic Phonetics – Acoustics of Speech Production; Review of Digital Signal Processing Concepts; Short-Time Fourier Transform, Filter-Bank and LPC Methods. Speech-Analysis: Features, Feature Extraction and Pattern Comparison Techniques: Speech Distortion Measures– Mathematical and Perceptual – Log–Spectral Distance, Cepstral Distances, Weighted Cepstral Distances and Filtering, Likelihood Distortions, Spectral Distortion Using a Warped Frequency Scale, LPC, PLP And MFCC Coefficients, Time Alignment and Normalization – Dynamic Time Warping, Multiple Time – Alignment Paths. SPEECH MODELING: Hidden Markov Models: Markov Processes, HMMs – Evaluation, Optimal State Sequence – Viterbi Search, Baum-Welch Parameter Re-Estimation, Implementation Issues.

Course Outcome (CO): At the end of course, the student will be able

CO1: To learn the fundamentals of natural language processing.
CO2: To understand the use of CFG and PCFG in NLP.
CO3: To understand the role of semantics of sentences and pragmatic.
CO4: To Introduce Speech Production and Related Parameters of Speech.

References

1. Daniel Jurafsky, James H. Martin—Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, —Natural Language Processing with Python, First Edition, OReilly Media, 2009.
3. Lawrence Rabiner and Biing-Hwang Juang, "Fundamentals of Speech Recognition", Pearson Education, 2003. Daniel Jurafsky and James H Martin, "Speech and Language Processing – An Introduction To Natural Language Processing, Computational Linguistics, And Speech Recognition", Pearson Education, 2002.
4. Frederick Jelinek, "Statistical Methods of Speech Recognition", MIT Press, 1997.
5. Breck Baldwin, —Language Processing with Java and LingPipe Cookbook, Atlantic Publisher.

Distributed System (AIML306)

Credit: 3+0

Course Objective: This course is an introduction to the design of distributed systems and algorithms that support distributed computing.

Syllabus

1. **Characterization of Distributed Systems:** Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges. Architectural models, Fundamental Models. Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's & vectors logical clocks. Concepts in Message Passing Systems: causal order, total order, total causal order, Techniques for Message Ordering, Causal ordering of messages, global state, and termination detection.

2. **Distributed Mutual Exclusion:** Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non-token-based algorithms, performance metric for distributed mutual exclusion algorithms. Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized deadlock detection, distributed deadlock detection, path pushing algorithms, edge chasing algorithms.

3. **Agreement Protocols:** Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system. Distributed Resource Management: Issues in distributed File Systems, Mechanism for building distributed file systems, Design issues in Distributed Shared Memory, Algorithm for Implementation of Distributed Shared Memory.

4. **Failure Recovery in Distributed Systems & Transactions and Concurrency Control:** Concepts in Backward and Forward recovery, Recovery in Concurrent systems, obtaining consistent Checkpoints, Recovery in Distributed Database Systems. Fault Tolerance: Issues in Fault Tolerance, Commit Protocols, Voting protocols, Dynamic voting protocols. Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control. Distributed Transactions: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault - tolerant services, highly available services, Transactions with replicated data.

Course Outcome (CO) : At the end of course, the student will be able to

CO1: Provide hardware and software issues in modern distributed systems.
CO2: Get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.
CO3: Analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.
CO4: Know about Shared Memory Techniques and have sufficient knowledge about file access.

References

1. Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
2. Ramakrishna,Gehrke," Database Management Systems", McGraw Hill
3. Vijay K.Garg Elements of Distributed Computing , Wiley
4. Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design", Pearson Education
5. Tenanuanbaum, Steen," Distributed Systems", PHI

Neural Network & Deep Learning (AIML307)

Credit: 3+0

Course Objective: Deep learning is a class of machine learning algorithms which enables computers to learn from examples. Deep learning techniques have been used successfully for a variety of applications, including: automatic speech recognition, image recognition, natural language processing, drug discovery, and recommendation systems. In this course, students will learn the fundamentals of deep learning, and the main research activities in this field.

Syllabus

1. **Neural Network:** Neural Network Concepts and Paradigms- What Neural Networks are? Why they are useful, Neural network components and terminology- Topologies - Adaptation, Comparing neural networks and other information Processing methods- Stochastic- Kalman filters - Linear and Nonlinear regression - Correlation - Bayes classification -Vector quantization - Radial basis functions Preprocessing - Post processing.
2. **Introduction to Deep Networks:** Introduction to machine learning- Linear models (SVMs and Perceptrons, logistic regression)- Intro to Neural Nets: What a shallow network computes- Training a network: loss functions, back propagation and stochastic gradient descent- Neural networks as universal function approximates. DEEP NETWORKS: History of Deep Learning- A Probabilistic Theory of Deep Learning- Backpropagation and regularization, batch normalization- VC Dimension and Neural Nets-Deep Vs Shallow Networks-Convolutional Networks- Generative Adversarial Networks (GAN), Semi-supervised Learning.

3. **Dimensionality Reduction:** Linear (PCA, LDA) and manifolds, metric learning – Auto encoders and dimensionality reduction in networks - Introduction to Convnet - Architectures – AlexNet, VGG, Inception, ResNet - Training a Convnet: weights initialization, batch normalization, hyper parameter optimization.

4. **Optimization and Generalization:** Optimization in deep learning– non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM - Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning - Computational & Artificial Neuroscience. Case Study and Applications: Image net- Detection-Audio Wave Net-Natural Language Processing Word2Vec - Joint Detection-Bioinformatics- Face Recognition- Scene Understanding- Gathering Image Captions.

Course Outcome (CO): At the end of course, the student will be able

CO1: To present the mathematical, statistical and computational challenges of building neural networks.
CO2: To study the concepts of deep learning.
CO3: To introduce dimensionality reduction techniques.
CO4: To enable the students to know deep learning techniques to support real-time applications.

References

1. Cosma Rohilla Shalizi, Advanced Data Analysis from an Elementary Point of View, 2015.
2. Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
4. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

Neural Network & Deep Learning Lab (AIML357)

Credit: 1+0

1. Build a deep neural network model start with linear regression using a single variable.
2. Build a deep neural network model start with linear regression using multiple variables.
3. Write a program to convert speech into text.
4. Write a program to convert text into speech.
5. Write a program for Time-Series Forecasting with the LSTM Model.
6. Implementation of CNN
7. Implementation of RNN
8. Implementation of RESNET
9. Implementation of ALEXNET
10. Implementation of VGG 16.

Cloud Computing (AIML308)

Credit: 3+0

Course Objective: This course provides the students an insight into the basics of cloud computing along with virtualization, cloud computing is one of the fastest growing domains from a while now. It will provide the students basic understanding about cloud and virtualization along with how one can migrate over it.

Syllabus

1. **Introduction to Cloud Computing:** Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

2. **Cloud Enabling Technologies Service Oriented Architecture:** REST and Systems of Systems – Web Services – Publish, Subscribe Model – Basics of Virtualization – Types of Virtualizations – Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices –Virtualization Support and Disaster Recovery.

3. **Cloud Architecture, Services and Storage:** Layered Cloud Architecture Design – NIST Cloud Computing Reference Architecture – Public, Private and Hybrid Clouds – laa

4. IaaS - PaaS – SaaS – Architectural Design Challenges – Cloud Storage – Storage-as-a-Service – Advantages of Cloud Storage – Cloud Storage Providers – S3.

5. **Resource Management and Security in Cloud & Cloud Technologies and Advancements Hadoop:** Inter Cloud Resource Management – Resource Provisioning and Resource Provisioning Methods – Global Exchange of Cloud Resources – Security Overview – Cloud Security Challenges – Software-as-a-Service Security – Security Governance – Virtual Machine Security – IAM – Security Standards. **Cloud Technologies and Advancements Hadoop:** MapReduce – Virtual Box –– Google App Engine – Programming Environment for Google App Engine –– Open Stack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Describe architecture and underlying principles of cloud computing.
CO2: Explain the need, types and tools of Virtualization for cloud.
CO3: Describe Services Oriented Architecture and various types of cloud services.
CO4: Explain Inter cloud resources management cloud storage services and their providers Assess security services and standards for cloud computing.

References

1. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.
2. Rittinghouse, John W., and James F. Ransome, —Cloud Computing: Implementation, Management and Security, CRC Press, 2017.
3. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, —Mastering Cloud Computing, Tata Mcgraw Hill, 2013.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing – A Practical Approach, Tata Mcgraw Hill, 2009.
5. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice), O’Reilly, 2009.

Cloud Computing Lab(AIML358)

Credit: 1+0

1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not presentin CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8. Install Hadoop single node cluster and run simple applications like wordcount.

Image Processing (AIML309)

Credit: 3+0

Course Objective: Students will able to understand the basic concepts of two-dimensional signal acquisition, sampling, quantization and color model. Apply image processing techniques for image enhancement in both the spatial and frequency domains.

1. DIGITAL IMAGE FUNDAMENTALS: Steps in Digital Image Processing – Components – Elements of Visual Perception – Image Sensing and Acquisition – Image Sampling and Quantization – Relationships between pixels – Color image fundamentals – RGB, HSI models,Two-dimensional mathematical preliminaries, 2D transforms – DFT, DCT.

2. IMAGE ENHANCEMENT: Spatial Domain: Gray level transformations – Histogram processing – Basics of Spatial Filtering– Smoothing and Sharpening Spatial Filtering, Frequency Domain: Introduction to Fourier Transform– Smoothing and Sharpening frequency domain filters – Ideal, Butterworth and Gaussian filters, Homomorphic filtering, Color image enhancement.

3. IMAGE RESTORATION AND IMAGE SEGMENTATION: Image Restoration – degradation model, Properties, Noise models – Mean Filters – Order Statistics – Adaptive filters – Band reject Filters – Band pass Filters – Notch Filters – Optimum NotchFiltering – Inverse Filtering – Wiener filtering IMAGE SEGMENTATION: Edge detection, Edge linking via Hough transform – Thresholding – Region based segmentation – Region growing – Region splitting and merging – Morphological processing- erosion and dilation, Segmentation by morphological watersheds – basic concepts – Dam construction – Watershed segmentation algorithm.

4. IMAGE COMPRESSION AND RECOGNITION: Need for data compression, Huffman, Run Length Encoding, Shift codes, Arithmetic coding, JPEGstandard, MPEG. Boundary representation, Boundary description, Fourier Descriptor, Regional Descriptors – Topological feature, Texture – Patterns and Pattern classes – Recognition based on matching.

References

1. Rafael C. Gonzalez, Richard E. Woods,Digital Image Processing Pearson, 3rd Edition, 2010
2. Anil K. Jain,Fundamentals of Digital Image Processing Pearson, 2002.
3. Kenneth R. Castleman,Digital Image Processing Pearson, 2006.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,Digital Image Processing using MATLAB Pearson Education, Inc., 2011.

Syllabus of Professional Elective Courses Offered by Departments

Elective-1

(Offered by Department)

Computer Vision (EAIML116)

Credit: 3+0

Course Objective:

To understand the fundamental concepts of image processing, camera geometry, and projective transformations in computer vision. To explore stereo vision techniques, feature detection, matching, and model fitting for 3D scene reconstruction and object recognition.

Syllabus

1. Fundamentals of Image Processing & Geometric Transformations: Fundamentals of Image Processing: Introduction to Computer Vision, Digital Image Representation (RGB, Grayscale, Binary), Image Enhancement (Histogram Equalization, Filtering - Gaussian, Median, etc.), Image Restoration (Noise Removal, De-blurring), Edge Detection (Sobel, Prewitt, Canny). 2-D Projective Geometry, Homography, and Properties of Homography: Introduction to 2D Projective Geometry, Homogeneous Coordinates and Transformation Matrices, Homography and its Applications, Properties of Homography and Perspective Transformations. Camera Geometry: Camera Projection Models (Pinhole Camera Model), Intrinsic and Extrinsic Camera Parameters, Camera Calibration Techniques

2. Stereo Vision & Feature Extraction

Stereo Geometry - I: Fundamentals of Stereo Vision, Depth Perception in Human Vision, Epipolar Constraints and Fundamental Matrix, Correspondence Problem in Stereo Vision.

Stereo Geometry - II: Stereo Rectification and Disparity Estimation, Structure from Motion (SfM), 3D Reconstruction using Stereo Images, Applications in Robotics and AR/VR.

Feature Detection and Description: Key point Detection (Harris Corner, FAST), Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB). Feature Matching and Model Fitting: Feature Matching Techniques (Brute Force, FLANN), RANSAC Algorithm for Model Fitting, Image Registration and Transformation Estimation, Applications in Object Recognition.

3. High-Level Vision - Learning-Based Approaches

Clustering and Classification: Supervised vs. Unsupervised Learning in Computer Vision, K-Means and Mean-Shift Clustering, Support Vector Machines (SVM) for Image Classification, Random Forest and Decision Trees for Vision Tasks, Applications in Image Segmentation.

Dimensionality Reduction and Sparse Representation: Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), Manifold Learning (t-SNE and UMAP), Sparse Representations in Vision, Dictionary Learning and Applications.

4. Deep Learning for Computer Vision

Deep Neural Architecture and Applications: Introduction to Deep Learning in Computer Vision, Convolutional Neural Networks (CNN) Architectures, Transfer Learning and Pretrained Models (VGG, ResNet), Object Detection (YOLO, Faster R-CNN), Image Segmentation (U-Net, Mask R-CNN), Applications in Autonomous Systems and Healthcare.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand and apply fundamental image processing techniques, including filtering, enhancement, and edge detection.
CO2: Analyze and implement camera geometry, homography, and stereo vision concepts for depth estimation and 3D reconstruction.
CO3: Perform feature detection, matching, and model fitting for object recognition and image registration.
CO4: Utilize clustering, classification, and dimensionality reduction techniques for high-level vision tasks.

References

1. **Richard Szeliski** – *Computer Vision: Algorithms and Applications*, Springer, 2nd Edition, 2022.
2. **David A. Forsyth, Jean Ponce** – *Computer Vision: A Modern Approach*, Pearson, 2nd Edition, 2012.
3. **Gonzalez & Woods** – *Digital Image Processing*, Pearson, 4th Edition, 2018.

Cognitive and Affective Systems (EAIML117)

Credit: 3+0

Course Objective:

To understand the fundamentals of affective computing and its role in human-computer interaction. To explore emotion theories and their application in designing emotionally intelligent systems. To study various modalities for emotion recognition, including facial expressions, voice, text, and physiological signals.

Syllabus

1. Fundamentals of Affective Computing and Emotion Theory

Fundamentals of Affective Computing: Introduction to affective computing, historical background and significance, role of emotions in human-computer interaction, applications in AI, challenges and ethical considerations.

Emotion Theory and Emotional Design: Theories of emotion (Ekman's Basic Emotions, Russell's Circumplex Model, Scherer's Appraisal Theory), cognitive and physiological aspects of emotions, affective design principles in human-cantered AI, impact of emotions on decision-making and behaviour, affective computing in user experience design.

2. Experimental Design and Emotion Recognition in Modalities

Experimental Design: Affect Elicitation; Research and Development Tools: Methods for eliciting emotions (stimuli-based, context-based, and self-report methods), laboratory vs. real-world affective computing experiments, data collection and annotation techniques, research tools and datasets in affective computing, evaluation metrics and performance analysis.

Emotions in Facial Expressions: Facial Action Coding System (FACS), facial feature extraction techniques (geometric-based, appearance-based), deep learning for facial emotion recognition, datasets and benchmarks (FER, AffectNet, EmotioNet), applications in social robotics and virtual assistants.

3. Emotion Recognition in Voice and Text.

Emotions in Voice: Acoustic features for emotion recognition (pitch, intensity, spectral features), speech signal processing for affect detection, machine learning techniques for speech emotion classification, emotional speech synthesis and voice modulation, applications in conversational AI and healthcare.

Emotions in Text: Natural language processing (NLP) for sentiment and emotion analysis, lexicon-based vs. machine learning-based approaches, deep learning models for affective text analysis (RNNs, LSTMs, transformers), challenges in textual emotion recognition

4. Multimodal Emotion Recognition and Applications

Multimodal Emotion Recognition: Fusion techniques for combining multiple modalities (early fusion, late fusion, hybrid fusion), multimodal datasets and challenges, deep learning approaches for multimodal affective computing, real-time emotion recognition systems, applications in human-robot interaction, gaming, and personalized AI.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the fundamentals of affective computing, emotion theories, and their significance in AI and human-computer interaction.

CO2: Design and implement emotion recognition models based on facial expressions, voice, text, and physiological signals.

CO3: Apply experimental design techniques for affect elicitation, data collection, and evaluation in affective computing research.

CO4: Develop and analyze machine learning and deep learning models for unimodal and multimodal emotion recognition.

References

1. Rosalind W. Picard – Affective Computing, MIT Press, 2000.
2. Klaus R. Scherer, Tanja Bänziger, Etienne Roesch – Blueprint for Affective Computing: A Sourcebook and Manual, Oxford University Press, 2010.
3. Björn Schuller, Anton Batliner – Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing, Wiley, 2013.
4. Sidney D'Mello, Art Graesser, Björn Schuller, Jean-Claude Martin – Affective Computing and Intelligent Interaction: The 2021 Handbook, Springer, 2021.
5. James A. Russell, Lisa Feldman Barrett – The Psychology of Emotion in Restorative Practice, Oxford University Press, 2019.

Computational Intelligence
(EAIML118)

Credit: 3+0

Course Objectives:

The course goal is to make students familiar with basic principles of various computational methods of data processing that can commonly be called computational intelligence (CI). After the course the students will be able to conceptually understand the important terms and algorithms of CI, such that they would be able to choose appropriate method(s) for a given task.

Syllabus

1. **Introduction:** 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.

2. **Knowledge Representation and Reasoning:** 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. UNCERTAINTY: Non monotonic reasoning-Fuzzy Logic-Fuzzy rules-fuzzy inference-Temporal Logic-Temporal Reasoning-Neural Networks-Neuro-fuzzy Inference.

3. **Learning:** Probability basics - Bayes Rule and its Applications - Bayesian Networks – Exact and Approximate Inference in Bayesian Networks - Hidden Markov Models - Forms of Learning - Supervised Learning - Learning Decision Trees – Regression and Classification with Linear Models - Artificial NeuralNetworks – Nonparametric Models - Support Vector Machines - Statistical Learning - Learning with Complete Data - Learning with Hidden Variables- The EM Algorithm – Reinforcement Learning

4. **Intelligence and Applications:** Natural language processing-Morphological Analysis-Syntax Analysis-Semantic Analysis-All applications – Language Models - Information Retrieval – Information Extraction – Machine Translation – Machine Learning - Symbol-Based – Machine Learning: Connectionist – Machine Learning.

Course Outcome (CO): Upon completion of the course, the students will be able to

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 machine learning.

References

1. Stuart Russell, Peter Norvig, —Artificial Intelligence: A Modern Approach, Third Edition, Pearson Education / Prentice Hall of India, 2010.
2. Elaine Rich and Kevin Knight, —Artificial Intelligence, Third Edition, Tata McGraw-Hill, 2010.
3. Patrick H. Winston. "Artificial Intelligence", Third edition, Pearson Edition, 2006.
4. Dan W.Patterson, —Introduction to Artificial Intelligence and Expert Systems, PHI, 2006.
5. Nils J. Nilsson, —Artificial Intelligence: A new Synthesis, Harcourt Asia Pvt. Ltd., 2000.

High Performance Computing (EAIML119)

Credit: 3+0

Course Objective: The objective of this course is to teach participants the basic concepts of parallel computing and equip them with knowledge which will be sufficient enough for them to write parallel programs using industry-standard parallel programming frameworks.

Syllabus

1. **Overview of Grid Computing Technology& Open Grid Services Architecture:** History of Grid Computing, High Performance Computing, Cluster Computing. Peer-to-Peer Computing, Internet Computing, Grid Computing Model and Protocols, Types of Grids: Desktop Grids, Cluster Grids, Data Grids, High- Performance Grids, Applications and Architectures of High-Performance Grids, High Performance Application Development Environment. Open Grid Services Architecture**:** Introduction, Requirements, Capabilities, Security Considerations, GLOBUS Toolkit

2. **Overview of Cluster Computing:** Cluster Computer and its Architecture, Clusters Classifications, Components for Clusters, Cluster Middleware and SSI, Resource Management and Scheduling, Programming, Environments and Tools, Cluster Applications, Cluster Systems,

3. **Beowulf Cluster:** The Beowulf Model, Application Domains, Beowulf System Architecture, Software Practices, Parallel Programming with MPL, Parallel Virtual Machine (PVM).

4. **Overview of Cloud Computing:** Types of Cloud, Cyber infrastructure, Service Oriented Architecture Cloud Computing Components: Infrastructure, Storage, Platform, Application, Services, Clients, Cloud Computing Architecture.

Course Outcome (CO) : At the end of course, the student will be able

CO1: To understand the basic concept of Computer architecture and Modern Processor.
CO2: To understand the basic concepts of access optimization and parallel computers.
CO3: To describe different parallel processing platforms involved in achieving high performance computing.
CO4: Develop efficient and high-performance parallel programming.

References

1. Laurence T.Yang, Minyi Guo – High Performance Computing Paradigm and Infrastructure John Wiley
2. Ahmar Abbas, “Grid Computing: Practical Guide to Technology & Applications”, Firewall Media, 2004.
3. Joshy Joseph and Craig Fellenstein , “Grid Computing” Pearson Education, 2004.
4. lan Foster, et al.,“The Open Grid Services Architecture”, Version 1.5 (GFD.80). Open Grid Forum, 2006.

Parallel and Distributed Computing (EAIML120)

Credit: 3+0

Course Objective: To learn the advanced concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students.

Syllabus

1. **Parallelism Fundamentals and Parallel Architecture:** Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness. **Parallel Architecture:** Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, coprocessing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

2. **Parallel Decomposition and Parallel Performance:** Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost-Performance trade-off;

3. **Distributed Computing:** Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges, A Model of Distributed Computations, A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- Passing, Consensus, Conditional Actions, Critical Paths, Scalability, cache coherence in multiprocessor systems, synchronization mechanism. CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

4. **Parallel Algorithms design, Analysis, and Programming:** Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms.

Course outcomes (CO): On completion of this course, the students will be able to

CO1: Apply the fundamentals of parallel and distributed computing including parallel architectures
CO2: Apply parallel algorithms and key technologies.
CO3: Develop and execute basic parallel and distributed applications using basic programming models.
CO4: Analyze the performance issues in parallel computing and trade-offs.

References

1. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley (2008).
2. A Grama, A Gupra, G Karypis, V Kumar. Introduction to Parallel Computing, Addison Wesley (2003).

3. B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing with Opencl. Morgan Kaufmann and Elsevier (2011).
4. T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley (2004).
5. Quinn, M. J.,Parallel Programming in C with MPI and OpenMP, McGraw-Hill (2004)

Data-Driven Reliability Analysis with AI and ML (EAIML121)

Credit: 3+0

Course Objective:

To introduce fundamental reliability engineering concepts and statistical models. To explore the application of AI and ML techniques for failure prediction and reliability analysis. To apply supervised and unsupervised learning methods to reliability modelling.

Syllabus

1. Fundamentals of Reliability Analysis

Introduction to Reliability Engineering: Concepts of reliability, maintainability, and availability, reliability life cycle, failure modes, types of failures, and failure data analysis. Statistical Foundations for Reliability Analysis: Probability distributions (exponential, Weibull, normal, log-normal), survival analysis, hazard function, mean time to failure (MTTF), and mean time between failures (MTBF).

2. Machine Learning for Reliability Modelling

Supervised and Unsupervised Learning for Reliability: Regression models for reliability prediction, classification techniques for fault detection, clustering for anomaly detection, feature selection, and engineering. Predictive Maintenance and Failure Prognostics: Time-series forecasting, degradation modelling, predictive maintenance using AI and ML, real-time fault detection, and diagnostics.

3. AI and Deep Learning in Reliability Analysis

Deep Learning for Failure Prediction: Neural networks for reliability assessment, convolutional neural networks (CNNs) for defect detection, recurrent neural networks (RNNs) for time-series analysis, reinforcement learning in predictive maintenance. Natural Language Processing (NLP) for Reliability: AI-driven analysis of maintenance logs, text mining for failure pattern identification, automated root cause analysis using NLP.

4. Advanced Topics in AI-Driven Reliability Engineering

Big Data and IoT for Reliability Analysis: Role of big data in reliability engineering, IoT-based predictive maintenance, real-time condition monitoring, cloud computing in reliability analysis.

Ethical and Safety Considerations in AI-Driven Reliability: Challenges in AI-based reliability prediction, bias in ML models, explainability and interpretability in reliability AI systems, regulatory and ethical implications.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the principles of reliability engineering and statistical failure analysis.
CO2: Apply machine learning models for predictive maintenance and failure detection.
CO3: Implement deep learning techniques for defect recognition and reliability prediction.
CO4: Utilize IoT and big data analytics for real-time reliability assessment.
CO5: Evaluate ethical and safety considerations in AI-based reliability analysis.

References

1. **E. E. Lewis** – *Introduction to Reliability Engineering*, Wiley, 1996.
2. **Wei Lei, Hamid Reza Karimi, P. W. Tse** – *Data-Driven Prognostics and Health Management: Theory, Methods, and Applications*, Springer, 2021.
3. **Andrew Ng, Kian Katanforoosh, Younes Bensouda Mourri** – *Machine Learning Yearning*, DeepLearning.AI, 2018.
4. **Mohamed Medhat Gaber** – *Data Stream Processing in Ubiquitous Computing: Data Management and Mining*, Springer, 2010.
5. **Ben S. Blanchard, Dinesh C. Verma, Elmer L. Peterson** – *Maintainability: A Key to Effective Serviceability and Maintenance Management*, Wiley, 1995.

Applied Machine Learning for Engineering and Science (EAIML122)

Credit: 3+0

Course Objective:

To introduce fundamental machine learning concepts and techniques relevant to engineering and scientific applications. To develop an understanding of supervised and unsupervised learning methods for solving real-world engineering problems.

Syllabus

1. Fundamentals of Machine Learning for Engineering and Science

Introduction to Machine Learning: Basics of supervised, unsupervised, and reinforcement learning, key ML concepts, types of data, role of ML in engineering and scientific applications. Statistical Foundations for ML: Probability theory, linear algebra, optimization techniques, Bayesian inference, and their applications in engineering problems.

2. Supervised and Unsupervised Learning for Engineering Applications

Regression and Classification Techniques: Linear regression, logistic regression, decision trees, support vector machines (SVM), ensemble learning, applications in scientific modelling. Clustering and Dimensionality Reduction: K-means clustering, hierarchical clustering, principal component analysis (PCA), t-SNE, applications in signal processing and material science.

3. Deep Learning and Advanced ML Methods

Neural Networks and Deep Learning: Basics of artificial neural networks (ANNs), convolutional neural networks (CNNs) for image-based scientific applications, recurrent neural networks (RNNs) for time-series analysis in engineering. Reinforcement Learning and Optimization: Reinforcement learning concepts, applications in control systems, robotics, and scientific simulations.

4. Real-World Applications and Ethical Considerations

Machine Learning for Engineering Domains: ML applications in mechanical, electrical, civil, biomedical, and chemical engineering, case studies on predictive maintenance, computational fluid dynamics (CFD), and materials discovery. Ethical Considerations in ML for Science and Engineering: Bias in scientific data, interpretability of ML models, explainable AI (XAI), regulatory aspects in engineering AI applications.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Identify engineering and scientific problems solvable with ML.
CO2: Develop ML models for prediction, optimization, and automation.
CO3: Apply deep learning for complex scientific and engineering tasks.
CO4: Enhance decision-making and fault detection using ML techniques.
CO5: Evaluate the reliability and ethics of AI-driven solutions.

References

1. Christopher M. Bishop – Pattern Recognition and Machine Learning, Springer, 2006.
2. Tom M. Mitchell – Machine Learning, McGraw-Hill, 1997.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, MIT Press, 2016.
4. Kevin P. Murphy – Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
5. Andrew Ng – Machine Learning Yearning, DeepLearning.AI, 2018.
6. Trevor Hastie, Robert Tibshirani, Jerome Friedman – The Elements of Statistical Learning, Springer, 2009.

Pervasive Computing (EAIML123)

Credit: 3+0

Course Objective: This course provides a fundamental theoretical concept in Pervasive Computing and its applications in various areas.

Syllabus

1. **Pervasive Computing Concepts:** Perspectives of Pervasive Computing, Challenges, Technology; The Structure and Elements of Pervasive Computing Systems: Infrastructure and Devices, Middleware for Pervasive Computing Systems, Pervasive Computing Environments

2. **Context Collection, User Tracking, and Context Reasoning:** Resource Management in Pervasive Computing: Efficient Resource Allocation in Pervasive Environments, Transparent Task Migration, Implementation and Illustrations.

3. **HCI Interface in Pervasive Environments:** HCI Service and Interaction Migration, Context-Driven HCI Service Selection, Scenario Study: Video Calls at a Smart Office, A Web Service–Based HCI Migration Framework.

4. **Pervasive Mobile Transactions:** Mobile Transaction Framework, Context-Aware Pervasive Transaction Model, Dynamic Transaction Management, Formal Transaction Verification, Evaluations

Course Outcome (CO): At the end of course, the student will be able to

CO1: Understand the fundamental theoretical concepts in pervasive computing.
CO2: Understand the aspects of context awareness.
CO3: Study the methods for efficient resource allocation and task migration.

References

1. Minyi Guo, Jingyu Zhou, Feilong Tang, Yao Shen,” Pervasive Computing: Concepts, Technologies and Applications”,CRC Press, 2016.
2. Obaidat, Mohammad S., Mieso Denko, and Isaac Woungang, eds. Pervasive computing and networking. John Wiley & Sons, 2011.
3. Laurence T. Yang, Handbook on Mobile and Ubiquitous Computing Status and Perspective, 2012, CRC Press.

Generative Models (EAIML124)

Credit: 3+0

Course Objective:

Introduce the fundamental concepts and principles of generative models. Explore different generative approaches, including probabilistic, adversarial, and autoregressive models. Provide hands-on experience with training generative models for real-world applications.

Syllabus

1. Introduction to Generative Models: Overview of generative modelling, discriminative vs. generative models, probabilistic graphical models, Bayesian networks, Markov models, and hidden Markov models (HMM). Basics of probability distributions in generative modelling, latent variable models, and maximum likelihood estimation (MLE).

2. Variational and Bayesian Methods: Bayesian inference, variational autoencoders (VAE), evidence lower bound (ELBO), variational inference, Gaussian mixture models (GMM), expectation-maximization (EM) algorithm, and normalizing flows. Applications of VAEs in data synthesis, image generation, and semi-supervised learning.

3. Generative Adversarial Networks (GANs): Introduction to GANs, architecture and components (generator and discriminator), min-max optimization, training challenges (mode collapse, instability), Wasserstein GAN (WGAN), Conditional GAN (cGAN), StyleGAN, and CycleGAN. Applications of GANs in image synthesis, data augmentation, and domain adaptation.

4. Autoregressive and Diffusion-Based Models: Autoregressive models: PixelCNN, PixelRNN, WaveNet, Transformer-based generative models (GPT, BERT for generation). Diffusion models: Denoising diffusion probabilistic models (DDPM), score-based generative models, stability and scalability of diffusion models. Applications in text, image, and speech generation.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand various generative modeling techniques and their mathematical foundations.
CO2: Implement probabilistic and variational methods for generative tasks.
CO3: Develop and train GANs, VAEs, and autoregressive models for data generation.
CO4: Apply generative models in creative AI, healthcare, and industry applications.

References

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville – *Deep Learning*, MIT Press, 2016.
2. David Foster – *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, O'Reilly, 2019.
3. Sebastian Raschka, Vahid Mirjalili – *Python Machine Learning*, Packt Publishing, 2020.
4. Diederik P. Kingma, Max Welling – *Auto-Encoding Variational Bayes (VAE)*, ArXiv, 2013.
5. Alec Radford et al. – *Generative Adversarial Networks (GANs)*, ArXiv, 2014.
6. Jonathan Ho et al. – *Denoising Diffusion Probabilistic Models (DDPM)*, ArXiv, 2020

Randomized Algorithms (EAIML125)

Credit: 3+0

Course Objective: Randomized Algorithms are the state of the art in contemporary algorithm design. They are usually simple, sometimes even easy to analyze, and they work well in practice. This course provides an introduction to basic concepts in the design and analysis of randomized algorithms.

Syllabus

1. **Randomized Data Structures:** Random Treaps, Skip Lists, Hash Tables, Universal Family of Hash Functions, Perfect Hashing.

2. **Randomized Computational Geometry:** Illustrations of randomized incremental algorithms like randomized convex hull construction, geometric duality, half space intersections, Delaunay Triangulation, trapezoidal decomposition; illustrations of random sampling like point location in arrangements, and linear programming.

3. **Online Algorithms & Distributed Algorithms:** Adversary models, online paging against oblivious and adaptive adversaries, Yao's minimax principle, lower bound for online paging against an oblivious adversary, the k-server problem. **Distributed Algorithms:** Symmetry breaking problems like leader election, Byzantine agreement, maximal independent set, and colouring; algorithms for dynamic networks; the k-machine model for processing large graphs.

4. **Streaming Algorithms:** the streaming model, approximate counting, reservoir sampling, AMS sketching. **Property testing algorithms:** the property testing model, testing whether a graph is connected, bipartite (enforce and test paradigm), and triangle free (using Szemeredi's regularity lemma).

Course Outcomes (CO): At the end of this course, students will be able to

CO1: Design and analyze algorithms for basic numerical problems.
CO2: Use randomization in the design and analysis of efficient algorithms.
CO3: Efficiently solve optimization problems using approximation algorithms.
CO4: Identify problems that are unlikely to admit efficient algorithms and argue for their difficulty via computational complexity theory.

References

1. Randomized Algorithms, by Motwani and Raghavan, Cambridge University Press, 1995.

2. Probability and Computing: Randomized Algorithms and Probabilistic Analysis, by Mitzenmacher and Upfal, Cambridge University Press, 2nd edition, 2017.

Syllabus of Professional Elective Courses

Elective-II

(Offered by Department)

Computational Science and High-Performance Computing (EAIML214)

Credit: 3+0

Course Objective: Introduce fundamental concepts of computational science and HPC. Explore parallel and distributed computing techniques for scientific applications. Implement efficient numerical methods for large-scale simulations.

Syllabus

1. Fundamentals of Computational Science and HPC
Introduction to computational science, importance in scientific research and engineering, basics of high-performance computing (HPC), parallel computing architectures, computational complexity, and performance metrics.

2. Parallel and Distributed Computing
Parallel programming models (shared memory, distributed memory, hybrid), introduction to MPI and OpenMP, CUDA programming for GPU acceleration, load balancing, and memory hierarchy optimization.

3. Numerical Methods and Scientific Simulations
Efficient numerical algorithms for scientific computing, solving large-scale linear and nonlinear equations, iterative solvers, finite difference and finite element methods, applications in physics, chemistry.

4. Advanced Topics and Real-World Applications
Performance optimization techniques, scalability challenges in HPC, cloud-based HPC solutions, real-world applications in climate modeling, bioinformatics, computational fluid dynamics (CFD), materials science, and AI-driven HPC.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the fundamentals of computational science and high-performance computing.
CO2: Develop parallel and distributed computing solutions using MPI, OpenMP, and CUDA.
CO3: Implement numerical algorithms for large-scale scientific simulations.
CO4: Optimize computational performance and scalability in HPC applications.

References

1. Ian Foster – *Designing and Building Parallel Programs*, Addison-Wesley, 1995.
2. Peter Pacheco – *Parallel Programming with MPI*, Morgan Kaufmann, 1996.
3. Georg Hager, Gerhard Wellein – *Introduction to High-Performance Computing for Scientists and Engineers*, CRC Press, 2010.
4. David Kirk, Wen-mei Hwu – *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, 2016.
5. William Gropp, Ewing Lusk, Anthony Skjellum – *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, 2014.

Confidential Computing with AI and ML (EAIML215)

Credit: 3+0

Course Objective: Introduce concepts of confidential computing and secure AI/ML techniques. Explore privacy-preserving machine learning approaches.

Syllabus

1. Introduction to Confidential Computing and AI Security
Overview of confidential computing, need for secure AI and ML, threat models in AI systems, trusted execution environments (TEEs), homomorphic encryption, differential privacy, and secure enclaves.

2. Privacy-Preserving Machine Learning Techniques
Federated learning, secure multi-party computation (SMPC), homomorphic encryption for AI, differential privacy in ML models, adversarial robustness, and privacy-aware AI model training.

3. Secure Data Processing and Model Deployment
Confidential data processing in AI pipelines, secure cloud-based AI computation, encrypted model inference, access control and identity management in AI-driven applications, and ethical considerations in AI security.

4. Applications and Advanced Topics
Secure AI in healthcare, finance, and autonomous systems, blockchain and AI for secure data sharing, regulatory compliance (GDPR, HIPAA) in AI, zero-trust architectures, and future trends in AI security.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the principles of confidential computing and AI security.
CO2: Implement privacy-preserving techniques in machine learning.
CO3: Develop secure AI pipelines using encryption and federated learning.
CO4: Apply confidential computing techniques in real-world AI applications.

References

1. Shafi Goldwasser, Vinod Vaikuntanathan – *Cryptographic Primitives for AI and Privacy-Preserving Machine Learning*, MIT Press, 2020.
2. Florian Tramèr, Dan Boneh – *Privacy-Preserving Machine Learning: Theory and Practice*, Springer, 2022.
3. Sebastien Andreux et al. – *Federated Learning for AI Privacy and Security*, Cambridge University Press, 2021.
4. Gennaro Costagliola, Antonio De Luca – *Secure AI: Methods and Applications*, Elsevier, 2023.
5. Chris Peikert – *Homomorphic Encryption and Secure AI*, CRC Press, 2021.

Service Oriented Architecture (EAIML216)

Credit: 3+0

Course Objective: To introduce the concepts and design principles of SOA, Non-technical aspects such as governance, impact on culture and organization, as well as the various interoperability standards, technology infrastructure and security considerations associated with SOA implementations.

Syllabus

1. **Introduction:** SOA and MSA Basics: Service Orientation in Daily Life, Evolution of SOA and MSA. Service oriented Architecture and Microservices architecture – Drivers for SOA, Dimensions of SOA, Conceptual Model of SOA, Standards and Guidelines for SOA, Emergence of MSA. Enterprise-Wide SOA: Considerations for Enterprise-wide SOA, Strawman Architecture for Enterprise-wide SOA, Enterprise SOA Reference Architecture, Object-oriented Analysis and Design (OOAD) Process, Service-oriented Analysis and Design (SOAD) Process, SOA Methodology for Enterprise.

2. **Service-Oriented Applications:** Considerations for Service-oriented Applications, Patterns for SOA, Pattern-based Architecture for Service-oriented Applications, Composite Applications, Composite Application Programming Model. Service-Oriented Analysis and Design: Need for Models, Principles of Service Design, Nonfunctional Properties for Services, Design of Activity Services (or Business Services), Design of Data Services, Design of Client Services, Design of Business Process Services.

3. **Technologies for SOA:** Technologies for Service Enablement, Technologies for Service Integration, Technologies for Service Orchestration. SOA Governance and Implementation: Strategic Architecture Governance, Service Design-time Governance, Service Run-time Governance, Approach for Enterprise-wide SOA Implementation.

4. **Big Data and SOA with Best Practices:** Concepts, Big Data and its characteristics, Technologies for Big Data, Service-orientation for Big Data Solutions. Business Case for SOA: Stakeholder Objectives, Benefits of SOA, Cost Savings, Return on Investment (ROI), Build a Case for SOA.**SOA** Best Practices**:** SOA Strategy – Best Practices, SOA Development – Best Practices, SOA Governance – Best Practices. EA and SOA for Business and IT Alignment: Enterprise Architecture, Need for Business and It Alignment, EA and SOA for Business and It Alignment.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Comprehend the need for SOA and its systematic evolution.
CO2: Apply SOA technologies to enterprise domain.
CO3: Design and analyze various SOA patterns and techniques.
CO4: Compare and evaluate best strategies and practices of SOA.

References

1. Shankar Kambhampaty; Service **-** Oriented Architecture & Microservices Architecture: For Enterprise, Cloud, Big Data and Mobile; Wiley; 3rd Edition; 2018; ISBN: 9788126564064.
2. Icon Group International; The 2018-2023 World Outlook for Service-Oriented Architecture (SOA) Software and Services; ICON Group International; 1st Edition, 2017; ASIN: B06WGPN8YD.
3. Thomas Erl; Service Oriented Architecture Concepts Technology & Design; Pearson Education Limited; 2015; ISBN-13: 9788131714904.
4. Guido Schmutz, Peter Welkenbach, Daniel Liebhart; Service Oriented Architecture: An Integration Blueprint; Shroff Publishers & Distributors; 2010; ISBN-13: 9789350231081.

Introduction to Internet of Things (EAIML217)

Credit: 3+0

Course Objective: Introduce the fundamental concepts, architecture, and applications of IoT. Explain networking, communication protocols, and sensor networks in IoT. Provide hands-on experience with IoT hardware like Arduino and Raspberry Pi.

Syllabus

1. Fundamentals of IoT and Networking: Introduction to IoT: concepts, scope, applications, sensing and actuation, basics of networking, communication models, and IoT architecture. Basics of networking (wired and wireless), OSI and TCP/IP models, and key communication protocols used in IoT.

2. Communication Protocols and IoT Hardware: IoT communication protocols (MQTT, CoAP, AMQP, Bluetooth, Zigbee, LoRa, etc.), sensor networks and their architecture, machine-to-machine (M2M) communications, interoperability challenges in IoT. Introduction to Arduino programming, integrating sensors and actuators with Arduino, basics of Python programming, and introduction to Raspberry Pi.

3. IoT Implementation and Data Processing: IoT implementation with Raspberry Pi, Software-Defined Networking (SDN) for IoT applications, data handling and analytics, introduction to cloud computing for IoT, edge and fog computing concepts, and the role of Sensor-Cloud in IoT infrastructure.

4. Advanced IoT Applications and Case Studies: IoT applications in smart cities, smart homes, connected vehicles, smart grids, and industrial IoT. Security, privacy, and scalability challenges in IoT systems. Case studies on IoT applications in agriculture, healthcare, and activity monitoring.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand IoT architecture, networking, and communication models.
CO2: Implement IoT systems using Arduino, Raspberry Pi, and sensor integration.
CO3: Work with IoT communication protocols and data analytics.
CO4: Apply cloud, fog, and edge computing in IoT solutions.

References

1. Arshdeep Bahga, Vijay Madisetti – Internet of Things: A Hands-On Approach, Universities Press, 2015.
2. Olivier Hersent, David Boswarthick, Omar Elloumi – The Internet of Things: Key Applications and Protocols, Wiley, 2012.
3. Adrian McEwen, Hakim Cassimally – Designing the Internet of Things, Wiley, 2013.
4. Benson Hougland – IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, 2017.
5. Donald Norris – Programming the Internet of Things (IoT) with Python, McGraw-Hill, 2016.
6. Pethuru Raj, Anupama C. Raman – The Internet of Things: Enabling Technologies, Platforms

Quantum Computing (EAIML218)

Credit: 3+0

Course Objective: This course focuses on the fundamentals of Quantum Computing and its application in information technology.

Syllabus

1. **Fundamental Concepts & Quantum Computation:** Global Perspectives, Quantum Bits, Quantum Computation, Quantum Algorithms, Quantum Information, Postulates of Quantum Mechanisms. Quantum Computation**:** Quantum Circuits – Quantum algorithms, Single Orbit operations, Control Operations, Measurement, Universal Quantum Gates, Simulation of Quantum Systems, Quantum Fourier transform, Phase estimation, Applications, Quantum search algorithms – Quantum counting – Speeding up the solution of NP – complete problems – Quantum Search for an unstructured database.

2. **Quantum Computers:** Guiding Principles, Conditions for Quantum Computation, Harmonic Oscillator Quantum Computer, Optical Photon Quantum Computer – Optical cavity Quantum electrodynamics, Ion traps, Nuclear Magnetic resonance.

3. **Quantum Information:** Quantum noise and Quantum Operations – Classical Noise and Markov Processes, Quantum Operations, Examples of Quantum noise and Quantum Operations – Applications of Quantum operations, Limitations of the Quantum operations formalism, Distance Measures for Quantum information.

4. **Quantum Error Correction:** Introduction, Shor code, Theory of Quantum Error –Correction, Constructing Quantum Codes, Stabilizer codes, Fault – Tolerant Quantum Computation, Entropy and information – Shannon Entropy, Basic properties of Entropy, Von Neumann, Strong Sub Additivity, Data Compression, Entanglement as a physical resource.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Distinguish problems of different computational complexity and explain why certain problems are rendered tractable by quantum computation with reference to the relevant concepts in quantum theory.
CO2: Demonstrate an understanding of a quantum computing algorithm by simulating it on a classical computer, and state some of the practical challenges in building a quantum computer.
CO3: Contribute to a medium-scale application program as part of a co-operative team, making use of appropriate collaborative development tools (such as version control systems).
CO4: Produce code and documentation that is comprehensible to a group of different programmers and present the theoretical background and results of a project in written and verbal form.

References

1. Micheal A. Nielsen. &Issac L. Chiang, "Quantum Computation and Quantum Information", Cambridge University Press, Fint South Asian edition, 2002.
2. Eleanor G. Rieffel, Wolfgang H. Polak , "Quantum Computing - A Gentle Introduction" (Scientific and Engineering Computation) Paperback – Import, Oct 2014
3. Computing since Democritus by Scott Aaronson, Computer Science: An Introduction by N. DavidMermin

Distributed Algorithms (EAIML219)

Credit: 3+0

Course Objective: The course provides students with the foundation knowledge to understand, analysis and design distributed algorithms. This shall be useful to a wide variety of research topics from the theory of distributed algorithms to protocol design, e.g., broadcasting protocols for discovery purposes in ad-hoc networks.

Syllabus

1. **Review of Prerequisite Topics:** Graph theory, probability theory covering Markov's inequality, Chebyshev's inequality, Chernoff bounds, Markov chains and random walks.

2. **Models for Distributed Computer Networks:** Message passing and shared memory models, synchronous and asynchronous timing models, failure models. Complexity measures like time, space, and message complexity.

3. **Fundamental Problems on Distributed Networks:** Maximal independent set, minimum spanning tree, vertex colouring, dominating set, routing algorithms, leader election, Byzantine agreement, synchronizers, graph spanners, dynamic networks.

4. **Application Specific Problems:** Storage and retrieval of data in peer-to-peer computing, coverage and routing in sensor networks, and rumour spreading in social networking.

Course Outcome (CO): At the end of course, the student will be able to
CO1: Demonstrate basic concepts and principles Distributed Algorithm.
CO2: Apply Distributed Algorithm in Computer Networks.
CO3: Demonstrate challenges in Distributed Algorithm.
CO4: Apply Distributed Algorithms in storage and retrieval of information.

References

1. Distributed Computing: a Locality-Sensitive Approach, by David Peleg.
2. Distributed Algorithms, by Nancy Lynch.
3. Distributed Computing: Fundamentals, Simulations, and Advanced Topics, by Hagit Attiya and Jennifer Welch.
4. Randomized Algorithms, by Rajeev Motwani and Prabhakar Raghavan.
5. Principles of Distributed Computing, lecture notes by Roger Wattenhofer.

Blockchain Architecture Design (EAIML220)

Credit: 3+0

Course Objective: This course is an introduction to Blockchain Architecture and its application in finance and governance.

Syllabus

1. **Introduction to Blockchain & Consensus:** Digital Money to Distributed Ledgers, Design Primitives: Protocols, Security, Consensus, Permissions, Privacy. Blockchain Architecture and Design: Basic crypto primitives: Hash, Signature,) Hash chain to Blockchain, Basic consensus mechanisms. **Consensus:** Requirements for the consensus protocols, Proof of Work (PoW), Scalability aspects of Blockchain consensus protocols Permissioned Block chains: Design goals, Consensus protocols for Permissioned Blockchains.

2. **Hyperledger Fabric (A):** Decomposing the consensus process, Hyperledger fabric components, Chain code Design and Implementation Hyperledger Fabric (B): Beyond Chain code: fabric SDK and Front End (b) Hyperledger composer tool.

3. **Blockchain in Financial Software and Systems (FSS):** (i) Settlements, (ii) KYC, (iii) Capital markets, (iv) Insurance. Blockchain in Trade/Supply Chain: (i) Provenance of goods, visibility, trade/supply chain finance, invoice management discounting, etc.

4. **Blockchain for Government:** (i) Digital identity, land records and other kinds of record keeping between government entities, (ii) public distribution system social welfare systems Blockchain Cryptography, Privacy and Security on Blockchain.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Describe the basic understanding of Blockchain architecture along with its primitive.
CO2: Explain the requirements for basic protocol along with scalability aspects.
CO3: Design and deploy the consensus process using frontend and backend.
CO4: Apply Blockchain techniques for different use cases like Finance, Trade/Supply and Govern ment activities.

References

1. Mstering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos
2. Blockchain by Melanie Swa, O'Reilly
3. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric
4. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits - https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html.

Big Data Analytics using Python (EAIML221)

Credit: 3+0

Course Objective: Introduce fundamental concepts of big data and its significance in modern applications. Equip students with Python-based tools for handling, analysing, and visualizing large datasets. Explore big data processing frameworks like Hadoop and Spark for scalable computations.

Syllabus

1. Introduction to Big Data and Python for Analytics: Definition and characteristics of big data (volume, velocity, variety, veracity), big data ecosystem, challenges, and opportunities. Introduction to Python for data analytics, NumPy, Pandas, and Matplotlib for data manipulation and visualization.

2. Big Data Processing Frameworks and Storage: Hadoop ecosystem (HDFS, MapReduce, YARN), Apache Spark for big data processing, Spark RDDs, DataFrames, and MLlib. NoSQL databases (MongoDB, Cassandra) and their integration with Python programming languages etc .

3. Data Preprocessing, Machine Learning, and Streaming Analytics: Data cleaning, transformation, and feature engineering. Scalable machine learning using Spark MLlib and Scikit-Learn. Real-time data analytics with Apache Kafka and Spark Streaming.

4. Big Data Applications and Case Studies: Big data applications in healthcare, finance, social media, and e-commerce. Ethical considerations, privacy concerns, and regulatory compliance in big data. Hands-on projects and case studies on large-scale data analytics using Python.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand big data concepts, storage systems, and processing frameworks.
CO2: Use Python libraries for big data analytics and visualization.
CO3: Implement machine learning models on large-scale datasets.
CO4: Work with real-time data processing frameworks like Spark Streaming and Kafka.

References

1. Joel Grus – *Data Science from Scratch: First Principles with Python*, O'Reilly, 2019.
2. Frank Kane – *Hands-On Big Data Analytics with Spark*, Packt Publishing, 2017.
3. Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee – *Learning Spark: Lightning-Fast Data Analytics*, O'Reilly, 2020.
4. Benjamin Bengfort, Jenny Kim – *Data Analytics with Hadoop: An Introduction for Data Scientists*, O'Reilly, 2016.
5. Bill Chambers, Matei Zaharia – *Spark: The Definitive Guide: Big Data Processing Made Simple*, O'Reilly, 2018.

Machine Learning for Computational Biology (EAIML222)

Credit: 3+0

Course Objective: Provide an introduction to computational biology and its intersection with machine learning. Develop an understanding of sequence analysis techniques using ML methods. Explore ML applications in structural and functional genomics.

Syllabus

1. Foundations of Computational Biology and Machine Learning: Introduction to computational biology, biological sequences, structures, and systems. Overview of machine learning in biology, supervised and unsupervised learning, probability and statistical models, and essential Python libraries for bioinformatics**.**

2. Sequence Analysis and Machine Learning Models: DNA, RNA, and protein sequence analysis, sequence alignment (global and local), Hidden Markov Models (HMMs), probabilistic graphical models, neural networks for sequence prediction, and motif discovery using ML techniques.

3. Structural and Functional Genomics: Machine learning approaches for protein structure prediction, molecular docking, and functional genomics. Feature extraction from biological data, clustering methods for gene expression analysis, and deep learning applications in genomics.

4. Systems Biology and Advanced Applications: Network-based models for biological systems, graph neural networks in bioinformatics, machine learning for drug discovery, personalized medicine, and AI-driven healthcare solutions. Case studies on disease modelling, computational neuroscience.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Understand the fundamentals of computational biology and machine learning.
CO2: Apply machine learning models to analyze biological sequences.
CO3: Utilize ML techniques for structural and functional genomics.
CO4: Implement systems biology models using AI and ML approaches.

References

1. Pierre Baldi, Søren Brunak – *Bioinformatics: The Machine Learning Approach*, MIT Press, 2001.
2. Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly, 2019.
3. Sebastian Raschka, Vahid Mirjalili – *Python Machine Learning*, Packt Publishing, 2020.
4. Andreas D. Baxevanis, B. F. Francis Ouellette – *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, Wiley, 2005.
5. Barton Poulson – *Machine Learning for Data Science Handbook*, O'Reilly, 2021.

Software Defined Network (EAIML223)

Credit: 3+0

Course Objective: This course focuses on fundamentals of Software Defined Networks and its application.

Syllabus

1. **Introduction:** History of Software Defined Networking (SDN) – Modern Data Center – Traditional Switch Architecture – Why SDN – Evolution of SDN – How SDN Works – Centralized and Distributed Control and Date Planes

2. **Open Flow &SDN Controllers:** Open Flow Specification – Drawbacks of Open SDN, SDN via APIs, SDN via Hypervisor- Based Overlays – SDN via Opening up the Device – SDN Controllers – General Concepts. **Data Centers** Multitenant and Virtualized Multitenant Data Center – SDN Solutions for the Data Center Network – VLANs – EVPN – VxLAN – NVGRE

3. **SDN Programming Programming SDNs:** Northbound Application Programming Interface, Current Languages and Tools, Composition of SDNs – Network Functions Virtualization (NFV) and Software Defined Networks: Concepts, Implementation and Applications

4. **SDN :**Juniper SDN Framework – IETF SDN Framework – Open Daylight Controller – Floodlight Controller – Bandwidth Calendaring – Data Center Orchestration

Course Outcome (CO): Upon completion of the course, the students will be able to

CO1: Analyze the evolution of software defined networks.
CO2: Express the various components of SDN and their usage.
CO3: Explain the use of SDN in the current networking scenario.
CO4: Design and develop various applications of SDN.

References

1. Paul Goransson and Chuck Black, —Software Defined Networks: A Comprehensive Approach, First Edition, Morgan Kaufmann, 2014.
2. Thomas D. Nadeau, Ken Gray, —SDN: Software Defined Networks, O'Reilly Med

Open Elective Courses

(Offered by Department)

Machine Learning for Engineering and Science Applications

(AIMLOE01)

Credit: 3+0

Course Objective: To provide the knowledge of Machine Learning for Engineering and science applications and its implementations in context of understanding the applications of ML.

1. Mathematical Basics: Introduction to Machine Learning, Linear Algebra Mathematical Basics Probability Computational Basics – Numerical computation and optimization, Introduction to Machine learning packages.
2. Linear and Logistic Regression – Bias/Variance Tradeoff, Regularization, Variants of Gradient Descent, MLE, MAP, Application Neural Networks – Multilayer Perceptron, Backpropagation, Applications Convolutional Neural Networks CNN Operations, CNN architectures Convolutional Neural Networks Training, Transfer Learning, Applications.
3. Recurrent Neural Networks RNN, LSTM, GRU, Applications Classical Techniques Bayesian Regression, Binary Trees, Random Forests, SVM, Naïve Bayes, Applications Classical Techniques k-Means, kNN, GMM, Expectation Maximization, Applications.
4. Advanced Techniques: Structured Probabilistic Models, Monte Carlo Methods, Advanced Techniques Autoencoders, Generative Adversarial Network.

References

1. Deep Learning, Goodfellow et al, MIT Press, 20172.
2. Pattern Recognition and Machine Learning, Christopher Bishop, Springer, 20093.
3. References to research papers will be provided through the course.

Machine Learning For Soil and Crop Management

(AIMLOE02)

Credit: 3+0

Course Objective: To provide the knowledge of Machine Learning for Engineering and science applications and its implementations in context of understanding the applications of ML.

1. General Overview of Ml and Dl Applications in Agriculture Basics of Multivariate Data Analytics Principal Component Analysis and Regression Applications in Agriculture Applications of Classification and Clustering Methods in Agriculture.

2. Diffuse Reflectance Spectroscopy: Basics and Applications for Crop and Soil Use Of Ml for Portable, Proximal Soil and Crop Sensors Ml and Dl for Soil and Crop Image Processing

3. Uav and Ml Applications in Agriculture Hyperspectral Remote Sensing and Ml Applications In Agriculture Digital Soil Mapping – General Overview.

4. Digital Soil Mapping With Continuous Variables Digital Soil Mapping With Categorical Variables

References

1. Introduction to Multivariate Statistical Analysis in Chemometrics by Kurt Varmuza and Peter Filzmoser.
2. Using R for Digital Soil Mapping by Malone, Minasny, and McBratney

Artificial Intelligence: Knowledge Representation and Reasoning

(AIMLOE03)

Credit: 3+0

Course Objective: To provide the knowledge of Machine Learning for Engineering and science applications and its implementations in context of understanding the applications of ML.

1. Introduction: History and Philosophy. Symbolic Reasoning. Truth, Logic, and Provability. Propositional Logic. Direct Proofs. The Tableau Method.

2. First Order Logic. Universal Instantiation. The Unification Algorithm. Forward and Backward Chaining. The Resolution Refutation Method. Horn Clauses and Logic Programming. Prolog.

3. Rule Based Systems. The OPS5 Language. The Rete Algorithm. Representation in First Order Logic. Conceptual Dependency. Frames. Description Logics and the Web Ontology Language.

4. **Taxonomies and Inheritance.** Default Reasoning. Circumscription. Auto-epistemic Reasoning. Event Calculus Epistemic Logic. Knowledge and Belief.

References

1. Ronald J. Brachman, Hector J. Levesque: Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.
2. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.

Deep Learning for Natural Language Processing

(AIMLOE04)

Credit: 3+0

Course Objective: To provide the knowledge of Machine Learning for Engineering and science applications and its implementations in context of understanding the applications of ML.

Introduction to Deep Networks: Machine learning- Linear models (SVMs and, Decision Tree, Random Forest, Naïve Bayes, logistic regression)- Neural Networks: What a shallow network computes- Training a network: loss functions, back propagation and Gradient Descent, Stochastic Gradient Descent, Min-Batch Stochastic Gradient Descent, RMS Prop, ADAM, Different Types of Activation Functions.
History of Deep Learning- A Probabilistic Theory of Deep Learning- Backpropagation and regularization, batch normalization- VC Dimension and Neural Nets-Deep Vs Shallow Networks-Convolutional Networks- Generative Adversarial Networks (GAN), Semi-supervised Learning.

Dimensionality Reduction: Principle Component Analysis, Linear Discriminant Analysis, metric learning – Auto encoders and dimensionality reduction in networks - Introduction to Convnet - Architectures – AlexNet, VGG, Inception, ResNet - weights initialization, batch normalization, hyper parameter optimization.

Optimization and Generalization: Regularization Techniques (L1, L2, Elastic Net), Optimization in deep learning– non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM - Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning.

Case Study and Applications: Image net- Detection-Audio Wave Net-Natural Language Processing Word2Vec - Joint Detection-Bioinformatics- Face Recognition- Scene Understanding- Gathering Image Captions.

Course Outcome (CO): At the end of course, the student will be able

CO1: To present the mathematical, statistical and computational challenges of building neural networks.
CO2: To study the concepts of deep learning.
CO3: To introduce dimensionality reduction techniques.
CO4: To enable the students to know deep learning techniques to support real-time applications.

References

1. Cosma Rohilla Shalizi, Advanced Data Analysis from an Elementary Point of View, 2015.
2. Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
4. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.