South Indian Education Society's
Graduate School of Technology
NAAC 'A+' , NBA Accredited (EXTC,CE,IT)

Autonomous Institute Affiliated to University of Mumbai

Department of Artificial Intelligence (AI) and Data Science

Curriculum Structure for Undergraduate Academic Program in

Artificial Intelligence (AI) and Data Science

Third Year Syllabus

Board of Studies
Department of Artificial Intelligence (AI) and Data Science

Academic Council
SIES Graduate School of Technology

Effective from: AY 2026-27

PREAMBLE

Dear Students, Faculty, and Stakeholders,

Greetings and congratulations to all the stakeholders of SIES Graduate School of Technology for getting autonomous status to the college from the year 2024-25. University Grant Commission vide letter No. F. 2- 10/2023(AC-Policy) dated 10 July 2024 conferred the autonomous status affiliated to University of Mumbai for a period of 10 years from the academic year 2024-2025 to 2033-2034 as per clause 7.5 of the UGC (Conferment of Autonomous Status Upon Colleges and Measures for Maintenance of Standards in Autonomous Colleges) Regulations,2023. Autonomy will help design and implement curriculum which is more learner centric and robust to cater to the changing need of Society and Industry.

The Department of Artificial Intelligence (AI) and Data Science, SIES Graduate School of Technology, Nerul, Navi Mumbai, was established in 2021 and the current intake capacity is 120. The Department has designed its curricula using a top-down methodology. A clear and measurable set of learning objectives, the design of content that is aligned with the learning objectives, the integration of experiential learning through projects, skill laboratories, internships, and industry collaboration, the mapping of the program outcomes to courses, and the setting of the stage for ongoing evaluation and improvement are among the steps. Stakeholder consultation is the first step in the process. After identifying the industry requirements, four honors/minor tracks—Computing Technologies, Networking and Security, Multimodal Analysis and exceptional courses in Artificial Intelligence—are introduced.

The content of program electives, laboratory courses, core courses, and honors/minor courses has all been designed with the current industry trends in mind. We have added courses that will undoubtedly prepare our graduates for different industries in India.

Our program is meant to give students a thorough grasp of fundamental subjects including few core computing, artificial intelligence, web technology and data processing, in line with the transformative vision outlined in the National Education Policy (NEP) 2020. Students get the abilities necessary to handle challenging problems through interdisciplinary courses, skill labs, and specially created laboratory courses. Through independent laboratory courses, students are taught to practical engineering concepts such as cloud computing and data handling and visualization etc. The curriculum offers program electives in a variety of subjects, such as blockchain technology, cybersecurity and ethical hacking, multimodal analytics etc. to accommodate students with a wide range of interests.

Teachers have a plethora of chances within the independent curriculum to innovate and improve the learning experience for students. Instructors can take part in research and development projects, actively contribute to curriculum creation, and provide internships for students to experience new things.
We aspire to empower our graduates to emerge as leaders, innovators, and global ambassadors of excellence in Artificial Intelligence and Data Science by cultivating creativity, resilience, and a curious mindset. We invite all interested stakeholders to join us in transforming engineering education as we embark on this groundbreaking journey. Together, let's strive for excellence, innovation, and a lasting impact on society.

Chairperson
Board of Studies
Artificial Intelligence (AI) and Data Science
SIES Graduate School of Technology

HEAD of the Department
Artificial Intelligence & Data Science
SIES Graduate School of Technology
Sri Chandrasekarendra Saraswathy Vidyapuram
Plot 1C/D/E, Sector-V, Nerul, Navi Mumbai-400706

Chairperson
Academic Council
SIES Graduate School of Technology

I/c PRINCIPAL
SIES-GRADUATE SCHOOL OF TECHNOLOGY (AUTONOMOUS)
Plot 1C/D/E Sri Chandrasekarendra Saraswathy Vidyapuram
Sector V Nerul Navi Mumbai-400706

SIES Graduate School of Technology
DEPARTMENT OF ARTIFICIAL INTELLIGENCE (AI) AND DATA SCIENCE
BACHELOR OF ENGINEERING

Semester wise Credit distribution structure for Four Year UG Engineering

Program - Artificial Intelligence (AI) and Data Science

Semester		I	II	III	IV	V	VI	VII	VIII	Total Credits
Basic Science Course (BSC)	BSC/ESC	7	6		--	--	--	--	--	13
Engineering Science Course (ESC)		9	10		--	--	--	--	--	19
Programme Core Course (PCC)	Program Courses	--	--	17	11	12	10	04	--	54
Programme Elective Course (PEC)		--	--	--	--	03	04	07	--	14
Multidisciplinary Minor (MDM)	Multidisciplinary Courses		-	--	03	04	04	04	--	15
Open Elective (OE) Other than a particular program		--	--	--	--	--	--	03	03	06
Vocational and Skill Enhancement Course (VSEC)	Skill Courses	01	01	02	02	--	02	--	--	08
Ability Enhancement Course (AEC - 01, AEC-02)	Humanities Social Science and Management (HSSM)		02	--	--	02	--	--	--	04
Entrepreneurship/Economics/ Management Courses		--		02	02	--	--	--	--	04
Indian Knowledge System (IKS)			02		--	--	--	--	--	02
Value Education Course (VEC)		--	--	--	02	--	--	--	--	02
Research Methodology (RM)	Experiential Learning Courses	--	--	--	--	--	--	--	03	03
Community Engagement. Project (CEP) / Field Project (FP)		--	--	01	01	--	--	-	-	02
Project		--	--	--	--	01	02	02	01	06
Internship/On Job Training (OJT)		--	---		--	--	--		12	12
Co-curricular Courses (CC)	Liberal Learning Courses	04			--	--	--	--	--	04
Total Credits (Major)		**21**	**21**	**22**	**21**	**22**	**22**	**20**	**19**	**168**

CURRICULUM STRUCTURE

AND

THIRD YEAR SYLLABUS

(ARTIFICIAL INTELLIGENCE (AI) and DATA SCIENCE)

Academic Year 2026-27

Nomenclature of the courses in the curriculum	
Abbreviation	**Title**
BSC	Basic Science Courses
ESC	Engineering Science Courses
PCC	Program Core Courses
PEC	Program Elective Course
MDM	Multi-Disciplinary Minor
OE	Open Elective
VSEC	Vocational and Skill Enhancement Course
AEC	Ability Enhancement Course
IKS	Indian Knowledge System
VEC	Value Education Course
RM	Research Methodology
CEP/FP	Community Engagement Project/Field Pro
OJT	Internship/On Job Training
CC	Co-Curricular Courses
ISE	In Semester Examination
MSE	Mid Semester Examination
ESE	End Semester Examination
CIAP	Continuous Internal Assessment Practical
ESEP	End Semester Examination Practical

Program Structure for First Year

W.E.F. A.Y. 2025-26

Semester I

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract	Tut.	Theor	Pract.	Tut.	Total
FEC101	Applied Mathematics -I	BSC	3	--	--	3	--	--	3
FEC1021/ FEC1022	Applied Physics/ Applied Chemistry @	BSC	3	--	--	3	--	--	3
FEC103	Basic Electrical & Electronics Engineering	ESC	2	--	--	2	--	--	2
FEC104	C-Programming	ESC	2	--	--	2	--	--	2
FEC105	Applied Mechanics and Robot Dynamics	ESC	2	--	--	2	--	--	2
FEL1011/ FEL1012	Applied Physics Lab/ Applied Chemistry Lab @	BSC	--	1	--	--	0.5	--	0.5
FEL102	Basic Electrical & Electronics Engineering Lab	ESC	--	2	--	--	1	--	1
FEL103	C-Programming Lab	ESC	--	2	--	--	1	--	1
FEL104	Applied Mechanics and Robot Dynamics Lab	ESC	--	2	--	--	1	--	1
FEL105	Engineering Workshop-I	VSEC	--	2	--	--	1	--	1
FEL106	Health, Wellness and Mindfulness	CC	--	2#+2	--	--	2	--	2
FEL107	Induction Cum Universal Human Values	CC	--	5*	--	--	2.5	--	2.5
	Total		**12**	**18**	--	**12**	**9**	--	**21**

Examination Scheme-FY Semester-I

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
FEC101	Applied Mathematics -I	20	20	60	3	--	--	100
FEC1021/ FEC1022	Applied Physics/ Applied Chemistry @	20	20	60	3	--	--	100
FEC103	Basic Electrical & Electronics Engineering	15	15	45	2	--	--	75
FEC104	C-Programming	15	15	45	2	--	--	75
FEC105	Applied Mechanics and Robot Dynamics	15	15	45	2	--	--	75
FEL1011/ FEL1012	Applied Physics Lab/ Applied Chemistry Lab@	--	--	--	--	25	--	25
FEL102	Basic Electrical & Electronics Engineering Lab	--	--	--	--	25	25	50
FEL103	C-Programming Lab	--	--	--	--	25	25	50
FEL104	Applied Mechanics and Robot Dynamics Lab	--	--	--	--	25	25	50
FEL105	Engineering Workshop-I	--	--	--	--	25	--	25
FEL106	Health, Wellness and Mindfulness	--	--	--	--	25	–	25
FEL107	Induction Cum Universal Human Values	--	--	--	--	25	--	25
Total		**85**	**85**	**255**		**175**	**75**	**675**

@Physics/Chemistry in one semester.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.
Theory 1 credit for 1 hour and practical 1 credit for 2 hours.
*Indicates workload of a learner for UHV. Faculty Load: ½ hour per week per four groups
Two hours of practical class to be conducted for full class as demo/ discussion.
ISE: In Semester Examination: To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test etc. of 20/15 marks.
MSE: Mid Semester Examination: To be conducted as written examination for 20/15 marks of duration 1 Hr.
ESE: End Semester Examination
CIAP: Continuous Internal Assessment Practical. Term work will be assessed as Continuous Internal Assessment Practical (CIAP). ESEP: End Semester Examination Practical. Oral/Practical Examination will be conducted as End Semester Examination Practical (ESEP).

SIES Graduate School of Technology
DEPARTMENT OF ARTIFICIAL INTELLIGENCE (AI) AND DATA SCIENCE
BACHELOR OF ENGINEERING

Program Structure for First Year
W.E.F. A.Y. 2025-26
Semester II

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
FEC201	Applied Mathematics -II	BSC	3	--	--	3	--	--	3
FEC2021/ FEC2022	Applied Physics/ Applied Chemistry @	BSC	3	--	--	3	--	--	3
FEC203	Engineering Graphics	ESC	2	--	--	2	--	--	2
FEC204	Digital System Design	ESC	3	--	--	3	--	--	3
FEC205	Professional Communication Techniques	AEC	2	--	--	2	--	--	2
FEL2011/ FEL2012	Applied Physics Lab/ Applied Chemistry Lab @	BSC	--	1	--	--	0.5	--	0.5
FEL202	Engineering Graphics Lab	ESC	--	2	--	--	1	--	1
FEL203	Digital System Design Lab	ESC	--	2	--	--	1	--	1
FEL204	Professional Communication Techniques Lab	AEC	--	1	--	--	0.5	--	0.5
FEL205	Object Oriented Programming Methodology Lab	ESC	--	2*+2	--	--	2	--	2
FEL206	Engineering Workshop-II	VSEC	--	2	--	--	1	--	1
FEL207	Indian Knowledge System	HSSM	--	2*+2	--	--	2	--	2
	Total		**13**	**16**	--	**13**	**8**	--	**21**

Examination Scheme-FY Semester-II

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
FEC201	Applied Mathematics -II	20	20	60	03	--	--	100
FEC2021/ FEC2022	Applied Physics/ Applied Chemistry @	20	20	60	03	--	--	100
FEC203	Engineering Graphics	15	15	45	03	--	--	75
FEC204	Digital System Design	20	20	60	03	--	--	100
FEC205	Professional Communication Techniques	15	15	45	02	--	--	75
FEL2011/ FEL2012	Applied Physics Lab/ Applied Chemistry Lab @	--	--	--	--	25	--	25
FEL202	Engineering Graphics Lab	--	--	--	--	25	25	50
FEL203	Digital System Design Lab	--	--	--	--	25	25	50
FEL204	Professional Communication Techniques Lab	--	--	--	--	25	--	25
FEL205	Object Oriented Programming Methodology Lab	--	--	--	--	25	25	50
FEL206	Engineering Workshop-II	--	--	--	--	25	--	25
FEL207	Indian Knowledge System	--	--	--	--	25	–	25
Total		**90**	**90**	**270**	**--**	**175**	**75**	**700**

@Physics/Chemistry in one semester.
* Two hours of practical class to be conducted for full class as demo/ discussion.
# Course evaluation is an activity based which may be an individual or group of students.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.
Theory 1 credit for 1 hour and practical 1 credit for 2 hours.
ISE: In Semester Examination: To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test etc. of 20/15 marks.
MSE: Mid Semester Examination: To be conducted as written examination for 20/15 marks of duration 1 Hr.
ESE: End Semester Examination
CIAP: Continuous Internal Assessment Practical. Term work will be assessed as Continuous Internal Assessment Practical (CIAP). ESEP: End Semester Examination Practical. Oral/Practical Examination will be conducted as End Semester Examination Practical (ESEP).

Program Structure for Second Year

W.E.F. A.Y. 2026-27

Semester III

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC301	Applied Mathematics III	PCC	3	--	--	3	--	--	3
AIDSC302	Data Structure	PCC	3	--	--	3	--	--	3
AIDSC303	Discrete Structure and Graph Theory	PCC	3	--	--	3	--	--	3
AIDSC304	Database Management Systems	PCC	3	--	--	3	--	--	3
AIDSC305	Computer Organization and Architecture	PCC	2	--	--	2	--	--	2
AIDSC306	Engineering Economics	HSSM	2	--	--	2	--	--	2
AIDSL301	Data Structure Lab	PCC	--	2	--	--	1	--	1
AIDSL302	Database Management Systems Lab	PCC	--	2	--	--	1	--	1
AIDSL303	Computer Organization and Architecture Lab	PCC	--	2	--	--	1	--	1
AIDSL304	Skill Lab- Python Programming	VSEC	--	2*+2	--	--	2	--	2
AIDSM301	Mini Project 1A	CEP	--	2#	--	--	1	--	1
Total			**16**	**12**	**--**	**16**	**6**	**--**	**22**

Examination Scheme - AIDS Semester-III

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC301	Applied Mathematics III	20	20	60	3	--	--	100
AIDSC302	Data Structure	20	20	60	3	--	--	100
AIDSC303	Discrete Structure and Graph Theory	20	20	60	3	--	--	100
AIDSC304	Database Management Systems	20	20	60	3	--	--	100
AIDSC305	Computer Organization and Architecture	15	15	45	2	--	--	75
AIDSC306	Engineering Economics	25	25	--	--	--	--	50
AIDSL301	Data Structure Lab	--	--	--	--	25	25	50
AIDSL302	Database Management Systems Lab	--	--	--	--	25	25	50
AIDSL303	Computer Organization and Architecture Lab	--	--	--	--	25	--	25
AIDSL304	Skill Lab-Python Programming	--	--	--	--	25	25	50
AIDSM301	Mini Project 1A	--	--	--	--	25	25	50
Total		120	120	285	--	125	100	**750**

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Program Structure for Second Year

W.E.F. A.Y. 2026-27

Semester IV

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC401	Applied Mathematics-IV	PCC	3	--	--	3	--	--	3
AIDSC402	Operating System	PCC	3		--	3	--	--	3
AIDSC403	Analysis of Algorithm	PCC	3	--	--	3	--	--	3
AIDSC404	Critical Thinking and Design	HSSM	2	--	--	2	--	--	2
MDMC40X1	Multidisciplinary Minor Course I (MDM I)	MDM	3		--	3	--	--	3
AIDSL401	Operating Systems Lab	PCC	--	2	--	--	1	--	1
AIDSL402	Analysis of Algorithm Lab	PCC	--	2	--	--	1	--	1
AIDSL403	Skill Lab- Web Computing and Networking	VSEC	--	2*+2	--	--	2	--	2
AIDSL404	Value Education (UHV)	HSSM (VEC)	--	4	--	--	2	--	2
AIDSM401	Mini Project 1B	CEP	--	2#	--	--	1	--	1
Total			14	14	--	14	07	--	21

Examination Scheme - AIDS Semester-IV

Course Code	Course Name	Theory				CIAP	ESEP	Total
		Internal		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC401	Applied Mathematics-IV	20	20	60	3	--	--	100
AIDSC402	Operating System	20	20	60	3	--	--	100
AIDSC403	Analysis of Algorithm	20	20	60	3	--	--	100
AIDSC405	Critical Thinking and Design	15	15	45	2	--	--	75
MDMC40X1	Multidisciplinary Minor Course I (MDM I)	20	20	60	3	--	--	100
AIDSL401	Operating Systems Lab	--	--	--	--	25	25	50
AIDSL402	Analysis of Algorithm Lab	--	--	--	--	25	25	50
AIDSL403	Skill Lab- Web Computing and Networking	--	--	--	--	25	25	50
AIDSL404	Value Education (UHV)	--	--	--	--	50	--	50
AIDSM401	Mini Project 1B	--	--	--	--	25	25	50
Total		95	95	285	14	150	100	725

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Program Structure for Third Year

W.E.F. A.Y. 2027-28

Semester V

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC501	Artificial Intelligence	PCC	3	-	-	3	-	-	3
AIDSC502	Data Engineering	PCC	3	-	-	3	-	-	3
AIDSC503	Data warehousing & Mining	PCC	3	-	-	3	-	-	3
AIDSPE501X	Program Elective-I	PEC	3	-	-	3	-	-	3
MDMC50X2	Multi-Disciplinary course-II	MDM	3	-	-	3	-	-	3
AIDSL501	Artificial Intelligence Lab	PCC	-	2	-	-	1	-	1
AIDSL502	Data Engineering Lab	PCC	-	2	-	-	1	-	1
AIDSL503	Data warehousing & Mining Lab	PCC	-	2	-	-	1	-	1
AIDSL504	Interpersonal & Career Skills	HSSM (AEC)	-	2*+2	-	-	2	-	2
MDML50X1	Multi-Disciplinary Minor course-II Lab/Tut	MDM	-	2&	-	-	1&	-	1
AIDSM501	Mini Project 2	Project	--	2#	--	--	1	--	1
Total			15	14	--	15	7	-	22

Examination Scheme - AIDS Semester-V

Course Code	Course Name	Examination Scheme						
		Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC501	Artificial Intelligence	20	20	60	3	-	-	100
AIDSC502	Data Engineering	20	20	60	3	-	-	100
AIDSC503	Data warehousing & Mining	20	20	60	3	-	-	100
AIDSPE501X	Program Elective-I	20	20	60	3	-	-	100
MDMC50X2	Multi-Disciplinary course-II	20	20	60	3	-	-	100
AIDSL501	Artificial Intelligence Lab	-	-	-	-	25	25	50
AIDSL502	Data Engineering Lab	-	-	-	-	25	-	25
AIDSL503	Data warehousing & Mining Lab	-	-	-	-	25	25	50
AIDSL504	Interpersonal and Career Skills	-	-	-	-	50	-	50
MDML50X1	Multi-Disciplinary Minor course-II Lab/Tut	-	-	-	-	25	-	25
AIDSM501	Mini Project -2	-	-	-	-	25	25	50
Total		100	100	300	-	175	75	750

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

& Tuts only for Management MDM (MDMC5062)

Program Elective – I

Technology Bucket			
Computing Technologies	**Blockchain Technology**	**Data Analytics**	**Virtual Reality and Game Theory**
AIDSPEC5041: Internet of Things	**AIDSPEC5042:** Bitcoin and Cryptocurrency	**AIDSPEC5043:** Statistics for Artificial Intelligence & Data Science	**AIDSPEC5044:** Computer Graphics and Game theory

Program Structure for Third Year
W.E.F. A.Y. 2027-28
Semester VI

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC601	Machine Learning	PCC	3		-	3	-	-	3
AIDSC602	Big Data Analytics	PCC	3		-	3	-	-	3
AIDSC603	Software Engineering & Project Management	PCC	2		-	2	-	-	2
AIDSPE601X	Program Elective-II	PEC	3		-	3	-	-	3
MDMC60X3	Multi-Disciplinary Minor course-III	MDM	3		-	3	-	-	3
AIDSL601	Intelligent Data Analytics Lab	PCC	-	2	-	-	1	-	1
AIDSL602	Software Engineering & Project Management Lab	PCC	-	2	-	-	1	-	1
AIDSL603	Skill Lab- Data Handling and Visualization	VSEC	-	2*+2	-	-	2	-	2
AIDSPEL601X	Program Elective-II Lab	PEC	-	2	-	-	1	-	1
MDML60X2	Multi-Disciplinary Minor course-III Lab/Tut	MDM	-	2[&]	-	-	1[&]	-	1
AIDSM601	Major Project I	Project	--	4[#]	-	--	2	--	2
Total			14	16	-	14	8	-	22

Examination Scheme - AIDS Semester-VI

Course Code	Course Name	Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC601	Machine Learning	20	20	60	3	-	-	100
AIDSC602	Big Data Analytics	20	20	60	3	-	-	100
AIDSC603	Software Engineering & Project Management	15	15	45	2	-	-	75
AIDSPE601X	Program Elective-II	20	20	60	3	-	-	100
MDMC60X3	Multi-Disciplinary Minor course-III	20	20	60	3	-	-	100
AIDSL601	Intelligent Data Analytics Lab	-	-	-	-	25	25	50
AIDSL602	Software Engineering & Project Management Lab	-	-	-	-	25	-	25
AIDSL603	Skill Lab-Data Handling and Visualization	-	-	-	-	25	25	50
AIDSPEL601X	Program Elective-II Lab	-	-	-	-	25	-	25
MDML60X2	Multi-Disciplinary Minor course-III Lab	-	-	-	-	25	-	25
AIDSM601	Major Project I	-	-	-	-	25	25	50
Total		95	95	285	14	150	75	700

* Two hours of practical class to be conducted for full class as demo/ discussion.

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.

$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

& Tuts only for Management MDM (MDMC6063)

Program Elective – II

Technology Bucket			
Computing Technologies	**Blockchain Technology**	**Data Analytics**	**Virtual Reality and Game Theory**
AIDSPEC6011: Robotics and Applications	AIDSPEC6012: Blockchain Technology	AIDSPEC6013: Image Processing and Video processing	AIDSPEC6014: Augmented reality and Virtual Reality

Program Structure for Final Year
W.E.F. A.Y. 2028-29
Semester VII

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC701	Deep Learning	PCC	3	-	-	3	-	-	3
AIDSPEC701X	Program Elective-III	PEC	3	-	-	3	-	-	3
AIDSCPE702X	Program Elective-IV	PEC	3	-	-	3	-	-	3
MDMC70X4	Multi-Disciplinary Minor course-IV	MDM	3	-	-	3	-	-	3
OEC701X	Open Elective-I	OE	3	-	-	3	-	-	3
AIDSL701	Deep Learning Lab	PCC	-	2	-	-	1	-	1
AIDSPEL701X	Program Elective-III Lab	PEC	-	2	-	-	1	-	1
MDML70X3	Multi-Disciplinary Minor course-IV Lab/Tut	MDM	-	2[&]	-	-	1[&]	-	1
AIDSM701	Major Project II	Project	-	4[#]	-	-	2	-	2
Total			**15**	**10**		**15**	**5**		**20**

Examination Scheme - AIDS Semester-VII

Course Code	Course Name	Theory				CIAP	ESEP	Total
		Internal Assessment		ESE[$]	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC701	Deep Learning	20	20	60	3	--	-	100
MDMC70X4	Multi-Disciplinary Minor course -IV	20	20	60	3	--	-	100
AIDSPEC701X	Program Elective-III	20	20	60	3	--	-	100
AIDSCPE702X	Program Elective-IV	20	20	60	3	--	-	100
OEC701X	Open Elective-I	20	20	60	3	--	-	100
AIDSL701	Deep Learning Lab	--	-	--	-	25	25	50
AIDSPEL701X	Program Elective-III Lab	--	-	--	-	25	25	50
MDML70X3	Multi-Disciplinary Minor course - Lab IV	--	-	--	-	25	--	25
AIDSM701	Major Project II	--	--	--	--	25	25	50
Total		100	100	300	--	75	75	**675**

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.
& Tuts only for Management MDM (MDMC 7064)

Program Elective – III

Technology Bucket			
Computing Technologies	**Blockchain Technology**	**Data Analytics**	**Virtual Reality and Game Theory**
AIDSPEC7011: Wiresless Sensor Networks	**AIDSPEC7012:** Advanced Blockchain Technology	**AIDSPEC7013:** Natural Language Processing	**AIDSPEC7014:** UX/ UI Design

Program Elective – IV

Technology Bucket			
Computing Technologies	**Blockchain Technology**	**Data Analytics**	**Virtual Reality and Game Theory**
AIDSPEC7021: Edge and Fog Computing	**AIDSPEC7022:** Decentralized Finances	**AIDSPEC7023:** Social Media Analytics	**AIDSPEC7024:** Computational Game Theory

Open Elective -I

Course Code	Course Name
OEC7011	Project Management
OEC7012	Finance Management
OEC7013	Management Information System
OEC7014	Entrepreneurship Development and
OEC7015	Operation Research
OEC7016	Disaster Management and Mitigation Measures
OEC7017	Product Design

Program Structure for Final Year
W.E.F. A.Y. 2028-29
Semester VIII

Course Code	Course Name	Category	Teaching Scheme (Contact Hours)			Credits Assigned			
			Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
AIDSC801	Research Methodology	RM	3	-	-	3	-	-	3
OEC801X	Open Elective-II	OE	3	-	-	3	-	-	3
AIDSM801	Major Project III	Project	-	2#	-		1	-	1
AIDSINT801	Internship/Project/Research	Internship	-	-	-	-	-	-	12
Total			**6**	**28**	**-**	**6**	**14**	**-**	**19**

Examination Scheme - AIDS Semester-VIII

Course Code	Course Name	Theory				CIAP	ESEP	Total
		Internal Assessment		ESE$	Exam Duration (Hrs.)			
		ISE	MSE					
AIDSC801	Research Methodology	20	20	60	3	-	-	100
OEC801X	Open Elective-II	20	20	60	3	-	-	100
AIDSM801	Major Project III	-	-	-	-	100	50	150
AIDSNT801	Internship/Project/Researc	-	-	-	-	200	--	200
Total		40	40	120	-	300	50	**550**

# Indicates workload of a learner (Not faculty) for Mini Project 1A. Faculty Load: ½ hour per week per four groups.
$ ESE of duration 3 hours are of 80 marks and scaled to 60. ESE duration of 2 hours are of 60 marks and scaled to 45.

Open Elective -II

Course Code	Course Name
OEC8011	Enterprise Resource Planning
OEC8012	Cyber Security and Laws
OEC8013	Energy Audit and Management
OEC8014	IPR and Patenting
OEC8015	Environmental Management
OEC8016	Digital Business Management
OEC8017	Human Resource Management

Multidisciplinary Minor (MDM)

Track	Minor Track	Partner Institute if any	Year/ Sem	Module	Code	Eligibility
1	Machine Learning	SIES GST	SE/IV	Artificial Intelligence	MDMC4011	IT/EXTC/CS E IOT
			TE/V	Machine Learning	MDMC5012	
				Machine Learning Lab	MDML5011	
			TE/VI	Natural Language Processing	MDMC6013	
				Natural Language Processing Lab	MDML6012	
			BE/VII	Deep Learning	MDMC7014	
				Deep Learning lab	MDML7013	
2	Data Science	SIES GST	SE/IV	Statistical Foundation for Data Science	MDMC4021	ECS/CE/EXTC
			TE/V	Data Analytics & Visualization	MDMC5022	
				Data Analytics & Visualization Lab	MDML5021	
			TE/VI	Decision Making & Business Intelligence	MDMC6023	
				Decision Making & Business Intelligence Lab	MDML6022	
			BE/VII	Big Data Analytics	MDMC7024	
				Big Data Analytics Lab	MDML7023	
3	Embedded Systems	SIES GST	SE/IV	Microprocessor and Microcontrollers	MDMC4031	CE/AIDS/AIML
			TE/V	Embedded Systems and RTOS	MDMC5032	
				Embedded Systems and RTOS Lab	MDML5031	
			TE/VI	Sensor Technology	MDMC6033	
				Sensor Technology Lab	MDML6032	
			BE/VII	Industrial Internet of Things	MDMC7034	
				Industrial Internet of Things Lab	MDML7033	
4	Cyber Security	SIES GST	SE/IV	Computer Network	MDMC4041	AIDS/AIML
			TE/V	Cryptography & Network Security	MDMC5042	
				Cryptography & Network Security Lab	MDML5041	
			TE/VI	Cloud Computing and Security	MDMC6043	
				Cloud Computing and Security Lab	MDML6042	
			BE/VII	Digital Forensics	MDMC7044	
				Digital Forensics Lab	MDML7043	
5	System Programming	SIES GST	SE/IV	Advance Data Structure	MDMC4051	CSEIOT/ECS/ IT
			TE/V	Advance Algorithm	MDMC5052	
				Advance Algorithm Lab	MDML5051	
			TE/VI	System Programming and Compiler Construction	MDMC6053	
				System Programming and Compiler Construction Lab	MDML6052	
			BE/VII	Distributed Systems	MDMC7054	
				Distributed Systems Lab	MDML7053	
6	Management	SIESSBS	SE/IV	Cost Management	MDMC4061	EXTC/CE/IT / ECS/AIDS/ AIML/CSE IOT
			TE/V	Supply Chain Management	MDMC5062	
			TE/VI	Organizational Behavior and Human Resource Management	MDMC6063	
			BE/VII	Marketing Management	MDMC7064	

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSC501	Artificial Intelligence	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIDSC501	Artificial Intelligence	20	20	60	--	--	100

Pre-requisite:
FEC104 C- Programming CSL405
Python Programming

Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO4:Conduct Investigations of Complex Problems
5. PO5:Engineering Tool Usage
6. PO6:The Engineer and the world

Course Objectives:

1. Introduce core concepts, history, and scope of Artificial Intelligence.
2. Model real-world problems as search, planning, and reasoning tasks.
3. Teach algorithmic strategies for search and adversarial decision making.
4. Cover knowledge representation, logical inference, and ontologies.
5. Explain probabilistic reasoning and decision-making under uncertainty.
6. Expose students to AI tools, applications, ethics and provide hands-on experience.

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

1. Explain AI fundamentals, agent architectures, and problem formulation.
2. Apply uninformed/informed and adversarial search algorithms to solve problems.
3. Represent knowledge using logic, rules, and basic ontologies and perform inference.
4. Use probabilistic models (Bayes nets, HMMs, MDPs) for reasoning under uncertainty.
5. Develop basic planners and understand fundamentals of reinforcement learning.
6. Analyze AI applications, use standard AI libraries, and address ethical issues.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction and Agents**	**05**	
	1.1	Introduction: History, Turing test, milestones, Intelligent agents, agent architectures, PEAS, Environments, task types, performance measures, Problem formulation (state space, goal, path cost), AI vs ML vs DL, AI applications overview, Tools overview (Python, notebooks, libraries)		**CO1**
		Self-Learning Topics: Python AI overview: NumPy, SciPy, Matplotlib, Jupyter, Anaconda setup		
2.0		**Search for Solution**	**07**	
	2.1	Uninformed search: BFS, DFS, Depth-limited, Iterative deepening, Heuristics: admissibility, consistence. Informed search: Greedy best-first, A* (proofs, examples) Local search: Hill climbing, simulated annealing, Genetic algorithms and search trade-offs, Adversarial search: Minimax, Alpha-Beta pruning, Introduction to Constraint Satisfaction Problems		**CO2**
		Self-Learning Topics: Meta-heuristics overview: Ant Colony Optimization, PSO basics		
3.0		**Knowledge representation and Reasoning**	**08**	
	3.1	Knowledge-based agents; KB architectures, Propositional logic: syntax & semantics, Inference: resolution, forward/backward chaining, First-order logic: syntax, quantifiers, examples, Unification, substitution, resolution in FOL, Rule-based systems and production systems, Ontologies, semantic web basics, KR design patterns, Prolog Programming.		**CO3**
		Self-Learning Topics: Knowledge graphs and semantic web tools (RDF, SPARQL)		
4.0		**Probabilistic reasoning and uncertainty**	**07**	
	4.1	Probability basics for AI, conditional probability, Bayesian networks, Inference in Bayes nets MLE for simple models, Markov models & HMMs (forward-backward, Viterbi intro) MDPs: states, actions, rewards, Introduction to Bellman equation		**CO4**
		Self-Learning Topics: Hidden Markov Model applications (POS tagging, speech processing), Fuzzy Logic & Reasoning.		
5.0		**Planning and Reinforcement**	**06**	
	5.1	Classical planning: STRIPS representation, State-space planners: forward/backward search, Partial-order planning, Graph Plan algorithm overview, Introduction to reinforcement learning: reward, policy, value Tabular RL: dynamic programming, policy iteration (basic).		**CO5**
		Self-Learning Topics: Applications of planning in robotics and autonomous systems		
6.0		**Applications, Ethics & Tools**	**06**	
	6.1	AI in NLP and Vision (case studies), AI in robotics & healthcare (case		**CO6**

		studies), LLMs, transformer basics & tool ecosystem (scikit-learn, TF, PyTorch), Explainable AI, fairness, bias & responsible AI, Ethics, regulations, deployment risks, safety.		
		Self-Learning Topics: AI model deployment: Flask/FastAPI basics, Docker intro, Agentic AI.		
		Total	**39**	

Textbooks:

1.Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach-4th Edition, Pearson 2020.
2.Elaine Rich and Kevin Knight, Artificial Intelligence-5th Edition, Tata McGraw-Hill, July 2025.
3.George F Luger, Artificial Intelligence-Fourth Edition, Pearson, 2024

Reference books:

1.Nils J. Nilsson, Principles of Artificial Intelligence, First Edition, Springer, 2014
2.Tom M. Mitchell, Machine Learning, First Edition, McGraw-Hill Education Pvt Ltd, 2013.
3.Saroj Kaushik, Artificial Intelligence and Expert Systems, Cengage Learning, 2022.
4.Jurafsky & Martin, Speech and Language Processing (for NLP), Second Edition, Pearson, 2026.

Online References:

- MIT OCW — 6.034 / AI course materials (lectures & assignments). (MIT Open Course Ware)
- Stanford CS221 — course site & materials. (cs221.stanford.edu)
- NPTEL AI courses (IITs) — online video lectures & notes. (NPTEL)
- OpenAI Academy / OpenAI resources for education. (OpenAI Academy)
- Google AI education / Learn AI skills pages. (Google for Education)
- scikit-learn documentation (ML basics & examples). (scikit-learn.org)
- TensorFlow official site (tutorials & API). (TensorFlow)
- PyTorch official site (docs & tutorials). (PyTorch)

Course Assessment: ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks= 05 marks for attendance + 15 marks for activities

MSE

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question 1(20 marks) : - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2(40 marks) : - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3(20 marks) :- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSC 502	Data Engineering	3	-	-	3	-		3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSC 502	Data Engineering	20	20	60	-	-	100

Pre-requisite:

1. Database Management Systems
2. Python Programming
3. Operating System

Program Outcomes Addressed

1. PO1 – Engineering Knowledge
2. PO2 – Problem Analysis
3. PO3 – Design/Development of Solutions
4. PO5 – Modern Tool Usage
5. PO6 – Engineer & Society
6. PO7 – Environment & Sustainability
7. PO12 – Life-long Learning

Course Objectives:

1. To provide foundational knowledge of data engineering concepts, lifecycle, responsibilities, and modern data stack components.
2. To develop an understanding of data generation mechanisms in operational systems, storage architectures, and principles of data reliability, consistency, and quality.
3. To enable learners to design and implement data ingestion workflows, understand streaming/batch ingestion, and apply basic data lineage analysis.
4. To impart skills in data orchestration, modeling approaches, transformation techniques, and their role in analytical and operational ecosystems.
5. To introduce distributed data processing frameworks such as Hadoop and Spark, and provide hands-on experience with parallel computation concepts.
6. To familiarize students with data serving layers, data security fundamentals, and emerging trends, including real-time pipelines, data mesh, and ML-integrated data architectures.

Course Outcomes: Learners will be able to

1. Explain the concepts, scope, and lifecycle of data engineering, including data formats, quality dimensions, and engineering roles.

2. Describe and differentiate various data generation mechanisms, storage systems, consistency models, and cloud-native architectures for data management.
3. Design and implement basic data ingestion workflows for batch and streaming systems, apply serialization techniques, and analyze data lineage within data pipelines.
4. Construct data models such as normalized schemas, star/snowflake schemas, and implement orchestration workflows using tools such as Airflow.
5. Apply distributed data processing principles using Hadoop and Spark, perform transformations, optimize execution workflows, and evaluate system performance.
6. Develop a conceptual data serving layer, incorporate security and governance controls, and examine future trends in real-time data engineering and ML-driven architectures.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0	**01**	**Foundations of Data Engineering**	**05**	
	1.1	Understanding Data Engineering: Definition, Architecture, Role, & Scope. Data Engineering Lifecycle: Generation- Storage- Ingestion Transformation- Serving, OLTP vs OLAP		**CO1**
	1.2	Data types & formats, Data Quality Fundamentals: accuracy, completeness, consistency, lineage, validation rules.		
	1.3	Data Engineering Responsibilities & Skills: • Technical responsibilities: pipelines, systems, integrations • Business responsibilities: data quality, governance, collaboration • Org-level setup: internal vs external data engineers		
		Self-learning Topics: Data maturity levels • Emerging tools and modern data stack snapshot • High-level case studies (e.g., Netflix, Uber, Swiggy) • Hands-on exploration of datasets • Converting between common data formats • Simple modelling of a business dataset into star schema		
2.0	**02**	**Data Generation & Storage Architectures.**	**07**	
	2.1	Change Data Capture (CDC): • Database Logs, CRUD, insert-only models. • Real-time event generation (Kafka-style messaging)		**CO2**
	2.2	Data Generation in Source System: Operational DBs (OLTP), API systems, log-based systems, Batch vs Streaming data generation.		
	2.3	Storage Fundamentals: File, block, and object storage, Serialization, compression, and caching basics, GCP Big Query, Cloud Storage. Data Storage Systems: Distributed storage concepts, Eventual vs strong consistency, Indexing, partitioning, clustering fundamentals		
	2.4	Data Warehouses, Lakes & Lakehouses: When to use what, Cloud-native storage architecture, Delta Lake architecture, Data Catalogs, schema management, Time travel, Table optimization (Z-order, vacuum)		

		Self-learning Topics. Reading server logs, Sample API pulls, Modern File Formats (Parquet, Avro, ORC), Introduction to Delta Lake / Iceberg (Lakehouse Table Formats).		
3.0	**03**	**Data Ingestion and Lineage Analysis.**	**06**	
	3.1	Ingestion Fundamentals: Bounded vs unbounded data, Stream ingestion (Kafka, AWS Kinesis conceptually)		**CO3**
	3.2	Methods of Ingestion, Direct DB ingestion, API ingestion, Message queue & event streaming, File-based ingestion (CSV/JSON/Parquet)		
	3.3	Engineering Concerns: Serialization/deserialization, Throughput, latency, durability, DLQs (Dead Letter Queues), Ordering and reply.		
	3.4	Introduction to Data Lineage, Building a Data Flow, Edge Intelligence in Data Flow, Data Lineage — Benefits and Importance, Error handling strategies		
		Self-learning Topics: Batch ingestion (ELT/ETL), building a small batch pipeline using Python, a Simple streaming pipeline using Kafka / Redpanda / MQTT, and applying data cleaning rules on real datasets		
4.0	**04**	**Data Orchestration, Modelling and Transformation**	**07**	
	4.1	Data pipeline orchestration: need and principles, Apache Airflow fundamentals: DAGs, operators, tasks, scheduling, Monitoring & logging for data systems		**CO4**
	4.2	Queries & Optimization: Query life cycle, Cost-based optimization, Practical indexing, partitioning, clustering		
	4.3	Data Modeling: Conceptual, logical, physical models, Normalization & denormalization, Data Vault concepts (light intro)		
	4.4	Transformations (Batch & Streaming): ELT workflows, Materialized views, Federation & query virtualization, Streaming transformations (windowing, late events)		
		Self-learning Topics: **Introduction to Tableau:** Creating dashboards, storyboards, interactive filters, and connecting to different data sources.		
5.0	**05**	**Distributed Data Processing**	**06**	
	5.1	Distributed computing: fundamentals and needs Characterization of Distributed Systems: Types of distributed systems (Difference between tightly coupled and loosely coupled multiprocessor systems) Distributed Computing System Models. Distributed Share Memory.		**CO5**
	5.2	Distributed File Systems: Physical Organization of Apache Spark architecture: RDDs, Data Frames, DAG scheduler, Spark transformations, and actions.		
	5.3	Cluster modes: local, standalone, Kubernetes Performance tuning basics: caching, partitioning, shuffle optimization		

		Self-learning Topics: Fault Tolerance & Data Locality in Distributed Processing, Modern Data Storage Formats & Lakehouse Basics (Parquet, ORC, Delta Lake)		
6.0	**06**	**Data Serving, Security & Future Trends**	**08**	
	6.1	**Serving Data for Analytics & ML**: Business vs operational vs embedded analytics, Data products concept, Semantic layers & metrics layers, Reverse ETL (operationalizing data)		**CO6**
	6.2	**Security & Privacy in Data Engineering**: Principle of least privilege, Encryption basics, Logging, monitoring, audits, Shared responsibility model in cloud		
	6.3	**Future of Data Engineering**: Shift towards real-time pipelines, Fusion of data + applications, Data mesh concepts, Rise of ML-integrated data systems, Live data stack and streaming-first architectures		
		Self-learning Topics: Modern OLAP Engines (DuckDB, ClickHouse, Druid, BigQuery), Data Quality Frameworks (Great Expectations, Soda, Monte Carlo)		
		Total	**39**	

Textbooks:

1. J. Reis and M. Housley, *Fundamentals of Data Engineering*. Sebastopol, CA: O'Reilly Media, 2022.
2. T. Akidau and S. Chernyak, *Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing*. Sebastopol, CA: O'Reilly Media, 2018.
3. A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Upper Saddle River, NJ: Prentice Hall/Pearson, 2007.
4. H. Garcia-Molina, J. D. Ullman, and J. Widom, *Database Systems: The Complete Book*, 2nd ed. Upper Saddle River, NJ: Pearson Education, 2008.
5. R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA: Pearson, 2017.

Reference books:

1. J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 3rd ed. Cambridge: Cambridge University Press, 2020.
2. B. Antony, *Professional Hadoop*. Indianapolis, IN: Wrox/Wiley, 2016.
3. A. P. More and P. Shah, *Apache Kafka Essentials*. Birmingham, UK: Packt Publishing, 2015.
4. M. Islam et al., *Kafka Streams Real-Time Processing*. Birmingham, UK: Packt Publishing, 2018.
5. P. Ponniah, *Data Warehousing Fundamentals for IT Professionals*, 2nd ed. Hoboken, NJ: Wiley, 2010.

Online References:

1. **Google Cloud – Data Engineering Learning Path**
   https://cloud.google.com/learn/training/data-engineering-and-analytics
2. AWS Big Data & Data Engineering Documentation (Kinesis, EMR, Glue, Redshift, S3)
   https://docs.aws.amazon.com/whitepapers/latest/big-data-analytics-options/what-is.html
3. Databricks Academy – Apache Spark Learning Resources
   https://www.databricks.com/learn/training
4. Snowflake University – Data Warehousing & ELT Workflows
   https://learn.snowflake.com/en/
5. Confluent Developer – Kafka Learning Portal & Certification Material
   https://developer.confluent.io/learn-kafka/
6. NPTEL – Database Management Systems (IIT Madras)
   https://nptel.ac.in/courses/106106093
7. NPTEL – Introduction to Data Analytics (IISc Bangalore)
   https://nptel.ac.in/courses/106108171

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

2. Question paper will comprise of 03 questions.
3. Question 1(20 marks) : - Solve any 04 out of 06. All questions carry 05 marks each.
4. Question 2(40 marks) : - Solve any 04 out of 06. All questions carry 10 marks each.
5. Question 3(20 marks) :- Solve any 04 out of 06. All questions carry 05 marks each.
6. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSC503	Data Warehousing and Mining	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIDSC503	Data Warehousing and Mining	20	20	60	--	--	100

Pre- requisite:

1. AIDSL302: Database Management System

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of solutions
4. PO4: Conduct investigations of complex problems
5. PO5: Modern Tool Usage
6. PO11: Life-long Learning

Course Objectives:

1. To identify the significance of Data Warehousing and Mining.
2. To develop understanding of various data preprocessing and data exploration techniques for knowledge discovery.
3. To analyze data and choose relevant classification algorithms for solving real-world problems.
4. To analyze data and choose relevant clustering algorithms for solving real-world problems.
5. To apply association rule mining in finding frequent patterns in various applications.
6. To apply web data mining in analyzing web structure.

Course Outcomes: Learners will be able to

1. Design data warehouse with dimensional modelling and apply OLAP operations.
2. Apply various data preprocessing and visualization techniques.
3. Classify real-world problems using appropriate classification algorithms.
4. Identify the need for clustering in real world scenarios and apply the same.
5. Analyze Association rules mining to identify frequent patterns.
6. Describe complex information and social networks with respect to web mining.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Data warehouse fundamentals and Online Analytical Processing**		
	1.1	Introduction to Data Warehouse, Data warehouse architecture, Data warehouse versus Data Marts	**08**	**CO1**
	1.2	Data Warehouse Schemas; Star Schema, Snowflake Schema, Factless Fact Table, Fact Constellation Schema. Update to the dimension tables. Major steps in ETL process		
	1.3	OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot.		
		Self-learning Topics: Cloud-based data warehouses		
2.0		**Introduction to Data Mining, Data Exploration and Data Pre-processing**		
	2.1	Data Mining Task Primitives, KDD process, Applications of Data Mining, Data Preprocessing: summarization, Cleaning, Data Integration & transformation	**08**	**CO2**
	2.2	Data reduction: Attribute subset selection, Histograms, Clustering and Sampling, Data Discretization: Normalization, Binning, Histogram Analysis and Concept hierarchy generation.		
	2.3	Statistical Description of Data, Data Exploration: Types of Attributes,		
	2.4	Proximity Measures for Nominal Attributes and Binary Attributes, Dissimilarity of Numeric Data		
		Self-learning Topics. Data Visualization, Explore BI tools.		
3.0		**Classification Basic Concepts**		
	3.1	Basic Concepts: Classification method, Decision Tree Induction, Attribute Selection Measures, ID3 algorithm	**08**	**CO3**
	3.2	Naïve Bayes classification, k-Nearest neighbor		
	3.3	Model evaluation and selection, Accuracy and Error measures, Precision, Recall, F1-score, specificity Confusion matrix, etc.		
	3.4	Evaluating the Accuracy of a Classifier: Holdout & Random Subsampling, Cross Validation, Bootstrap.		
		Self-learning Topics: Concept of Random Forest		
4.0		**Clustering Analysis**		
	4.1	Types of data in Cluster analysis, Partitioning Methods (k-Means, k-Medoids).	**05**	**CO4**
	4.2	Hierarchical Methods (Agglomerative, Divisive). Density based clustering: DBSCAN algorithm		
		Self-Learning: Advanced clustering algorithms BIRCH, CLIQUE		

5.0		**Association Rules mining**		
	5.1	Market Basket Analysis, Frequent Item sets, Closed Item sets, and Association Rule, Frequent Pattern Mining	**06**	**CO5**
	5.2	Apriori Algorithm, Association Rule Generation, Improving the Efficiency of Apriori, Hash based technique, Transaction Reduction, ECLAT		
	5.3	Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules.		
		Self-learning Topics: Constraint-based mining		
6.0		**Web Mining**		
	6.1	Introduction, Web Content Mining: Crawlers, Personalization	**04**	**CO6**
	6.2	Web Structure Mining: Page Rank, HITS or Clever		
	6.3	Web Usage Mining.		
	6.4	Text classification and categorization		
		Self-Learning: API-based data extraction		
		Total	**39**	

Textbooks:

1. Jiawei Han, Micheline Kamber, “Data mining: concepts and techniques", Morgan Kaufmann Publisher 2012, third edition, ISBN 978-0-12-381479-1
2. G. K. Gupta, “Introduction to Data mining with Case Studies", PHI Learning Private Limited, Delhi 2014, third edition, ISBN-978-81-203-5002
3. William H Inmon, “Building the data Warehouse”, Wiley Publication 2005, fourth edition, ISBN: 978-0-764- 59944-6.

Reference books:

1. Reema Theraja, “Data warehousing”, Oxford University Press 2009.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to Data Mining”, Pearson Publisher 2nd edition.
3. Ian H. Witten, Eibe Frank and Mark A. Hall, “Data Mining”, Morgan Kaufmann 3rd edition.

Online References:

1. Han and Camber PPT: https://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm
2. SWAYAM Course: https://onlinecourses.swayam2.ac.in/imb25_mg200/preview
3. Data Mining NPTEL Course: https://onlinecourses.nptel.ac.in/noc21_cs06/preview

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question 1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 5011	Internet of things	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIDSPEC 5011	Internet of things	20	20	60	--	--	100

Pre- requisite:

1. Basic Programming Language
2. Computer Organization and Architecture

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design /Development of Solutions
4. PO4: Conduct Investigations
5. PO6: The Engineer and the world
6. PO11: Life Long Learning

Course Objectives:

1. To introduce the fundamental concepts and architecture of the Internet of Things.
2. To introduce data handling methods and its tools for various IoT Applications.
3. To explore various Sensors and Actuators for any given Problem statement.
4. To introduce and understand IoT communication requirements across Physical, Datalink, Network and Application layers
5. To introduce the architecture of IoT systems, their unique constraints, and the importance of security and privacy in IoT environments.
6. To explore real-world IoT applications in domains such as smart homes, agriculture, and e-health

Course Outcomes: Learners will be able to

1. Summarize the fundamental concepts of Internet of Things.
2. Analyze various data handling methods in IoT-based systems using appropriate tools or Platforms.
3. Analyse physical, data link, and network layer protocols for IoT communication.
4. Analyse network layer and Application protocols for IoT communication
5. Explain IoT Security architecture, communication models and device characteristics. Also can identify fundamental security and privacy challenges.
6. Analyse real-world IoT applications in smart homes, agriculture, and e-health.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0	**1**	**Introduction to Internet of Things**	**6**	
	1.1	Introduction to IoT-Define IoT, Characteristics of IoT, Evolutionary phases of Internet, IoT and Digitization, IoT impact, Convergence of Information of Technology and Operational Technology, IoT challenges.		**CO1**
	1.2	IoT World Forum (IoTWF) Standardized Architecture, Drivers behind new Network Architecture, Comparing IoT Architectures, Core IoT functional Stack and IoT Data Management and Compute stack.		
	1.3	Sensors, Actuators and Smart Objects -Definition, Characteristics and Trends, Wireless Sensor Network and its Design constraints.		
		Self-learning Topics: Micro-Electro-Mechanical Systems (MEMS).		
2.0	**2**	**IoT Data collection and Analytics**	**6**	
	2.1	Introduction to IoT data collection, types and sources of IoT Data (Environmental data, Location data, Equipment data..), IoT Data collection mechanisms (Packet-based, Flow based, Log based), IoT Data collection Architecture.		**CO2**
	2.2	IoT Data Acquisition & Organization, Data Analytics in IoT, Trends and challenges in IoT Data Analytics.		
		Self-learning Topics: Data analytics tools such as Python with libraries like Pandas, NumPy, and Scikit-learn for data analysis and visualization. Learn about applying machine learning algorithms to IoT data for predictive analytics and anomaly detection.		
3.0	**3**	**IoT Protocols and Standards-Physical and Data link layer**	**8**	
	3.1	Overview of IoT protocol stack, role of Standard Organization in IoT (ITU, IETF, IEEE and 3GPP), Comparison of IoT and Traditional Internet Protocols.		**CO3**
	3.2	Physical and Data link Layer Protocols -IEEE 802.11(Wi-Fi), IEEE 802.15.4 (Zigbee,6LoWPAN), LoRa WAN, NB-IoT.		
		Self-learning Topics: Case studies on protocol selection and deployment in smart energy, logistics, and healthcare IoT.		
4.0	**4**	**IoT Protocols and Standards-Network and Application Layer**	**8**	
	4.1	Network and Transport Layer protocols-IPV6 for IoT,6LoWPAN adaptation, Transport protocols-TCP, UDP in IoT.		**CO4**
	4.2	Application Layer: MQTT, CoAP, DDS, HTTP & REST vs. SOAP.		

		Self-learning Topics: Routing protocols: Flat, hierarchical, location-based approaches, Energy-efficient routing protocols (LEACH, PEGASIS, geographic routing).		
5.0	**5**	**IoT Security & Privacy**	**6**	
	5.1	Introduction to IoT security, privacy concerns in IoT, overview of security challenges, Security across IoT communication layers.		**CO5**
	5.2	Privacy risks in IoT data collection, privacy-enhancing technologies (anonymization, pseudonymization).		
		Self-learning Topics: Develop a secure IoT healthcare monitoring system with encrypted communication and role-based access control.		
6.0	**6**	**IoT Applications**	**5**	
	6.1	IoT Applications Case studies are-Smart Home: Characteristics of Smart Home - Smart Home Energy Management, Smart Appliances, Communication Technologies for Smart Homes, maintenance, security, challenges.		**CO6**
	6.2	Smart Agriculture: characteristics and applications -Scarecrow, Smart Irrigation System, Crop Water Management, Integrated Pest Management, Sensor-based field and resource mapping, Remote equipment monitoring.		
	6.3	E-health: Characteristics of e-health and applications- monitoring of health parameters, smart medicine box, elderly people monitoring, challenges.		
		Self-learning Topics: Build and deploy a machine learning-based predictive maintenance model using IoT sensor data on an edge device.		
		Total	**39**	

Textbooks:

1. Designing the Internet of Things, 1st Edition, Arshdeep Bahga and Vijay Madisetti, University Press,2015
2. Internet of Things: Architecture and Design Principles,1st Edition, Raj Kamal, McGraw Hill Education,2017.
3. IoT Fundamentals Networking Technologies, Protocols and use cases for Internet of Things ,1st Edition, David Hanes, Gonzalo Salguerio, Pearson Education,2017.
4. Analytics for the Internet of Things (IoT),1st Edition, Andrew Minteer, Packt Publishing ,2017.
5. Hakima Chaouchi, "The Internet of Things - Connecting Objects to the Web", 1st Edition, Wiley, 2010.
6. Perry Lea, "Internet of things For Architects", 1st Edition, Packt Publication, 2018.Souvik Pal, Vicente García Díaz, and R. P. Mohanty, "IoT: Security and Privacy Paradigm", 1st Edition, CRC Press, 2022.

Reference books:

1. Adrian McEwen & Hakim Cassimally, "Designing the Internet of Things", 1st Edition, Wiley, 2014.
2. Donald Norris, "Raspberry Pi – Projects for the Evil Genius", 2nd Edition, McGraw Hill,2014.
3. Anand Tamboli ,"Build Your Own IoT Platform", 1st Edition, Apress, 2019.

Online References:

1. https://onlinecourses.nptel.ac.in/noc26_cs37/
2. https://onlinecourses.swayam2.ac.in/ntr24_ed44/
3. https://www.simplilearn.com/learn-iot-basics-skillup
4. https://www.coursera.org/learn/introduction-to-internet-of-things/
5. https://courses.mooc.fi/org/uh-cs/courses/introduction-to-the-internet-of-things-mooc

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks =15 marks for activities + 05 marks for attendance

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours is of 80 marks and scaled to 60.

1. Question paper will comprise 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 5012	Bitcoin and Cryptocurrency	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSPEC 5012	Bitcoin and Cryptocurrency	20	20	60	-	-	100

Pre- requisite:

1. Introduction to Cryptography

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO3: Design/Development of Solutions:
3. PO4: Conduct Investigations of Complex Problems:
4. PO8: Individual and Collaborative Team work
5. PO11: Life-Long Learning
6. PSO1: Problem solving

Course Objectives:

1. To get acquainted with the concept of Block and Blockchain.
2. To learn the concepts of consensus and mining in Blockchain.
3. To get familiar with the bitcoin currency and its history.
4. To understand and apply the concepts of keys, wallets, and transactions in the Bitcoin Network.
5. To acquire the knowledge of Bit coin network, nodes and their roles.
6. To analyze the applications& case studies of Blockchain.

Course Outcomes: Learners will be able to

1. Describe the basic concept of Block chain.
2. Associate knowledge of consensus and mining in Block chain.
3. Summarize the bit coin crypto currency at an abstract level.
4. Apply the concepts of keys, wallets and transactions in the Bit coin network.
5. Interpret the knowledge of Bit coin network, nodes and their roles.
6. Illustrate the applications of Block chain and analyze case studies.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction to Block chain**	**06**	
	1.1	Structure of a Block, Block Header, Block Identifiers: Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Block chain, Merkle Trees and Simplified Payment Verification (SPV).		**CO1**
		Self-learning Topics: Types of Blockchain		
2.0		**Consensus and Mining**	**07**	
	2.1	Decentralized Consensus, Byzantine General's Problem, Independent Verification of Transactions, Mining Nodes, Aggregating Transactions into Blocks, Constructing the Block header, Mining the Block, Successfully Mining the Block, validating a New Block, Assembling and Selecting Chains of Blocks, Block chain Forks		**CO2**
		Self-learning Topics. Other consensus algorithms Paxos and Raft		
3.0		**Introduction to Bit coin**	**04**	
	3.1	What is Bit coin and the history of Bit coin, Getting the first bit coin, finding the current price of bit coin and sending and receiving bit coin, Bit coin Transactions.		**CO3**
		Self-learning Topics: Other Cryptocurrencies, Ethereum, etc.		
4.0		**Concepts of Bit coin**	**10**	
	4.1	**Keys and addresses, Wallets and Transactions:** Public Key Cryptography and Crypto currency, Private and Public Keys, Bit coin Addresses, Base58 and Base58Check Encoding, Nondeterministic (Random) Wallets, Deterministic (Seeded) Wallets, HD Wallets (BIP-32/BIP- 44), Wallet Best Practices, Using a Bit coin Wallets, Transaction Outputs and Inputs, Transaction Fees, Transaction Scripts and Script Language, Turing Incompleteness, Stateless Verification, Script Construction (Lock + Unlock), Pay-to-Public-Key-Hash (P2PKH), Bitcoin Addresses, Balances, and Other Abstractions		**CO4**
		Self-learning Topics: Wallets technology for bitcoin, Security of wallet		
5.0		**Bit coin Networks**	**08**	
	5.1	Peer-to-Peer Network Architecture, Node Types and Roles, Incentive based Engineering the Extended Bitcoin Network, Bitcoin Relay Networks, Network Discovery, Full Nodes, Exchanging "Inventory", Simplified Payment Verification (SPV) Nodes, Bloom Filters, SPV Nodes and Privacy, Encrypted and Authenticated Connections, Transaction Pools		**CO5**

	Self-learning Topics: Blockchain Stack, web 2.0 and distributed ledger		
6.0	**Blockchain Applications & case studies**	**04**	
6.1	Domain-Specific Applications: FinTech, Internet of Things, Industrial and Manufacturing, Energy, Supply chain & Logistics, Records &. Identities, Healthcare, Case studies related to cryptocurrencies, Concept of Altcoin		**CO6**
	Self-learning Topics: Tokenization of real assets		
	Total	**39**	

Textbooks:

1. "Mastering Bitcoin, PROGRAMMING THE OPEN BLOCKCHAIN" , 3rd Edition by Andreas M. Antonopoulos, O'Reilly Media.
2. "Blockchain Applications: A Hands-On Approach", by Arshdeep Bahga, Vijay Madisetti, Paperback,1st Edition,2018.
3. "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Princeton University Press, 1st Edition,2016.

Reference books:

1. "Mastering Blockchain", by Imran Bashir, Third Edition, Packt Publishing, 3rd Edition,2020
2. "Mastering Ethereum: Building Smart Contracts and Dapps Paperback" by Andreas Antonopoulos, Gavin Wood, Publisher(s): O'Reilly Media., 1st Edition,2018.

Online References:

1. https://andersbrownworth.com/blockchain/
2. https://andersbrownworth.com/blockchain/Dublic-Drivate-kevs/
3. https ://www.coursera .org/learn/cryptocurrency
4. https://coinmarketcap.com/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 5013	Statistics for Artificial Intelligence & Data Science	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIDSPEC 5013	Statistics for Artificial Intelligence & Data Science	20	20	60	--	--	100

Pre-requisite:

AIDSC301-Applied Mathematics III, AIDSC401-Applied Mathematics IV

Program Outcomes addressed: PO1, PO2,PO4, PO6, PO11

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO4: Conduct Investigations of Complex Problems
4. PO6: PO6: The Engineer and The World
5. PO11: Life Long Learning

Course Objectives:

1. To understand key data types, data structures, and EDA techniques.
2. To apply sampling methods and compute sampling distributions & confidence intervals.
3. To analyse statistical experiments and perform hypothesis testing.
4. To evaluate parametric and non-parametric tests for different data scenarios.
5. To apply and interpret simple & multiple linear regression models.
6. To create basic time series models for trend, seasonality, and forecasting.

Course Outcomes:

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

1. Describe data structures, summary statistics, and EDA visuals.
2. Use sampling methods and compute SE, CI, and sampling distributions.
3. Analyse data using hypothesis tests.
4. Select and justify suitable statistical tests for given datasets.
5. Build and interpret regression models and check assumptions.
6. Develop time-series–based forecasting models using smoothing and decomposition.

Module No.	Unit No.	Topics	Hrs.	Course Outcome
1.0		**Exploratory Data Analysis**	**08**	
	1.1	Introduction to Data and Data Structures: Elements of Structured Data, Rectangular Data: Data Frames and Indexes), Non-Rectangular Data Structures, Estimates of Location (Mean, Median, Robust Estimates), and Estimates of Variability (Standard Deviation, Percentiles).		**CO1**
	1.2	Exploring and Visualizing Data Distributions: Percentiles and Boxplots, Frequency Tables and Histograms, Density Plots, Empirical Cumulative Distribution Function (ECDF), Quantile–Quantile (Q–Q) Plots, and Stem-and-Leaf Plots, Pie Charts, Pareto Charts		
	1.3	Exploring Relationships Between Variables: Exploring Two or More Variables, Correlation and Scatterplots, Hexagonal Binning and Contours (Numeric vs. Numeric Data), Categorical vs. Numeric Data, and Visualizing Multiple Variables.		
		Self-Learning: Comparing Summary Statistics using Bootstrap Methods, Outlier Detection and its Impact on Data Visualization		
2.0		**Data and Sampling Distributions, Probability**	**06**	
	2.1	Sampling Methods and Principles: Random Sampling and Sample Bias, Random Selection, Size Versus Quality, Sampling from Finite and Infinite Populations, Other Sampling Methods (Stratified Random Sampling, Cluster Sampling, Systematic Sampling, Convenience Sampling, Judgment Sampling), Sample Mean Versus Population Mean, Selection Bias, and Regression to the Mean.		**CO2**
		Fundamentals of Probability: Addition and Multiplication Theorems, Conditional Probability, Marginal and Joint Probability, Independent and Dependent Events Discrete Probability Distributions: Random Variables, Discrete Probability Distributions, Expected Value and Variance, Binomial Probability Distribution, Poisson Probability Distribution; Continuous Probability Distributions: Uniform Probability Distribution, Normal Curve, Standard Normal Probability Distribution		
	2.2	Sampling Distributions and Estimation: Sampling Distribution of a Statistic, Central Limit Theorem, Standard Error, Bootstrap and Resampling Techniques, Confidence Intervals, Interval Estimation (Population Mean – Known and Unknown Variance, Population Proportion), Determining Sample Size		
		Self-Learning: Applications of the Central Limit Theorem in Real-World Data, Comparing Sampling Methods for Accuracy and Efficiency, Naive Bayes Theorem		

3.0		**Statistical Experiments and Hypothesis Testing**	**06**	
	3.1	Statistical Experiments and A/B Testing: A/B Testing, Design of Statistical Experiments, Random Sampling and Randomization, Control and Treatment Groups, Web Stickiness Example, Significance Testing, Null and Alternative Hypotheses, One-Way vs Two-Way Tests, Parametric vs Non-Parametric Tests, Statistical Significance and p-Values, Type I and Type II Errors, Significance Level (α)		**CO3**
	3.2	Resampling and Inference on Population Parameters: Resampling Concepts, Bootstrap and Permutation Tests, Exhaustive vs Bootstrap Permutation Testing, Inference about Means and Proportions (Single and Two Populations), Inference about Population Variance and Comparison of Two Population Variances.		
		Self-Learning: Applications of Hypothesis Testing in Real-World A/B Experiments, Understanding the Role of Resampling Techniques (Bootstrap & Permutation) in Data Science		
4.0		**Parametric and Non-Parametric Tests**	**08**	
	4.1	Parametric Tests: Introduction to Parametric Tests and their assumptions, t-Test: One-sample, Independent two- sample, and Paired t-test; z-Test: For mean and proportion; F-Test: For equality of variances; Chi-Square Test: Goodness of fit and Test for Independence; ANOVA (Analysis of Variance): One-way and Two-way ANOVA — Concept, Assumptions, and Interpretation		**CO4**
	4.2	Non-Parametric Tests: Introduction to Non-Parametric Tests and when to use them, Sign Test, Wilcoxon Signed- Rank Test, Mann–Whitney U Test (Wilcoxon Rank-Sum Test), Kruskal–Wallis Test, Friedman Test, Rank Correlation (Spearman's ρ, Kendall's τ), Comparison between Parametric and Non-Parametric Tests		
		Self-Learning Topics: Real-World Applications of Parametric and Non-Parametric Tests in Business Analytics		
5.0		**Simple and Multiple Regression**	**08**	
	5.1	Simple Linear Regression: Concept and Applications of Regression; Simple Linear Regression Model and Regression Equation; Estimated Regression Equation using the Least Squares Method; Coefficient of Determination (R^2) and Correlation Coefficient (r); Adjusted R^2 — Concept and Need; Testing for Significance: t-test for Regression Coefficients		**CO5**
	5.2	Multiple Linear Regression: Multiple Regression Model and Least Squares Estimation; Interpretation of Regression Coefficients; Multiple Coefficient of Determination (R^2 and Adjusted R^2); Model Assumptions Linearity, Independence, Homoscedasticity, Normality; Testing for Significance: t-test and F-test for Model Validation		

		Self-Learning Topics: Multicollinearity and its Detection; Model Selection Techniques (Forward, Backward, Stepwise Regression)		
6.0		**Time Series Analysis and Forecasting**	**03**	
	6.1	Introduction to Time Series Analysis: Definition and Importance of Time Series Data; Components of Time Series: Trend, Seasonality, Cyclic, and Irregular Variations; Time Series Patterns, Autocorrelation Function (ACF): Concept, interpretation, identifying persistence, trend and seasonality		**CO6**
	6.2	Forecasting Techniques: Forecast Accuracy: Mean Absolute Deviation(MAD), Mean Squared Error(MSE), Mean Absolute Percentage Error (MAPE); Smoothing and Trend Methods: Moving Averages, Exponential Smoothing, Trend Projection, Seasonality analysis. Time Series Decomposition: Separation of Trend and Seasonal components.		
	6.3	Overview of Box-Jenkins Methodology: Identification, Estimation, and Diagnostics. ARMA and ARIMA Models: Autoregressive models, Moving Average models, ARMA/ARIMA formulation, autocorrelation (ACF) and partial autocorrelation (PACF) for model selection. Building and Evaluating ARIMA Models: Model fitting, residual analysis and diagnostics; AIC/BIC criteria; model validation; practical reasons to choose ARIMA and cautions. Dynamic and Seasonal Models: Dynamic ARIMA and SARIMA models, online model updates for changing data patterns, adaptive forecasting, and real-time prediction techniques.		
		Self-Learning Topics: Introduction to ARIMA Models for Forecasting, Real-world Applications of Time Series Forecasting in Business and Finance		
		Total	**39**	

Textbooks:

1. Bruce, Peter, Andrew Bruce, and Peter Gedeck. “Practical Statistics for Data Scientists: 50 Essential Concepts”, O’Reilly Media, 2017.
2. Anderson, David R., Dennis J. Sweeney, Thomas A. Williams, Jeffrey D. Camm, and James J. Cochran, “Statistics for Business and Economics”, 13th Edition, Cengage Learning, 2018.
3. Grus, Joel, “Data Science from Scratch: First Principles with Python”, O’Reilly Media, 2015. (1st Edition)
4. Nielsen, Aileen. “Practical Time Series Analysis: Prediction with Statistics and Machine Learning”,O’Reilly Media, 2019.
5. James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani,”An Introduction to Statistical Learning: With Applications in R”, Springer, 2013. (1st Edition)

Reference books:

1. Dodge, Yadolah (Ed.),”Statistical Data Analysis and Inference”, Elsevier, 2014.
2. Ismay, Chester, and Albert Y. Kim, “Statistical Inference via Data Science: A Modern Dive into R and the Tidyverse”,CRC Press, 2019.
3. Milton, J. S., and Arnold, J. C., “Introduction to Probability and Statistics”, Tata McGraw-Hill, 4th Edition, 2007.
4. Devore, Jay L., “Probability and Statistics for Engineering and the Sciences. Cengage

Publications", 9th Edition, 2015
5. Johnson, R. A., and Gupta, C. B. Miller and Freund's, "Probability and Statistics for Engineers", Pearson Education Asia, 7th Edition, 2007.
6. Chandrasekaran, A., and Kavitha, G. "Probability, Statistics, Random Processes and Queuing Theory", Dhanam Publications, 2014.

Online References:

1. https://onlinecourses.swayam2.ac.in/ini25_ma01/preview
2. https://onlinecourses.nptel.ac.in/noc21_ma74/preview
3. https://onlinecourses.nptel.ac.in/noc26_cs07/preview
4. https://onlinecourses.nptel.ac.in/noc24_ma19/preview
5. https://www.coursera.org/learn/statistical-inference-and-hypothesis-testing-in-data-science-applications]
6. https://www.coursera.org/learn/practical-time-series-analysis

Course Assessment:

ISE: To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks aaccording to subject scheme

MSE:: To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 3 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 5014	Computer Graphics and Game theory	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIDSPEC 5014	Computer Graphics and Game theory	20	20	60	-	-	100

Pre- requisite:

1. Basic Mathematics & C-Programming
2. Probability

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO3: Design/Development of Solutions:
3. PO4: Conduct Investigations of Complex Problems:
4. PO8: Individual and Collaborative Team work
5. PSO1: Problem solving

Course Objectives:

1. To equip students with the fundamental knowledge and basic technical competence in the field of Computer Graphics.
2. The course will also acquaint the student with algorithms for generating and rendering graphical models, mathematics for geometrical transformations.
3. The course will also enable students to apply various techniques of projections, shading, illumination and lighting to graphical models.
4. To make the students to understand various components in game theory.
5. To analyze various models in game theory for practical applications.

Course Outcomes: Learners will be able to

1. Illustrate various algorithms to generate lines, circles, curves, fractals, and polygons and colour them.
2. Apply 2D and 3D Transformations, viewing, and projections on a given object.
3. Explore curve representation techniques and Fractal Generation methods.
4. Illustrate the concept of shading, hidden surface elimination and Animation.
5. Analyze the basics of games and strategies.
6. Analyze various models of game theory.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction to Computer graphics and Output primitives**	**06**	
	1.1	Introduction: Bitmap and Vector based graphics, Overview of Coordinate System.	**8**	**CO1**
	1.2	Scan Conversion of: point, line using Digital Differential analyzer & Bresenham's algorithm, circle using midpoint approach and Bresenham.		
	1.3	Polygons: Concave, Convex, Inside/Outside Test Area Filling: Scan line Polygon Fill Algorithm, Boundary Fill and Flood Fill algorithm		
	1.4	Introduction to GPU Rendering Pipeline		
		Self-learning Topics: Pixel, resolution, aspect ratio, frame buffer, refresh rates, Display Devices, Properties of Light.		
2.0		**Two Dimensional and 3D Transformations and Projections**	**10**	
	2.1	Basic Geometrical 2D Transformations: Translation, Rotation, Scaling, Reflection, Shear, homogeneous Matrix representation and Composite transformation. Point Clipping, Line clipping: Cohen Sutherland Algorithm, Liang Barsky Algorithms. Polygon Clipping: Sutherland Hodgeman Polygon Clipping and Weiler Atherton, Text Clipping.	**08**	**CO2**
	2.2	Three Dimensional transformations: Translation, Rotation, Scaling, Rotation about an arbitrary axis Three-Dimensional Viewing Pipeline, Viewing Transformation, Projections: Parallel (Oblique and Orthographic), Perspective.		
		Self-learning Topics: Viewing Pipeline**,** View Coordinate reference frame, Window to Viewport Transformation**.**		
3.0		**Curves and Fractal Generation**	**05**	
	3.1	Curves: Introduction to curves, interpolation and approximation, Blending Function, Bezier and B-spline curves		**CO3**
	3.2	Fractals: Introduction, Classification, Fractal Generation- Snowflake, Sierpinski Gasket, Koch Curve, Cantor Middle-Thirds Set, Hilbert Curve.		
		Self-learning Topics: Application of curves and fractal in data visualization.		
4.0		**Visible Surface Detection and Animation**	**04**	
	4.1	Visible Surface Detection: Classification of Visible		**CO4**

		Detection algorithm, Back Surface detection method, Depth Buffer method, Area Subdivision method		
	4.2	Animation: Introduction to Animation, Traditional Animation Techniques, Principles of Animation, Key framing: Character and Facial Animation, Deformation,		
	4.3	AI-Assisted Animation (motion prediction, retargeting & automated clean-up)		
		Self-learning Topics: Motion tween, Morphing, Warping.		
5.0		**Introduction To Game Theory**	**06**	
	5.1	Basics of Games - strategy preferences - payoffs, rational Choice, representation of game. Basic solution concepts non-cooperative versus cooperative games.		**CO5**
	5.2	Strategic Games - Prisoner's Dilemma, Nash equilibria.		
	5.3	Strategy Types- Pure, mixed and optimal, finding value of game and saddle point.		
		Self-learning Topics: Bayesian Game theory and example		
6.0		**Game Theory in Practice**	**08**	
	6.1	Cournot's and Bertrand models of oligopoly, Demand and supply with profit		**CO6**
	6.2	Zero Sum games: Strategy and applications		
	6.3	Games with perfect and imperfect information		
	6.4	Auctions and bidding- open and closed auctions, sealed auctions bidding types		
	6.5	Bargaining in game theory, Repeated games		
		Self-learning Topics: Ebay auctions and VCG model		
		Total	**39**	

Textbooks:

1. Hearn & Baker, "Computer Graphics C version", 2nd Edition, Pearson Publication
2. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice in C", 2ndEdition, Pearson Publication.
3. Samit Bhattacharya, "Computer Graphics", Oxford Publication.
4. An introduction to Game theory by Martin J. Obsorne.

Reference books:

1. D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publications.
2. Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum"s Outlines McGraw-Hill Education.
3. Rajesh K. Maurya, "Computer Graphics", Wiley India Publication.
4. F. S. Hill, "Computer Graphics using OpenGL", Third edition, Pearson Publications.
5. M.Machler, E. Solan, S. Zameer, Game theory, Cambridge university press.

Online References:

1. https://www.classcentral.com/course/interactivegraphics-2067
2. https://swayam.gov.in/nd2_ntr20_ed15/preview
3. https://nptel.ac.in/courses/106/106/106106090/
4. https://www.edx.org/course/computer-graphics-2

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC5032	Embedded Systems and RTOS	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC5032	Embedded Systems and RTOS	20	20	60	--	--	100

Pre- requisite:

1. FEC204: Digital System Design
2. FEC104: C programming
3. MDMC4031: Microprocessors & Microcontrollers

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO6: The Engineer and The World
5. PO7: Ethics
6. PO11: Life-Long Learning

Course Objectives:

1. Interpret embedded system fundamentals, classifications, design metrics, and real-world applications.
2. Explore embedded processing cores, memory architectures, and ARM Cortex-M3 processor features.
3. Interface sensors and actuators using standard communication protocols and implement low-power, battery- aware embedded designs.
4. Comprehend safety and security standards including IEC 61508, IEC 60601, ISO 26262, and ISO 21434 for cybersecurity in embedded systems.
5. Apply RTOS principles, including task management, inter-process communication, memory management, and unit testing in embedded applications.
6. Implement real-time scheduling algorithms (RMS, EDF) and develop RTOS-based embedded applications.

Course Outcomes: Learners will be able to

1. Explain embedded system fundamentals, classifications, design metrics, and applications in real-world scenarios.
2. Describe embedded processing cores, memory types, and ARM Cortex-M3 architecture for system design. Interface sensors and actuators using communication protocols (UART, SPI, I2C) and apply low-power and battery-aware design techniques.
4. Analyze and compare IEC 61508, IEC 60601, ISO 26262 and evaluate awareness of cybersecurity needs in embedded systems.
5. Apply task management, inter-process communication, and memory management mechanisms in an RTOS environment.
6. Apply real-time scheduling algorithms to analyze system schedulability and develop practical RTOS-based embedded applications.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Embedded Systems Fundamentals**	**04**	
	1.1	Definition, Characteristics, Classification, Applications		**CO1**
	1.2	Design metrics of Embedded system (performance, power, cost, size)		
	1.3	Challenges and design trade-offs		
		Self-learning Topics: 8-bit vs 32-bit MCU comparison		
2.0		**Embedded Systems Hardware & Programming**	**09**	
	2.1	Embedded system architecture overview (MCU, SoC, SoC components)		**CO2**
	2.2	Basics of ARM Cortex-M3 processor		
	2.3	Memory types in embedded systems (Flash, SRAM, EEPROM, Cache)		
	2.4	Embedded C Concepts: Bitwise operations for port control, GPIO programming fundamentals, Polling vs Interrupt handling in C, Writing delay routines, LED blinking, switch interfacing		
		Self-learning Topics. Datasheet exploration exercise		
3.0		**Communication & Low Power Systems**	**07**	
	3.1	UART, I2C, SPI basics		**CO3**
	3.2	Wireless Interface overview: BLE / Wi-Fi		
	3.3	Sensor and Actuator basics with examples (temp, motor)		
	3.4	Low-power concepts (Sleep modes, duty cycling)		
	3.5	Battery and Power Management: Common battery types used in embedded systems, Power-saving modes (sleep/idle modes) in embedded systems		
		Self-learning Topics: ADC programming example		
4.0		**Industry Standards and Secure Coding**	**05**	
	4.1	Overview of Standards: IEC 61508 (Functional Safety), IEC 60601 (Medical Electrical Equipment), ISO 26262		**CO4**
		(Automotive Safety)		
	4.2	Basic lifecycle steps (Requirement → Development → Testing→ Verification & Validation)		
	4.3	Fundamentals of secure coding, common vulnerabilities, and unit testing		
		Self-learning Topics: Safety Critical Software Engineering under Functional Safety Standards		
5.0		**RTOS Concepts & Programming**	**09**	
	5.1	Concepts: Task, Thread, Multitasking, Latency		
	5.2	Inter-task communication: Semaphore, Mutex, Queue		
	5.3	Task scheduling basics: Static vs Dynamic scheduling		

	5.4	FreeRTOS programming fundamentals: Task creation, Delay & periodic tasks, Queue send/receive, Semaphore to avoid race conditions		**CO5**
		Self-learning Topics: RTOS-Based Application Design: Smart Home, Automatic Fan Controller, Motor speed control		
6.0		**Real Time Scheduling and Analysis**	**05**	
	6.1	Scheduling: Rate Monotonic Scheduling (RMS) & Earliest Deadline First Scheduling (EDF)		**CO6**
	6.2	Schedulability formula + simple numerical problems		
		Self-learning Topics: Additional real-time scheduling algorithms		
		Total	**39**	

Textbooks:

1. K. V. K. Prasad, Embedded or Real-Time Systems: Concepts, Design and Programming Black Book. New Delhi, India: Dreamtech Press, 2003.
2. R. Mall, Real-Time Systems: Theory and Practice. New Delhi, India: Prentice Hall, 2009.
3. Rajkamal, Embedded System: Architecture, Programming and Design. New Delhi, India: Tata McGraw-Hill, 2011.
4. J. Yiu, The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, 3rd ed. Oxford, UK: Elsevier, 2014.
5. P. Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems and IoT, 4th ed. Cham, Switzerland: Springer, 2021.

Reference books:

1. D. Simon, An Embedded Software Primer. New Delhi, India: Pearson, 2009.
2. J. W. Valvano, Embedded Microcomputer Systems: Real-Time Interfacing, 3rd ed. Boston, MA, USA: Cengage Learning, 2012.
3. B. Amos, Mastering FreeRTOS: Hands-On Real-Time Operating Systems. Birmingham, U.K.: Packt Publishing, 2020.
4. X. Fan, Real-Time Embedded Systems: Design Principles and Engineering Practices. Oxford, U.K.: Newnes (Elsevier), 2022.

Online References:

1. MOOC Course (NPTEL): https://onlinecourses.nptel.ac.in/noc20_cs16/preview
2. Real Time Operating System by Rajib Mall (IIT Kharagpur): https://onlinecourses.nptel.ac.in/noc25_cs78/preview
3. Embedded Systems (Real-Time Embedded Systems, IIT Kharagpur): https://nptel.ac.in/courses/108105057
4. Amazon Web Services, *FreeRTOS Official Documentation.* [Online]. Available: https://www.freertos.org

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question 1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC 5042	Cryptography & Network Security	03	-	-	03	-	-	03

Corse Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
MDMC 5042	Cryptography & Network Security	20	20	60	--	--	100

Pre-requisite:

1. CEC303: Discrete Structure and Graph Theory
2. CEC503: Computer Networks

Program Outcome mapped:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of solutions
4. PO6: The Engineer and The World
5. PO7: Ethics
6. PO11: Life-Long Learning

Course Objectives:

The course aims to provide students with:

1. A basic understanding of security concepts, classical cryptographic techniques, and the need for secure communication.
2. Knowledge of symmetric key cryptography, mathematical foundations, and modern block cipher algorithms.
3. Knowledge of asymmetric key cryptography, major public-key algorithms, and key management mechanisms.
4. Understanding of cryptographic hash functions, digital certificates, and authentication protocols.
5. Insight into access control mechanisms, formal security models, and Network Access Control (NAC) implementations.
6. The ability to analyze network security threats, DoS attacks, and Internet security protocols with practical relevance.

Course Outcomes: Learners will be able to

1. Apply basic security principles and classical cryptographic techniques to secure communication.
2. Explain and apply mathematical foundations and symmetric key cryptographic algorithms.
3. Apply asymmetric key cryptographic techniques and key management mechanisms.
4. Analyze cryptographic hash functions, digital certificates, and authentication protocols.
5. Evaluate access control models and assess NAC enforcement techniques.
6. Analyze network security threats, DoS attack mechanisms, and major Internet security protocols.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcomes
0		**Prerequisite**	**01**	
	0.1	Basic understanding of computer networks, operating systems, CIA basics		**--**
1		**Introduction to Security Concepts and Basic Cryptography**	**07**	
	1.1	Security Goals, Services, Mechanisms and Types of Security Attacks		**CO1**
	1.2	The OSI security architecture, Network security model		
	1.3	Concept of Cryptography, Classical Encryption techniques: mono- alphabetic and poly-alphabetic		
	1.4	Substitution Techniques: Caesar Cipher, Multiplicative Cipher, Affine Cipher, Autokey Cipher, Vigenère Cipher, Playfair Cipher, Hill Cipher		
	1.5	Transposition Techniques: Keyed and Keyless Transposition Ciphers, Introduction to Steganography		
		Self-Learning Topics: Homomorphic Encryption, Basics of Quantum Cryptography		
2		**Symmetric Key Cryptography**	**07**	
	2.1	Euclidean algorithm, Modular Arithmetic, Prime Numbers, Fermat's Theorem, Congruence Relation, Chienese Remainder Theorem		**CO2**
	2.2	Block Cipher Principles, Block Cipher Modes of Operation, DES, Double DES, Triple DES, Advanced Encryption Standard (AES), Stream Ciphers: RC5 algorithm		
		Self-Learning Topics: Basics of Linear cryptanalysis and Differential Cryptanalysis		
3		**Asymmetric Key Cryptography and Key Management**	**06**	
	3.1	Public key cryptography, Principles of public key cryptosystems, The RSA algorithm, The Knapsack algorithm,		**CO3**
	3.2	Key management Techniques: Using Symmetric and Asymmetric Algorithms and Trusted Third Party, Diffie Hellman Key exchange Algorithm		
		Self-Learning Topics: Elliptic Curve Cryptography (ECC), Basics of Lattice-based Cryptography		
4		**Cryptographic Hash Functions, Digital Certificates and Authentication Protocols**	**09**	
	4.1	Cryptographic Hash Functions, Properties of Secure Hash Function, MAC, HMAC, CMAC, MD5, SHA-256 (SHA-2 family) and SHA-		**CO4**
	4.2	Digital Certificate: X.509, PKI, Digital Signature Schemes: RSA, EIGamal		
	4.3	User Authentication and Entity Authentication, Needham Schroeder Authentication Protocol, Kerberos Authentication		
		Self-Learning Topics: SHA-512, Schnorr Signature Schemes		
5		**Access Control**	**03**	

	5.1	Access Control Lists & Access Control Models: Bell–LaPadula Model, Biba Model, Clark–Wilson Model		**CO5**
	5.2	NAC: Principle Elements of NAC, Principle NAC Enforcement Methods, How to Implement NAC Solutions		
		Self-Learning Topics: Basics of Covert Channels		
6		**Network Security**	**06**	
	6.1	Network security basics: TCP/IP vulnerabilities (Layer wise), Packet Sniffing, ARP spoofing, port scanning, IP spoofing, TCP SYN flood, DNS Spoofing		**CO6**
	6.2	Denial of Service: Classic DOS attacks, Source Address spoofing, ICMP flood, SYN flood, UDP flood		
	6.3	Internet Security Protocols: SSL, IPSEC, Secure Email: PGP, Firewalls, IDS and types		
		Self-Learning Topics: Types of malware, DoS Mitigation Techniques, Virtual Private Network (VPN)		
		Total	**39**	

Textbooks

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Pearson, 2024.
2. B. A. Forouzan, Cryptography and Network Security, Revised ed. New Delhi, India: Tata McGraw-Hill, 2011.
3. B. L. Menezes and R. Kumar, Cryptography, Network Security, and Cyber Laws, 1st ed. New Delhi, India: Cengage Learning, 2018.
4. E. Cole, Network Security Bible, 2nd ed. Hoboken, NJ, USA: Wiley, Mar. 2011.
5. W. Stallings and L. Brown, Computer Security: Principles and Practice, 4th ed. Boston, MA, USA: Pearson, 2018.

Reference Books

1. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 20th Anniversary ed. Hoboken, NJ, USA: Wiley, 2015.
2. A. Kahate, Cryptography and Network Security, 4th ed. New Delhi, India: McGraw-Hill Education, 2022.
3. C. Paar, J. Pelzl, and T. Güneysu, Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms, 2nd ed. Cham, Switzerland: Springer, 2024.

Online References

- https://www.geeksforgeeks.org/cryptography/
- https://onlinecourses.nptel.ac.in/noc22_cs90/preview
- https://www.coursera.org/learn/crypto
- https://www.coursera.org/learn/mathematical-foundations-cryptography
- https://www.tutorialspoint.com/cryptography
- https://cybersecurityguide.org/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE: To be conducted as a written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question 1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDM C5062	Supply Chain Management	03		01	03		01	04

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
MDM C5062	Supply Chain Management	20	20	60	25	------	125

Pre- requisite:

1. Basic MS Excel.

Program Outcomes

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO7: Global and Ethical Practices
4. PO8: Collaborative Teamwork
5. PO9: Communication
6. PO11: Lifelong Learning

Course Objectives:

1. To introduce the fundamental concepts, components, and processes of supply chain management in manufacturing and service sectors.
2. To develop understanding of demand forecasting, inventory management, procurement, and logistics as core elements of SCM.
3. To equip students with the ability to analyze supply chain networks, evaluate performance, and identify improvement opportunities.
4. To familiarize students with supply chain technologies such as ERP systems, RFID, IoT, and analytics tools.
5. To build decision-making skills related to sourcing, production planning, transportation, and distribution strategies.
6. To promote awareness of sustainability, risk management, and global challenges in supply chain operations.

Course Outcomes: Learners will be able

1. To explain key supply chain, logistics, and international shipping concepts.
2. To apply analytical tools to solve basic warehousing, transportation, and network design problems.
3. To analyse supply chain flows and identify performance gaps, risks, and integration challenges.
4. To evaluate supply chain efficiency using cost, service, sustainability, and responsiveness metrics.
5. To design technology-enabled and resilient supply chain solutions for domestic and global environments.
6. To demonstrate understanding of sustainable, ethical, and technology-enabled supply chain practices and assess their impact on organizational performance

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0	**1**	**Module 1: Introduction to Supply Chain, Logistics and International Shipping**	**09**	
		Introduction to Supply Chain, Decision Phases in a Supply Chain, Linkages and Decisions in SCM. Process and Cycle View of a Supply Chain. Supply Chain Models: Efficient Chain, Responsive Supply Chain and Agile Models. Importance of supply chain flows. Logistics and Shipping: functions. Objectives, Goals, Decisions. 1st Party, 2nd Party, 3rd Party, 4th Party Logistic service provider Introduction to International Shipping: Freight Forwarders, shipping documents, Ports and customs, Incoterms		**CO1, CO2**
		Self-learning Topics: Importance of supply chain flows. Logistics and Shipping: functions. Objectives, Goals, Decisions		
2.0	**2**	**Warehousing and Transportation**	**09**	
		Warehouse: role, type, functions. Warehouse location modelling: methods such as COG, MCOG, and the LP approach. Layout design, Space calculation, Warehouse automation, Hub and Spoke Model, WMS Distribution, Role, Importance, Levels, Channels, Structure, Functions. Channel partners, functions. Infrastructure, road, rail, air, water, pipeline. Freight Management, Freight cost. Transportation Network Route Planning, Containerization, and Packing. Effective / Cost Optimising		**CO2, CO3**
		Self-learning Topics: Role, Importance, Levels, Channels, Structure, Functions. Channel partners,		
3.0		**Performance Measurement and Controls in Supply Chain Management**	**03**	
		Supply Chain Cost Calculations' Supply Chain Reliability & Responsiveness Supply Chain Asset Management: cash to cash cycle, return on working capital & working capital Analytics Framework Based on SCOR Model		**CO3, CO4**
		Self-learning Topics: Concepts of Supply Chain Asset Management: cash to cash cycle & working capital		
4.0		**Designing a network in a supply chain**	**06**	
		Network design in the supply chain Alternate channels in distribution Location decisions in the supply chain using Excel Role and factors affecting network designing Network Optimisation Model Network design in an uncertain environment Flexibility in Supply Chain		**CO2, CO4 CO5**

		Self-learning Topics: Role and factors affecting network designing		
5.0		**Supply Chain Integration and Designing Global Supply Chain Networks**	**06**	
		Bullwhip effect, Collaborative Planning Forecasting Replenishment (CPRF) concept. Inventory Management and Risk Pooling, Strategic Alliances, Retailer-Supplier Partnerships: Types, Requirements, and Inventory Ownership. Outsourcing and related decisions Global Market: Technological/ Cost/ Political, and Economic Forces. Risks and Advantages of the International Supply Chain. International versus Regional products. Local autonomy versus central control. Regional differences in Logistics- Cultural differences/		**CO3, CO5**
		Self-learning Topics: Concepts related to Outsourcing and related decisions, Risks and Advantages of the International Supply		
6.0		**Ethical issues in Supply Chain and Technological Trends in Supply Chain**	**06**	
		Supply chain vulnerability. Conformance to applicable laws such as Contract and commercial laws, Trade regulation, government procurement regulations, patents, Copyrights, trademark laws, transportation and logistics laws and regulations Environmental laws. International practices. Confidentiality and proprietary information. Functional application of technologies in Supply Chain: Blockchain, AI, IOT, RFID, AR, VR, ERP Goldratt Supply Chains, Sustainable Supply Chain, Resilient supply chains, Green Supply chain, Lean supply chain.		**CO4, CO5**
		Self-learning Topics: Concepts of Blockchain, AI, IOT, RFID, AR, VR, ERP		
		Tutorial on application of supply chain management concepts in excel		
		Total		**39**

Textbooks

1. Supply Chain Management: Strategy, Planning, and Operation, 6th Edition, Sunil Chopra & Peter Meindl, Pearson Education

Reference books:

1. Supply Chain Analytics and Modelling: Quantitative Tools and Applications, 1st Edition, Kogan Page
2. Supply Chain Logistics Management, 2nd Edition, Bowersox, Donald J, 2014, McGraw Hills.
3. Supply Chain Management: The Basics & Beyond, Copacino, William C, 1997, ST. Lucie Press.
4. Logistics and Supply Chain Management, Martin Christopher

Online References:

1. https://www.emerald.com/ijlm
2. https://www.emeraldgrouppublishing.com/journal/scm

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question 1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL501	Artificial Intelligence Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSL501	Artificial Intelligence Lab	--	--	--	25	25	50

Pre-requisite:
FEC104 C- Programming
CSL405 Python Programming

Program Outcomes addressed:
1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO4:Conduct Investigations of Complex Problems
5. PO5:Engineering Tool Usage
6. PO6: The Engineer and the world

Lab Objectives:
1. To explore AI programming basics
2. To implement search strategies
3. To apply logic and reasoning
4. To apply Probabilistic models
5. To develop planning solutions
6. To work with modern AI tools

Lab Outcomes:
1. To implement core AI algorithms.
2. Apply search-based problem solving.
3. To apply logical reasoning methods.
4. To implement probabilistic classifiers.
5. To build planning Solutions
6. To Analyze AI tools and applications.

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
1	Route Planning with Uninformed Search: A city wants to compute emergency-vehicle routes between junctions without using any heuristics. Key tasks: • Model the road map as a graph using adjacency lists. • Implement BFS, DFS, Depth-Limited, and Iterative Deepening Search. • visualize traversal, Compare path cost and node expansions. Tools: Python, NetworkX (optional), Matplotlib for simple graph visualization.	**LO1**
2	*A* Search for Intelligent Navigation: A cab-aggregation service wants optimal routes using heuristic-guided search. Key tasks: • Implement A* search with admissible heuristics (Euclidean/Manhattan). • Compare Greedy best-first vs A*. • Demonstrate effect of heuristic admissibility and consistency. Tools: Python, NumPy.	**LO2**
3	8-Puzzle Solver: Sliding tile puzzle game app for mobile entertainment. Tasks: Represent state space; solve using BFS/A*; compute moves and time complexity. Tools: Python lists/tuples.	**LO2**
4	Water Jug Problem: Scenario: Measure exact water amounts using limited jugs in a survival game. Tasks: Model as a state space search; solve with BFS; find minimal steps. Tools: Python queues, sets for visited states	**LO2**
5	Optimization Using Hill Climbing and Simulated Annealing: Solve an optimization problem (such as the Travelling Salesman Problem) using local search techniques and analyze how stochasticity helps escape local minima. Key Tasks: • Represent TSP cities and the distance matrix. • Implement Hill Climbing for TSP. • Implement Simulated Annealing for the same dataset. • Compare convergence, path cost, and computation time.	**LO3**
6	Game Playing with Minimax & Alpha–Beta Pruning: A gaming startup needs an AI opponent for Tic-Tac-Toe. Key tasks: • Implement Minimax on a simple game tree. • Add Alpha–Beta pruning. • Compare node-expansion counts and decision quality. Tools: Python, GUI	**LO3**
7	Designing a Knowledge Base for Medical Diagnosis: A clinic needs a rule-based triage assistant for flu symptoms. Key tasks:	**LO3**

	• Encode knowledge in propositional logic / Horn clauses. • Implement forward and backward chaining. • Validate inferences on given patient profiles. Tools: Python; simple inference engine using lists/dicts; optional pyDatalog.	
8	First-Order Logic Inference: An Expert system diagnosing plant diseases from symptoms. Key Tasks: Convert facts/rules to FOL; implement unification/resolution; query conclusions. Tools: Python dictionaries, ProLog for the knowledge base.	**LO2**
9	First-Order Logic Reasoning with Unification: A university wants to check student eligibility rules ("Every student enrolled in AI must take the lab", etc.). Key tasks: • Encode the domain using predicates. • Implement unification and substitution functions. • Demonstrate FOL resolution on small rule sets. Tools: Python, ProLog.	**LO4**
10	Naïve Bayes Text Classifier for Email Spam Filtering: An email client wants an AI filter for spam detection. Key tasks: • Preprocess text (tokenization, stop-words). • Train a Naïve Bayes classifier. • Evaluate accuracy using a testing split. Tools: Python, scikit-learn, NLTK.	**LO4**
11	Bayesian Networks for Fault Diagnosis: A manufacturing plant wants to diagnose machine faults probabilistically. Key tasks: • Define nodes, CPTs, and dependencies. • Perform inference using variable elimination. • Compute posterior probabilities for given observations. Tools: Python, Numpy.	**LO5**
12	Hidden Markov Models for Activity Recognition: A fitness band wants to estimate a user's hidden activity state (walking, running, idle). Key tasks: • Build transition and emission matrices. • Implement Forward and Viterbi algorithms. • Decode most likely hidden state sequence. Tools: Python, scikit-learn.	**LO5**
13	STRIPS-Based Planning for a Delivery Robot: A warehouse robot must plan a sequence of actions to pick and deliver packages. Key tasks: • Define states, actions, preconditions, effects. • Implement STRIPS-style planner (forward search). • Validate plans on sample warehouse layouts. Tools: Python.	**LO5**

14	NLP with Transformers: Text Classification Using a Pre-Trained BERT Model: A social media analytics company wants to classify posts as positive/negative sentiment. Key tasks: • Load pre-trained BERT model for classification. • Fine-tune on a small dataset (or use zero-shot). • Evaluate accuracy and discuss ethics/bias issues. Tools: Python, Hugging Face Transformers, PyTorch/TensorFlow.	**LO6**
15	The Case Study and Implementation of an Expert System: The goal of this experiment is to study the structure and working principles of an Expert System, analyze its components through a real-world case study, and develop a prototype rule-based Expert System using Python. The experiment requires selecting a suitable domain—such as medical diagnosis, academic counselling, technical troubleshooting, or agriculture—extracting expert knowledge, representing it through production rules, and implementing an inference mechanism to generate expert-like decisions.	**LO6**
16	Design and Implementation of Agentic AI Workflows using n8n for Autonomous TaskExecution. To design and implement Agentic AI-based workflows using n8n, an open-source automation tool, by integrating APIs, AI models, and decision-making logic. The system should simulate intelligent agents capable of executing multi-step tasks such as data retrieval, processing, decision-making, and automated response generation.	**LO6**

Textbooks:

1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
2. Elaine Rich & Kevin Knight, Artificial Intelligence, McGraw-Hill.
3. Tom M. Mitchell, Machine Learning, McGraw-Hill.

Reference Books:

1. Nilsson, N.J., 1981. Principles of Artificial Intelligence. Palo Alto: Tioga Publishing.
2. Murphy, K.P., 2021. Machine Learning: A Probabilistic Perspective. 2nd ed. Cambridge, MA: MIT Press.
3. D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd ed. (Draft). Stanford University, 2026.
4. G. F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 6th ed. Boston, MA, USA: Pearson, 2008.
5. R. J. Brachman and H. J. Levesque, Knowledge Representation and Reasoning. San Francisco, CA, USA: Morgan Kaufmann, 2004.

Software Tools:

1. Python 3.10+
2. Jupyter Notebook / Google Colab
3. NumPy, Pandas
4. scikit-learn
5. NLTK / spaCy
6. Matplotlib / Seaborn
7. pgmpy (for Bayesian networks & probabilistic reasoning)
8. hmmlearn (for Hidden Markov Models)
9. NetworkX (for search algorithms, graph problems)
10. TensorFlow / Keras or PyTorch (for ML and deep learning)
11. Hugging Face Transformers (for NLP and LLM experiments)
12. rdflib (for semantic networks & ontology-based KR)
13. Any Python IDE such as VS Code, PyCharm, or Spyder

Term Work:

- Term work should include 12 experiments (Ensure that all LOs are covered).
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).
- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance).

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical & oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL502	Data Engineering Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSL502	Data Engineering Lab	--	--	--	25		25

Pre- requisite:

1. Basic Python programming (pandas, numpy, data handling)
2. Fundamentals of databases (ER modeling, SQL basics)
3. Understanding of data formats (CSV, JSON, structured/semi-structured data)

Program Outcomes Addressed

1. PO1 – Engineering Knowledge
2. PO2 – Problem Analysis
3. PO3 – Design/Development of Solutions
4. PO4 – Conduct Investigations of Complex Problems
5. PO5 – Engineering Tool Usage
6. PO9 – Communication
7. PO10 – Project Management and Finance
8. PO11 – Life-Long Learning

Lab Outcomes:

1. Perform data exploration, cleaning, modeling, and interpretation across structured and semi-structured datasets.
2. Design and implement data models using relational and NoSQL paradigms, applying normalization, schema design, and flexible document modeling.
3. Build ETL and API-driven ingestion pipelines that extract, transform, validate, and load data into analytical systems.
4. Construct OLAP models such as star schemas, cubes, and multidimensional aggregations, and analyze results to generate business insights.
5. Apply modern tools for data warehousing, dashboards, workflow orchestration, and distributed-style processing to automate pipelines and interpret results.
6. Work collaboratively to build and present an end-to-end mini data platform integrating ingestion, storage, validation, transformation, and analytics.

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
A: Foundation — Data Exploration, Cleaning & Modeling		
1	Exploratory Data Analysis & Data Cleaning (Python) • Load CSV dataset • Handle missing values, outliers, and inconsistent types • Generate summary statistics & visualizations Outcome: Understanding data quality and profiling.	**LO1**
2	Relational Schema Design & SQL Normalization • Convert problem statement → ER Diagram • Normalize to 3NF • Implement schema in PostgreSQL • Insertple data, run analytical SQL • Window functions (dense rank, lag/lead) • Query optimization (indexes, partitions, explain plans) • OLAP analytical SQL patterns Outcome: Modeling for structured systems.	**LO2**
3	Mini ETL: CSV → Cleaned Table in PostgreSQL • Extract CSV • Transform (clean, standardize, derive fields) • Load into relational DB • Automate with a simple CLI or script Outcome: First hands-on ETL pipeline.	**LO1**
B: Warehousing, OLAP & Business Insights		

4	To design reusable business metrics using SQL views on top of a warehouse schema and analyze decision-level insights.
	• Define SQL views for:
	  • Daily Revenue
	  • Monthly Sales Growth
	  • Repeat Customer Percentage
	• Query views instead of base tables
	Outcome: Compare readability and reuse vs raw SQL
		LO3

5	Data Cube & Multidimensional Aggregations • Multi-dimension GROUP BY • Cube, rollup, group sets • Compare execution plans Outcome: Practical OLAP operations.	**LO4**
6	Analytics Dashboard for Business KPIs	**LO5**

	Tools: Python/Streamlit Public • Load warehouse tables • Build 3-5 charts answering real questions • Create a mini dashboard Outcome: BI-style storytelling with data.	
C: NoSQL and API-Driven Data Engineering		
7	NoSQL Basics: CRUD & Indexing in MongoDB • Create MongoDB database • Model flexible schemas • Insert documents • Query with filters, projections & indexes Outcome: Understanding document stores.	**LO2**
8	Implementing Delta Lake Features in a Lakehouse Environment • Create a Delta table from a sample dataset. • Insert, update, or delete records to generate multiple table versions. • Demonstrate time travel by querying older versions of the Delta table. • Test schema enforcement by attempting to write with incompatible schema. • Record and interpret system responses. Tools: Option A – Local / Python Environment: • Python • PyDeltaLake library • Pandas • Local filesystem for storing Delta tables Option B – Cloud / Databricks Community Edition: • Databricks CE notebook environment • Delta Lake runtime • DBFS (Databricks File System) • Spark SQL / PySpark	**LO5**
9	Building REST APIs for Data Capture • Using Flask or FastAPI: • Create POST/GET endpoints • Validate JSON payloads • Store into MongoDB/PostgreSQL Outcome: How Data Enters Real DE Pipelines.	**LO3**
D: Distributed Data & Pipelines		
10	Event-Driven Pipelines using RabbitMQ/Redpanda (Kafka-lite) • Create producer • Create consumer	**LO5**

	• Consume streaming data • Store into the staging table Outcome: Foundation of streaming ingestion.	
11	Distributed Data Processing Without Spark/Hadoop Use DuckDB or Polars for large-scale local batch processing: • Load large CSV • Run parallel queries • Optimize using columnar formats Outcome: Teaches "distributed-style" parallel processing without requiring Spark.	**LO4**
E: Workflow, Automation & Final Pipelines		
12	Workflow Orchestration using Airflow or Prefect • Create simple DAG • Extract → Transform → Load • Add scheduling and logging Outcome: Automation & orchestration skills.	**LO3**
13	Data Observability and Quality Validation using Expectation Suites • Create an expectation suite for specifying null checks, value ranges, column constraints, and datatype rules. • Validate a sample dataset using the expectation suite. • Generate an HTML validation report summarizing passed and failed expectations. • Interpret the report and document insights. Outcome: Trust and governance in data.	**LO5**
14	End-to-End Batch Pipeline Project • Ingest → clean → validate → transform → load • Store final data in Parquet or PostgreSQL • Maintain logs and error files Outcome: Realistic DE batch workflow.	**LO6**
15	Application Built up: Build a Mini Data Platform Groups build a pipeline combining: • API ingestion • NoSQL/SQL modeling • Validation • ETL • Analytics dashboard Outcome: Full data engineering life cycle.	**LO6**

Textbooks:

1. Kleppmann, M. (2017) Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media.
2. Meyer, A. (2018) The Data Engineering Cookbook.
3. Bhagwan, R., et al. (2021) Data Storage Survey: Modern Data Formats, Partitioning Strategies, and Warehouse–Lake Architectures.

Reference books:

1. McKinney, W. (2022) Python for Data Analysis. 3rd edn. Sebastopol: O'Reilly Media.
2. Densmore, J. (2021) Data Pipelines Pocket Reference: Moving and Processing Data for Analytics. Sebastopol: O'Reilly Media.
3. J. Cito, P. Leitner, and H. C. Gall(2019), Engineering of Big Data Processing Systems. Cham, Switzerland: Springer.

Online Resources:

1. AWS Data Engineering Learner's Path (Free Tier Friendly)
   https://aws.amazon.com/learning-paths/data-engineer/
2. Google Cloud Big Data & ML Fundamentals (Coursera Audit Free)
3. MongoDB University Courses
   https://learn.mongodb.com
   (Excellent for schema design, aggregation pipeline)
4. Apache Airflow Documentation https://airflow.apache.org/docs/
   (Needed for workflow orchestration.)
5. Kafka (or Redpanda) Documentation
   https://redpanda.com/documentation/
   (Redpanda is easier to run on Windows)
6. FastAPI Documentation
   https://fastapi.tiangolo.com (REST API creation.)
7. DuckDB Documentation (If using local OLAP)
   https://duckdb.org/docs/ (Lightweight, easy for partitioning, Parquet

Software Tools

1. Python 3.x + VS Code
   For EDA, ETL, data cleaning, API development, feature engineering.
2. PostgreSQL (or MySQL)
   For relational schema design, fact/dimension tables, OLTP/OLAP queries.
3. MongoDB Community Server
   For NoSQL modeling, API storage.
4. Redpanda (Kafka-compatible lightweight broker)
   Windows-friendly event ingestion for.
5. Apache Airflow (via Docker or WSL)
   Needed only for 1 lab.
6. DuckDB (Standalone EXE)
   Zero-installation, ideal for Parquet, partitioning, OLAP-style queries.

Term Work:

- Term work should consist of at least 12 experiments.
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
- 25 Marks (Total Marks) = 20 Marks (Experiment) + 05 Marks (Attendance)

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL503	Data Warehousing and Mining Lab	--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSL503	Data Warehousing and Mining Lab	--	--	--	25	25	50

Pre- requisite:

1. FEC104: C-Programming
2. FEL205: Object Oriented Programming Methodology Lab
3. AIDSL304: Skill Lab (Python Programming)

Program Outcomes Addressed

1.PO1: Engineering Knowledge
2.PO2: Problem Analysis
3.PO3: Design/Development of solutions
4.PO4: Conduct investigations of complex problems
5.PO5: Modern Tool Usage

Lab Objectives:

1. Apply data warehouse design principles to construct basic warehouse schemas.
2. Implement data preprocessing techniques to clean and prepare real-time data.
3. Apply suitable classification algorithms using real-time datasets.
4. Apply clustering algorithms to analyze patterns within datasets.
5. Examine frequent pattern mining results to identify significant associations
6. Use WEKA software to perform data mining tasks including classification, clustering, and pattern mining.

Lab Outcomes:

1. Design data warehouses and perform various OLAP operations.
2. Analyze the data and apply various preprocessing techniques to clean the data.
3. Apply appropriate classification algorithms to any real-time data.
4. Apply appropriate clustering algorithms to any real-time data.
5. Apply frequent pattern matching algorithms to find useful patterns.
6. Apply web structure mining algorithm and use of WEKA tool.

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
1	A retail company stores sales data in multiple files, making analysis difficult. Management needs a centralized system for better decision-making. Design a data warehouse using star and snowflake schemas. Identify fact and dimension tables. Perform ETL using Power Query in Excel.	**LO1**
2	Create reports using Excel Power Pivot. Perform Slice, Dice, Roll-up, and Drill-down operations on retail company data warehouse (refer experiment no. 1). Use Pivot tables for summarization. Help management analyze trends efficiently.	**LO1**
3	A healthcare analytics company analyzes patient records to identify diabetes risk. The dataset includes features like glucose, BMI, age, insulin, and blood pressure, with possible missing values and outliers. Students must perform data preprocessing and EDA by handling null values, applying normalization and transformation, conducting descriptive analysis, and detecting/treating outliers using a Python notebook or visualization tool.	**LO2**
4	A retail company wants to better understand its customers for targeted marketing. The dataset contains attributes such as age, income, spending score, and purchase frequency. Apply discretization techniques like binning. Create visualizations such as histograms or charts. Extract useful insights.	**LO2**
5	HR Analytics helps us with interpreting organizational data. It finds out the people-related trends in the data and helps the HR Department take the appropriate steps to keep the organization running smoothly and profitably. Explore and analyze company's employees dataset to predict potential attrition cases, thereby helping the appropriate HR Personnel take the necessary steps to retain the employee.	**LO2**
6	A hospital wants to predict diseases using patient data. Historical data is available for training. Patient attributes used for prediction, such as: Age, Blood Pressure, Glucose Level, BMI, Insulin, Cholesterol (optional). Implement the KNN algorithm. Classify new patient records based on neighbors. Evaluate model accuracy.	**LO3**
7	An e-commerce platform wants to predict customer purchases. User behavior data is available.User profile data(User ID,Age,Gender Location) etc. Browsing Behavior (Number of pages visited,Time spent on website/app,Number of product views,Categories viewed,Search queries count) Implement any classification algorithm using Java or Python. Train and test the model. Measure performance.	**LO3**
8	An email system wants to detect spam messages. Use labeled data for training. Implement Naïve Bayes or Decision Tree algorithm. Classify emails as spam or not spam. Improve accuracy.	**LO3**
9	Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool (WEKA/R tool)	**LO3**
10	A telecom company wants to segment its customers to design targeted marketing strategies. The company collects customer usage data such as monthly bill amount, data usage, call duration, and number of recharge plans. Students are required to apply the K-Means or K-Medoid clustering algorithm to group customers into similar segments and analyze their behavior patterns.	**LO4**

11	A research lab is analyzing biological samples to identify groups with similar characteristics. Each sample is described by features such as protein expression levels, gene activity, or chemical composition. The relationships between samples are important, so the lab uses hierarchical clustering to understand how samples group together. Students are required to apply hierarchical clustering (agglomerative method), generate a dendrogram, and identify meaningful clusters.	**LO4**
12	A research team is analyzing spatial data where clusters are irregular in shape and contain noise. Traditional clustering methods like K-Means fail to detect these patterns effectively. Students must apply DBSCAN to identify dense regions and noise points	**LO4**
13	A supermarket wants to analyze purchase patterns. Identify frequently bought item sets. Apply Apriori algorithm. Generate association rules. Improve sales strategies.	**LO5**
14	A search engine needs to rank web pages. Ranking depends on links between pages. Implement PageRank algorithm. Calculate page importance scores. Display relevant pages first.	**LO6**
15	A company collects customer feedback from reviews and social media. The data is unstructured and difficult to analyze manually. Apply text preprocessing techniques like tokenization and stopword removal. Implement sentiment analysis or text classification. Extract insights for business decisions.	**LO6**

Textbooks:

1. Jiawei Han, Micheline Kamber, “Data mining: concepts and techniques", Morgan Kaufmann Publisher 2012, third edition, ISBN 978-0-12-381479-1
2. G. K. Gupta, “Introduction to Data mining with Case Studies", PHI Learning Private Limited, Delhi 2014, third edition, ISBN-978-81-203-5002-1.
3. William H Inmon, “Building the data Warehouse”, Wiley Publication 2005, fourth edition, ISBN: 978-0-764- 59944-6.

Reference Books

1. Dunham, M. H., “Data mining: Introductory and advanced topics”, Upper Saddle River, N.J: Pearson education/Prentice Hall 2003
2. Ralph Kimball, Margy Ross, “The Data Warehouse Toolkit”, 3rd Edition, Wiley 2013, ISBN-13: 978-11185308 01.
3. Giuseppe Bonaccorso, “Machine Learning Algorithms”, Packt Publishing Ltd. 2017, ISBN 978-178588-962- 2.

Online Resources:

1. Data Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining. Concepts-and Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series
2. Data Mining Weka_2nd_Ed_2005.pdf http://www.academia.dk/BiologiskAntropologi/Epidemiologi/DataMining/Witten_and_Frank
3. GitHub Pages documentation http://scikit-learn.org/stable/datasets/
4. Metrics and scoring: quantifying the quality of predictions https://scikitlearn.org/stable/modules/model_evaluation.html
5. Datasets https://www.kaggle.com/datasets

Software Tools

1. Google Colaboratory (Colab) — https://developers.google.com/colab
2. WEKA (machine-learning/data-mining tool) — https://ml.cms.waikato.ac.nz/weka/
3. Microsoft Power BI — https://www.microsoft.com/en-in/power-platform/products/power-bi/

Term Work:

- Term work shall consist of at least 12 experiments (Ensure that all the LOs are covered).
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical & oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theor	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL504	Interpersonal & Career Skills		2*+2			02		02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$^{\$}$			
		ISE	MSE				
AIDSL504	Interpersonal & Career	-	-	-	50		50

Pre-requisite:
Professional Communication Techniques Course

Program Outcome mapped:

1. PO6: The Engineer and the World
2. PO7: Ethics
3. PO8: Individual and Collaborative Teamwork
4. PO9: Communication
5. PO10: Project Management & Finance
6. PO11: Lifelong Learning

Course Objectives

1. To equip students with career development skills to align with the corporate and global requirements.
2. To develop students' ability to conduct and participate in professional business meetings.
3. To apply the knowledge of effective writing skills to produce technical/business documents.
4. To train students in creating effective, dynamic technical/business presentations.
5. To build strong interpersonal skills required for effective workplace communication.
6. To familiarize students with corporate ethics and Intellectual Property Rights.

Course Outcomes

1. Apply essential skills while appearing for recruitment processes or applying for higher education for a successful career.
2. Demonstrate professional behaviour; to plan, participate in, and document business meetings effectively.
3. Compose well-structured technical/business documents.
4. Design and deliver effective technical/business presentations using suitable strategies.
5. Employ strong interpersonal skills to overcome workplace challenges by understanding professional communication.
6. Practice principles of corporate ethics and understand Intellectual Property Rights.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.		**Career Skills**	**08**	
	1.1	SWOT Analysis		**CO1**
	1.2	Cover letter, Resume/CV:Resume Building, Types of Resumes (Chronological, Functional & Combination),Curriculum Vitae, Difference between Resume and CV		
	1.3	Group Discussion: Purpose, Types of GDs (Normal, Case- based & Role Plays)		
	1.4	GD Etiquettes, Parameters of Evaluating a GD		
	1.5	Interviews (Face to face and Virtual): Planning and Preparation, Types of Questions, Types of Interviews (Structured, Stress, Behavioural, Problem Solving		
	1.6	Modes of Interviews: Face-to-face, Telephonic		
	1.7	Virtual Interview planning and preparation, Online interview etiquettes		
	1.8	Statement of Purpose		
2.		**Business Meetings**	**03**	
	2.1	Documentation:Notice, Agenda, Minutes of the Meeting		**CO2**
	2.2	Conducting Business Meetings: Types of Meetings, Roles and Responsibilities of Chairperson, Secretary and Members, Meeting Etiquettes		
	2.3	Writing Minutes of the Meeting		
3.		**Advanced Technical Writing**	**06**	
	3.1	Purpose and Classification of Reports: Classification based on Subject Matter (Technology, Accounting, Finance, Marketing, etc.), Time Interval (Periodic, One-time, Special)		**CO3**
	3.2	Function (Informational, Analytical, etc.), Physical Factors (Memorandum, Letter, Short & Long)		
	3.3	Language and Style: Tense, Person & Voice of Reports, Numbering Style of Chapters, Sections and Figures/Tables, Proof-reading through Plagiarism Checkers		
	3.4	Project Report: Prefatory Parts (Front Matter), Report Proper (Main Body), Appended Parts (Back Matter)		
	3.5	Proposals: Format and Style of Technical Proposals, Parts of a Proposal		

	3.6	Technical Paper Writing: Parts of a Technical Paper, Language, Style and Formatting, Referencing in IEEE Format		
4.		**Effective Presentation**	**02**	
	4.1	Effective Presentation Strategies: Defining purpose, analyzing audience & location, Structuring a presentation, Types of presentation aids, Using body language and Voice		**CO4**
	4.2	Group Presentation: Sharing Responsibility in a Team, Building the contents and visuals together, Transition		
5.		**Interpersonal Skills**	**05**	
	5.1	Emotional Intelligence		**CO5**
	5.2	Leadership		
	5.3	Time Management		
	5.4	Decision Making		
	5.5	Responsible Use of social media		
6.		**Corporate Ethics and Intellectual Property Rights**	**02**	
	6.1	Introduction to Corporate Ethics		**CO6**
	6.2	Intellectual Property Rights: Copyrights, Trademarks, Patents, Industrial Designs, Geographical Indications, Integrated Circuits, Trade Secrets (Undisclosed Information)		
		Total	**26**	

Reference books:

1. J. Butterfield, Professionalism: Soft Skills for a Digital Workplace. Boston, Massachusettes: Cengage Learning, [] © Course Technology, Cengage Learning, 2011.
2. R. V. Lesikar and J. D. Petit, Report Writing for Business, 10th ed. McGraw-Hill/Irwin, 1995.
3. L. A. Olsen and T. N. Huckin, Technical Writing and Professional Communication, 2nd ed. McGraw-Hill Humanities, Social Sciences & World Languages, 1991.
4. L Ann Masters and H. Wallace, Personal Development for Life and Work., 12th ed. Mason, Oh: South-Western Cengage Learning, 2011.
5. H. A. Murphy and Herbert William Hildebrandt, Effective Business Communications, 7th ed. McGraw-Hill Companies, 1988.
6. F. Luthans, Organization Behaviour, 12th ed. New York: Mcgraw Hill, 2011.
7. R. C. Sharma and K. Mohan, Business Correspondence and Report Writing : A Practical Approach to Business & Technical Communication, 6th ed. New Delhi: Tata Mcgraw-

Hill, 2017.
8. K. Alex, Soft skills : Know yourself & know the world, 3rd ed. New Delhi: S. Chand & Company Ltd, 2012.
9. R Subramanian, Professional ethics, 2nd ed. New Delhi: Oxford University Press, 2013.
10. S. Robbins and T. A. Judge, Organisational Behaviour, 18th ed. Pearson Higher Education AU, 2024.

Online References:

1. Directorate General of Training (DGT).
Employability Skills Curriculum. Ministry of Skill Development & Entrepreneurship, Government of India, https://www.dgt.gov.in/sites/default/files/Employability_Skills_Curriculum.pdf.

2. Punjab State Open University (PSOU).
Technical Writing and IPR Syllabus. PSOU, https://psou.ac.in/asset/docs/course_syllabus/20221125183033165f0acdb4.pdf.

3. National Institute of Technology Kurukshetra (NITKKR).
Professional Ethics and Intellectual Property Rights (HSIR-13). NITKKR, https://nitkkr.ac.in/wp-content/uploads/2024/07/Professional-Ethics-and-IPR-HSIR-13.pdf.

4. Commonwealth Corporation.
Career Readiness Curriculum Guide. CommCorp, https://commcorp.org/wp-content/uploads/2016/07/resources_eyf-career-readiness-curriculum-guide.pdf.

Course Assessment:
Term Work (CIAP): **50 Marks**

Term Work (CIAP) shall consist of Oral & Written assessment for 25 marks each.

Oral evaluation to be conducted for **25 Marks** through:

a.	Report Presentation	10 marks
b.	Group Discussion	10 marks
c.	Mock Interviews	05 marks

Written evaluation to be conducted for **25 Marks** through:

a.	Project Report (25 to 30 pages)	10 marks
b.	Assignments based on all modules	10 marks
c.	Attendance	05 marks

CIAP will be conducted based on the above scheme for **50 Marks.**

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML5031	Embedded Systems and RTOS Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML5031	Embedded Systems and RTOS Lab	--	--	--	25	--	25

Pre- requisite:

1. FEC204: Digital System Design
2. FEL103: C Programming Lab
3. MDMC4031: Microprocessors & Microcontrollers

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering Tool Usage
5. PO6: The Engineer and The World
6. PO7: Ethics

Lab Objectives:

1. Develop hands-on skills in configuring microcontrollers (8051/ARM/STM32/Arduino) using digital I/O, timers, ADC, and interrupts.
2. Interface sensors, actuators, and communication peripherals such as push buttons, LDRs, buzzers, motors, and I2C-based devices.
3. Implement low-power design techniques and evaluate power efficiency in embedded applications.
4. Apply embedded communication standards and protocols including UART, I2C, and interrupt-driven interfaces.
5. Apply functional safety and cybersecurity standards (IEC 61508, IEC 60601, ISO 26262, ISO 21434) through case study analysis.
6. Develop Real-Time Operating System (RTOS) based embedded applications, demonstrating multitasking, scheduling, and inter-process synchronization.

Lab Outcomes:

1. Interface and program basic embedded peripherals and sensors using GPIO, ADC, PWM, and interrupt logic.
2. Configure communication interfaces such as UART and I2C to connect external modules like DS1307 RTC.
3. Design and implement low-power embedded applications and analyze energy behavior.
4. Interpret and apply embedded safety and security requirements for compliance-driven design.
5. Develop RTOS-based applications using tasks, queues, events, and semaphores.
6. Implement and evaluate scheduling algorithms (RMS/EDF) to ensure deterministic real-time behavior.

Suggested List of Experiments: Students are required to complete at least 12 experiments.

Sr. No.	Title of Experiments	LO
1	Interfacing a Push Button with Interrupt Handling on an Embedded Core (8051/ ARM/ STM32) **Task:** Interface a push button to a GPIO pin on an embedded core (8051, ARM, or STM32), configure the pin for external interrupt (rising/falling edge), and handle the interrupt in an ISR to perform actions like toggling an LED or incrementing a counter.	**LO1**
2	Interfacing a Buzzer / Audio Alert Device with an Embedded Core (8051/ ARM/ STM32) **Task:** Interface a buzzer or audio alert device with an embedded core (8051, ARM, or STM32) to generate simple tones or alerts via GPIO toggling or PWM signals in a controlled experiment.	**LO1**
3	Interfacing an Analog Light Sensor (LDR) and Displaying Values on Serial Monitor **Task:** Interface an analog Light Dependent Resistor (LDR) to an ESP32 ADC pin, read real-time light intensity via voltage divider, convert raw ADC values to approximate lux or percentage, and display readings on the serial monitor using a FreeRTOS task.	**LO2**
4	Interfacing a simple Temperature Sensor and Displaying Real-Time ADC Values on Serial Monitor **Task:** Interface a simple analog temperature sensor (LM35) to an ESP32 GPIO pin, read real-time ADC values, convert to voltage and temperature, and display raw ADC, voltage, and temperature readings on the serial monitor using a FreeRTOS task.	**LO2**
5	Implement the I2C communication to connect to DS1307 RTC **Task:** Implement I2C communication with the DS1307 Real Time Clock (RTC) module to initialize the device, set the current date/time, and periodically read/display the time in a FreeRTOS-based embedded application.	**LO2**
6	Implementing a Simple Temperature-Controlled Fan System (Sensor + DC Motor) **Task:** Implement a temperature-controlled fan system using a DHT11 sensor to read ambient temperature and adjust DC motor (fan) speed via PWM based on thresholds (e.g., off below 25°C, variable speed above) in a FreeRTOS application on ESP32.	**LO3**
7	Implement power management in any embedded core of your choice **Task:** Implement power management on an ESP32 embedded core by configuring low-power modes (light sleep and deep sleep), using wake-up sources (timer, GPIO touch), measuring current reduction, and integrating with FreeRTOS tasks for periodic operation.	**LO3**

8	Interfacing a DC Motor (Speed and Direction Control) with Embedded Core **Task:** Interface a DC motor with an embedded microcontroller to control both speed (using PWM) and direction (using H-bridge logic) in a FreeRTOS-based application, demonstrating forward/reverse rotation and variable speeds.	**LO3**
9	Case Study Analysis and Compliance Mapping for IEC 61508, IEC 60601, and ISO 26262 **Task:** Analyze a hypothetical case study (e.g., development of an automotive-integrated medical monitoring device) for functional safety compliance, mapping requirements across IEC 61508 (general), IEC 60601 (medical devices), and ISO 26262 (automotive) to identify overlaps, differences, and verification strategies.	**LO4**
10	Developing a Safety Lifecycle Flow and Verification Checklist Based on IEC 61508 or ISO 26262 **Task:** Develop a safety lifecycle flow diagram outlining key phases (concept, product development at system/hardware/software levels, production, operation, and decommissioning) based on IEC 61508 or ISO 26262 and create a verification checklist for compliance in each phase.	**LO4**
11	Porting of FreeRTOS to Arduino/STM32 **Task:** Port FreeRTOS to Arduino (AVR-based Uno) and STM32 platforms by the appropriate FreeRTOS port, configuring the kernel, creating basic tasks, and verifying multitasking behavior through LED blinking or serial output.	**LO5**
12	Write a Program to illustrate the Queue Management Features of FreeRTOS **Task:** Develop a C program in FreeRTOS to demonstrate queue management by creating a producer task that sends structured data items to a queue and multiple consumer tasks that receive and process items from the queue, illustrating blocking, non-blocking, and timeout behaviors.	**LO5**
13	Write a Program to illustrate the Event Management Features of FreeRTOS **Task:** Develop a C program in FreeRTOS to demonstrate event group features by creating multiple tasks that set specific event bits upon certain conditions and a monitoring task that waits for combinations of events (using logical AND/OR) to synchronize actions.	**LO5**
14	Write a Program to illustrate the use of Binary and Counting Semaphore for Task Synchronisation using FreeRTOS **Task:** Develop a C program in FreeRTOS demonstrating task synchronization with a binary semaphore (for one-to-one signaling between producer and consumer tasks) and a counting semaphore (for managing multiple resource instances or events in a producer- consumer scenario).	**LO5**
15	Write a Program to Create Multiple Tasks and understand the Multitasking capabilities of RTOS(FreeRTOS).	**LO6**
	Task: Create multiple independent tasks in FreeRTOS with different priorities and delays to demonstrate preemptive multitasking, context switching, and concurrent execution on a single-core microcontroller.	

Textbooks:

1. K. C. S. Murti, Design Principles for Embedded Systems. Boca Raton, FL, USA: CRC Press, 2017.
2. C. Hobbs, Embedded Software Development for Safety-Critical Systems, 1st ed. Oxford, U.K.: Newnes (Elsevier), 2015.
3. B. Amos, Hands-On RTOS with Microcontrollers, 1st ed. Birmingham, U.K.: Packt

Publishing, 2020.

Reference books:

1. M. J. Pont, The Engineering of Reliable Embedded Systems, 2nd ed. London, U.K.: Routledge, 2017.
2. P. Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems and IoT, 4th ed. Cham, Switzerland: Springer, 2021.

Online Resources:

1. ControllersTech Tutorials – https://controllerstech.com
2. FreeRTOS Official Documentation & Tutorials – https://www.freertos.org
3. Learn Embedded Systems GitHub Repository – https://github.com/iam-sandipmaity/learn-embedded-systems

Software Tools:

1. STM32CubeIDE / Keil / GCC Toolchain
2. FreeRTOS Kernel
3. MCU Debuggers / On-chip Debug Tools
4. Logic Analyzer (Hardware/Software)
5. Serial Terminal / Data Visualization Tools (PuTTY / TeraTerm / Arduino Serial Monitor)
6. Simulation Software (Proteus)

Term Work:

- Term work should consist of at least 12 experiments (Ensure that all the LOs are covered)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
- 25 Marks (Total Marks) = 20 Marks (Experiment) + 05 Marks (Attendance).

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDML 5041	Cryptography and Network Security Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 5041	Cryptography and Network Security Lab	--	--	--	25	-	25

Pre-requisite:
CEL502 Computer Network Lab

Program Outcomes Addressed:
1. PO2: Problem Analysis
2. PO3: Design/Development of Solutions
3. PO4: Conduct investigations of complex problems
4. PO5: Engineering tool usage
5. PO7: Ethics
6. PO11: Lifelong learning

Lab Objectives:
The course aims to enable students:
1. To implement basic symmetric and asymmetric cryptographic algorithms.
2. To understand key management, distribution, and authentication mechanisms.
3. To use network reconnaissance tools for information gathering.
4. To analyze network traffic using sniffers and packet-capture tools.
5. To detect vulnerabilities and simulate network attacks using open-source tools.
6. To configure and evaluate system and network security mechanisms.

Lab Outcomes:
1. Illustrate symmetric cryptography by implementing classical ciphers.
2. Demonstrate Key management, distribution and user authentication.
3. Examine the different network reconnaissance tools to gather information about networks.
4. Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.
5. Use open source tools to scan the network for vulnerabilities and simulate attacks.
6. Demonstrate the system and network security mechanism using open source tools.

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
1	Product cipher using Substitution and Transposition ciphers: Design and implement a secure text encryption system by combining substitution (e.g., Caesar/Vigenère cipher) and transposition techniques. Demonstrate encryption and decryption of sample messages and evaluate the strength of the combined (product) cipher against frequency analysis. Tools: Python / C++ / Java, Jupyter Notebook / VS Code	**LO1**
2	Case study: A secure text-hiding system using image-based steganography techniques (e.g., Least Significant Bit insertion). Demonstrate the process of embedding secret messages into digital images and retrieving them accurately. Evaluate the imperceptibility of the hidden data against visual inspection and statistical steganalysis, and compare the robustness of the system under different image formats and compression level	**LO1**
3	RSA Cryptosystem & Digital Signature (RSA/ElGamal): Implement the RSA algorithm for secure message encryption and decryption. Extend the implementation to generate and verify digital signatures using RSA or ElGamal. Analyze key generation, encryption time, and signature verification process. Tools: Python / C++ / Java, Jupyter Notebook / VS Code	**LO2**
4	Diffie Hellman Key exchange algorithm : Develop a secure communication module that implements the Diffie-Hellman key exchange algorithm to establish a shared secret between two parties. Demonstrate how the shared key is used for secure message transmission. Tools: Python , Jupyter Notebook / VS Code	**LO2**
5	Hashing Algorithms (MD5, SHA-2, SHA-3) – Integrity & Performance Analysis: Develop an application to compute hash values of messages with varying sizes using MD5, SHA-2, and SHA-3. Measure execution time, analyze collision resistance, and compare their security against known attacks. Tools: Python , Jupyter Notebook / Google Colab	**LO2**
6	AES/DES Encryption with Modes of Operation : Implement encryption and decryption of large text files using AES or DES in different modes of operation (ECB, CBC, CFB, OFB). Analyze the effect of each mode on security and performance. Tools: Python , Jupyter Notebook / Google Colab	**LO2**
7	Network Reconnaissance Tools Perform network reconnaissance on a given domain to gather information such as IP address, domain registrar, DNS records, and routing path. Analyze how each tool contributes to footprinting. Tools: whois, dig, nslookup, traceroute Platform: Linux (Kali Linux / Ubuntu Terminal)	**LO3**

8	Network Scanning using Nmap Install and use Nmap to scan a target system and identify open ports, services, and operating system details. Perform multiple scans including ping scan, TCP scan, UDP scan, and Xmas scan, and compare the outputs. Tools: nmap Platform: Kali Linux / Windows (with Nmap installed)	**LO4**
9	ARP Spoofing Detection and Analysis Simulate ARP spoofing in a controlled environment and detect it using monitoring tools. Generate gratuitous ARP packets and analyze traffic patterns to identify spoofing attacks. Tools: arping, Wireshark, arpwatch, nmap Platform: Kali Linux / Virtual Lab setup (VMware/VirtualBox)	**LO3 , LO4**
10	Simulation of DoS Attack Simulate a Denial-of-Service (DoS) attack in a controlled lab environment and analyze its impact on network performance and system availability. Evaluate packet flow and server response under attack conditions. Tools: hping, hping3 Optional : Wireshark (for traffic analysis) Platform: Kali Linux (Virtual Lab only)	**LO5**
11	Malware Analysis and Vulnerability Assessment a) Perform vulnerability scanning on a target system and identify security weaknesses. b) Simulate a keylogging attack in a controlled environment and analyze captured keystrokes using secure tools or frameworks. Tools: Nessus / Kali Linux tools (e.g., OpenVAS) Metasploit Framework (Meterpreter keylogging module) Platform: Kali Linux / Virtual Lab setup	**LO5**
12	Network Security Configuration (IPSec & IDS) a) Configure IPSec on a Linux system to establish secure communication between two nodes. b) Install and configure Snort IDS, capture network traffic, and analyze intrusion detection logs. Tools: IPSec (strongSwan), Snort Platform: Linux (Ubuntu/Kali)	**LO6**
13	Firewall Configuration using iptables Design and implement a host-based firewall using iptables. Create rules to allow/deny traffic based on IP address, port number, and protocol, and test their effectiveness. Tools: iptables Platform: Linux (Ubuntu/Kali)	**LO6**
14	Email Security using GPG Implement secure email communication using GPG by generating key pairs, encrypting, decrypting, and digitally signing messages. Demonstrate	**LO6**

	confidentiality and authentication. Tools: GPG (GNU Privacy Guard) Platform: Linux Terminal	
15	Role-Based Access Control (RBAC) Implementation Design and implement Role-Based Access Control (RBAC) by creating multiple users, assigning roles, and defining permissions. Demonstrate how access to system resources is restricted based on roles. Tools: Linux user management commands (useradd, groupadd, chmod, chown) Optional: Python (for simulation) Platform: Linux (Ubuntu/Kali)	**LO6**

Textbooks:

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Pearson, 2024.
2. B. A. Forouzan, Cryptography and Network Security, Revised ed. New Delhi, India: Tata McGraw-Hill, 2011.
3. B. L. Menezes and R. Kumar, Cryptography, Network Security, and Cyber Laws, 1st ed. New Delhi, India: Cengage Learning, 2018.
4. E. Cole, Network Security Bible, 2nd ed. Hoboken, NJ, USA: Wiley, Mar. 2011.
5. W. Stallings and L. Brown, Computer Security: Principles and Practice, 4th ed. Boston, MA, USA: Pearson, 2018.

Reference books:

1. M. Gregg, Build Your Own Security Lab: A Field Guide for Network Testing, 1st ed. Indianapolis, IN, USA: Wiley, 2008.
2. T. Boyles, CCNA Security Study Guide: Exam 640-553, 1st ed. Hoboken, NJ, USA: Wiley/Sybex, 2010.
3. D. Stuttard and M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.

Online Resources:

1. https://www.wireshark.org/download.html
2. https://nmap.org/
3. https://www.tenable.com/products/nessus
4. https://cse29-iiith.vlabs.ac.in/
5. https://www.cryptool.org/en/cto/
6. https://www.dcode.fr/en

Term Work:

- The term work shall include a total of 12 experiments.
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Mark (Attendance).

Course Code	Course Name	Credits Assigned			
		Theory	Practical	Tutorial	Total
AIDSM501	Mini Project 2	--	1	--	1

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSM501	Mini Project 2	--	--	--	25	25	50

Program Outcomes:

1. PO1: Engineering knowledge.
2. PO2: Problem Analysis.
3. PO3: Design/Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems.
5. PO5: Engineering Tool Usage.
6. PO6: The Engineer and The World.
7. PO7: Ethics.
8. PO8: Individual and Collaborative Team work.
9. PO9: Communication.
10. PO10: Project Management and Finance.
11. PO11: Life-Long Learning.

Objectives:

1. To identify user or societal needs and formulate clear problem statements.
2. To work collaboratively for effective problem-solving.
3. To apply engineering fundamentals to develop feasible solutions.
4. To promote self-learning and research-oriented thinking.

Outcomes:

1. Identify and define real-world or research problems through requirement gathering and stakeholder interaction.
2. Develop project objectives, scope, and feasibility using software engineering and SDLC principles.
3. Analyze requirements and develop system designs using standard modelling techniques.
4. Implement solutions using appropriate technologies, engineering practices, and version control.
5. Test and validate the system through structured test cases, datasets, and simulations.
6. Compose technical documentation, present project outcomes, and demonstrate teamwork, ethics, and self-learning.

Guidelines for Mini Project

1.1 General Guidelines

1. Students shall form groups of 3 to 4 members; groups with fewer than three or more than four students are not permitted.
2. Students should conduct surveys to identify needs and develop problem statements in consultation with faculty.
3. Each group shall submit an **implementation plan** using a Gantt Chart, PERT, or CPM chart, covering weekly activities and milestones.
4. Every group must maintain a **logbook** to record weekly progress, tasks completed, challenges faced, and supervisor's comments and signatures.
5. Faculty supervisors may provide necessary guidance; however, emphasis shall be on **self-learning, independent thinking, and initiative**.
6. Students should understand the problem thoroughly, **propose multiple solution approaches**, and select the most appropriate solution in consultation with the supervisor.
7. Students shall develop the selected solution into a working model using domain-specific tools, technologies, or components and demonstrate the same.
8. The solution must be validated using proper justification through test cases, benchmark data, experimentation, or simulations, and the final report must be compiled as per the guidelines for the project report.
9. To promote innovation, societal relevance, self-learning, and entrepreneurial mindset, it is recommended that a single project be carried out across two consecutive semesters (Mini Project II in Sem V & VI).
10. If a group completes the project with required quality in the odd semester, they may, based on supervisor recommendation, either extend the project with improvements or begin a new project in the even semester; this may be permitted on a case-by-case basis.
11. Students must follow **ethical practices**, maintain academic integrity, avoid plagiarism, and appropriately acknowledge all references and sources used.
12. Students are encouraged to use **standard engineering tools** such as version control systems (e.g., Git), project management tools, and proper documentation tools during project execution.
13. Evaluation will consider problem identification, planning, implementation quality, testing and validation, documentation, presentation, teamwork, and overall contribution.
14. Students should be encouraged to explore opportunities for **innovation, entrepreneurship**, and potential **intellectual property** creation such as patent, copyright, or research publication.
15. Students must follow **lab safety norms** and responsibly use equipment, hardware, and computational resources during the execution of the project.

1.2 Guidelines for the Project Report:

1. Report should not exceed 30 pages. Simply staple it to discourage use of plastic.
2. Report must contain Gantt chart, architectural diagram, Data Flow Diagrams, applicable UML diagrams, screenshot of outputs (Include only required information pages).
3. The recommended report writing format is in LaTeX.

2 Guidelines for Assessment of Mini Project:

2.1 Evaluation of Term Work(CIAP):

- A Review/Progress Monitoring Committee shall be constituted by the Head of Department. Mini Project progress shall be evaluated continuously, with a minimum of two reviews per semester.
- Continuous assessment shall also emphasize individual performance, including each student's contribution, understanding of the project, and responses during review interactions.
- The Review/Progress Monitoring Committee may consider the following points:
  a) students' group shall complete project in all aspects including,
     i. Identification of need/problem
     ii. Proposed final solution
     iii. Procurement of components/systems
     iv. Building prototype and testing
  b) Two reviews will be conducted for continuous assessment,
     i. **First Review:** Finalization of problem and proposed solution
     ii. **Second Review:** Implementation and testing of the solution
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).
  1. **First Review:** Readiness for building the working prototype
  2. **Second Review:** Poster presentation and demonstration of the working model in the last month of the semester.

2.2 Distribution of Marks

The distribution of Term Work marks shall be as follows:

a) Marks by Guide/Supervisor based on Logbook: 10
b) Marks by Review Committee: 10
c) Quality of Project Report: 05

2.3 Evaluation Rubrics CIAP (Term work)

Mini Project shall be assessed based on the following criteria;

a) Quality of Survey and Need Identification
b) Clarity and Completeness of Problem Definition
c) Innovativeness and Creativity in Proposed Solutions
d) Feasibility of Solution and Justification of Selected Approach
e) Cost Effectiveness and Societal/Environmental Impact
f) Quality of System Design and Use of Engineering Norms
g) Implementation Quality and Functioning of Prototype as per Requirements

h) Effective Use of Technical Skill Sets and Tools
i) Teamwork, Leadership, and Individual Contribution
j) Clarity in Documentation, Presentation, and Communication

All criteria in generic may be considered for evaluation of performance of students in mini project.

2.4 ESEP (Practical/Oral) Examination:

a) The report should be prepared as per the guidelines issued by the University of Mumbai.
b) Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organization's having experience of more than five years approved by the head of Institution.
c) Students shall be motivated to publish a paper based on the work in Conferences/students competitions.
d) End-semester Practical and oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

2.5 Evaluation Rubrics ESEP (Practical/Oral)

Mini Project shall be assessed based on the following rubrics:

a) Quality of problem identification & clarity
b) Innovation & creativity in solution
c) Feasibility, cost effectiveness & societal impact
d) Functionality and correctness of the working model
e) Effective use of engineering tools and skill sets
f) Adherence to engineering standards and norms
g) Individual contribution and teamwork

Written and oral communication skills

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSC601	Machine Learning	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSC601	Machine Learning	20	20	60	--	--	100

Pre- requisite:
AIDSC501: Artificial Intelligence

Program Outcomes:
1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of solutions
4. PO4: Conduct investigation of complex problem
5. PO7: Ethics

Course Objectives:
1. To introduce Machine learning concepts.
2. To explain various supervised Machine Learning Techniques.
3. To describe various unsupervised Machine Learning Techniques
4. To highlight the importance of Dimensionality Reduction and apply it in ML.
5. To introduce the foundational concepts, architectures, and learning mechanisms of Artificial Neural Networks and developing basic neural network–based learning models
6. To provide students with a comprehensive understanding of ensemble learning techniques.

Course Outcomes: Learners will be able to
1. Explain the basics of Machine Learning
2. Apply various supervised Machine Learning Techniques.
3. Apply various unsupervised Machine Learning Techniques.
4. Apply Dimensionality Reduction techniques.
5. Learn and construct basic neural network models using perceptron and multilayer network architectures to solve simple classification and regression problems.
6. Interpret how ensemble learning techniques are applied to real-world problems.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Machine Learning Basics**	**06**	
	1.1	Introduction to Machine learning, Key Terminologies, Types of ML: Supervised learning, Unsupervised learning, Reinforcement learning. Applications of ML, Steps in developing ML applications, Issues in Machine Learning, overfitting and underfitting of model.		**CO1**
	1.2	Introduction to Data Preprocessing, Data Collection, Data Cleaning, Data Splitting for training, cross validation, training error, and testing error.		
		Self-learning Topics: Advanced Data Preprocessing Techniques		
2.0		**Supervise Learning**	**08**	
	2.1	Linear Regression, assumptions of linear regression, Multiple Linear Regression, Logistic Regression. Regularized linear models: Ridge and Lasso regression		**CO2**
	2.2	SVM: Key concepts, working of Algorithm, Mathematical Intuition, soft margin vs hard margin SVM and Linear vs Non-linear SVM. SVM Kernel: Basics. Types of Kernels: Linear, Polynomial, RBF (Gaussian), Sigmoid kernels.		
	2.3	Performance measures for regression: Mean Absolute Error(MAE), Mean Squared Error(MSE), Root Mean Squared Error(RMSE), Root Mean Squared Log Error(RMSLE)		
		Self-learning Topics: Multivariate Linear regression		
3.0		**Unsupervised Learning**	**06**	
	3.1	Introduction, Supervised vs Unsupervised of Learning, Advantages and disadvantages Unsupervised learning, Application, Challenges		**CO3**
	3.2	Concept of Clustering, Hard clustering Vs soft clustering. Introduction to different types of Clustering Methods (Partitioning, Density-Based, Distribution Model-Based Hierarchical). Gaussian Mixture Models, Expectation-Maximization algorithm for clustering		
		Self-learning Topics: Clustering in Real Applications- Image segmentation, document clustering, customer		
4.0		**Dimensionality Reduction**	**05**	
	4.1	Introduction to Dimensionality Reduction, Curse of Dimensionality.		**CO4**
	4.2	Feature Selection (FS): Filter, Wrapper, Embedded approaches.		
	4.3	Feature Extraction (FE): Principal Component Analysis (PCA). Singular Value Decomposition and its applications. FS		

		Self-learning Topics: Autoencoders for Dimensionality		

		Reduction/ PCA vs Autoencoders.		
5.0		**Ensemble Learning**	**08**	
	5.1	Introduction, Need of ensemble learning, Basic Ensemble Techniques: Averaging methods, Voting methods		**CO5**
	5.2	Types of Ensemble Techniques: Bagging, Boosting, Stacking, Blending.		
	5.3	Bagging Methods: Random Forest. Boosting Methods: AdaBoost, XGBoost.		
		Self-learning Topics: Out-of-Bag (OOB) Error Estimation		
6.0		**Introduction to Artificial Neural Networks**	**06**	
	6.1	Biological Neuron, Artificial Neural Networks: basic concepts, evolution of neural network, biological neuron vs artificial neuron, McCulloch–Pitts model, ANN architecture, bias, weights and activation functions (binary, bipolar, sigmoid, Tanh, ReLU), perceptron model, linearly separable and non-separable patterns, and limitations of the perceptron.		**CO6**
	6.2	Perceptron Learning Rule, Delta Learning Rule (LMS / Widrow–Hoff), introduction to multilayer perceptron (MLP), forward propagation, Gradient Descent, the backpropagation algorithm.		
		Self-learning Topics: Applications of ANN in classification and regression tasks, SOM		
		Total	**39**	

Textbooks:

1. Peter Harrington, Machine Learning in Action, DreamTech Press, first Edition, 2012.
2. Nathalie Japkowicz & Mohak Shah, Evaluating Learning Algorithms, Cambridge, first Edition, 2011.
3. Ethem Alpaydın, Introduction to Machine Learning, MIT Press, fourth Edition, 2020.
4. Peter Flach, Machine Learning, Cambridge University Press, second Edition, 2022.
5. Tom M. Mitchell, Machine Learning, McGraw Hill, second Edition, 1997.
6. Zhi-Hua Zhou, Ensemble Methods, CRC Press, first Edition, 2012.

Reference books:

1. Han, Kamber, Pei, Data Mining: Concepts and Techniques, fourth Edition, Morgan Kaufmann, 2022.
2. Margaret. H. Dunham, Data Mining: Introductory and Advanced Topics, first

   Edition, Pearson, 2006.
3. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, first Edition,

   MIT Press, 2012.
4. Samir Roy and Chakraborty, Introduction to Soft Computing, Latest Pearson Edition (2024).
5. Richard Duda, Peter Hart, David G. Stork, Pattern Classification, second Edition, Wiley, 2000.

Online References:

1. Data sets for Machine Learning algorithms - https://www.kaggle.com/datasets
2. Machine Learning repository - https://archive.ics.uci.edu/ml/index.php
3. Machine Learning from Coursera - https://www.coursera.org/collections/machine-l
4. https://toward sd atascience.com/machine-learning/home
5. NPTEL course -Introduction to Machine Learning – IITKGP, Prof. Sudeshna Sar https://onlinecourses.nptel.ac.in/noc21_cs85/preview

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theor	Practical	Tutorial	Theor	Practic	Tutorial	Total
AIDSC602	Big Data Analytics	3			3			3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessmen		ESE$			
		ISE	MSE				
AIDSC602	Big Data Analytics	20	20	80			100

Pre-requisite:

1. Basic Programming Skills C, C++.

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO6: The Engineer and the World
6. PO7: Ethics
7. PO8: Individual and Collaborative Team work
8. PO9: Communication
9. PO10: Project Management and Finance
10. PO11: Life-long Learning.

Course Objectives:

1. To provide an overview of an exciting, growing field of Big Data analytics.
2. To discuss the challenges traditional data mining algorithms face when analyzing Big Data.
3. To introduce the tools required to manage and analyze big data such as Hadoop, NoSql MapReduce.
4. To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
5. To introduce to the students several types of big data, like social media, web graphs and data streams.
6. To enable students to have skills that will help them to solve complex real-world problems in decision support

Course Outcomes:

1. Explain the motivation for big data systems and identify the main sources of Big Data in the real world.
2. Demonstrate an ability to use frameworks like Hadoop, NOSQL to store, retrieve, and process Big Data for Analytics efficiently.
3. Implement several data-intensive tasks using the MapReduce Paradigm.

4. Apply several newer algorithms for Clustering, Classifying, and finding associations in Big Data.
5. Design algorithms to analyze Big data like streams, Web Graphs, and Social Media data.
6. Design and implement successful Recommendation engines for enterprises.

Module No.	Unit No.	Topics	Hrs	Mapped to Course Outcome
1.0	**1**	**Introduction to Big Data**	**05**	
	1.1	Introduction to Big Data, Characteristics, and Types of Big		**CO1**
	1.2	Traditional vs. Big Data business approach, Big Data Challenges, Examples of Big Data in Real Life, Big Data Applications.		
	1.3	Core Hadoop Components; Hadoop Ecosystem; Overview of: Apache Spark, Pig, Hive, HBase, Sqoop, Oozie, Mahout, Zookeeper, Flume.		
		Self-learning Topics: Big Data vs Data Warehouse, Analyzing Data with UNIX tools.		
2.0		**Big Data Frameworks and NOSQL**	**06**	
	2.1	HDFS (Hadoop Distributed File System) Overview & Architecture, Processing data with Hadoop, Managing resources and applications with Hadoop YARN (Yet Another Resource Negotiator).		
	2.2	Introduction to NoSQL - Features and Types- Advantages & Disadvantages - Application of NoSQL. NoSQL data architecture patterns: Key-value stores, Graph stores, Column family (Bigtable) stores, Document stores,		
	2.3	Overview of shared-nothing architecture, selection of appropriate data distribution models, peer-to-peer system design principles, and the four keyways NoSQL databases address Big Data challenges.		
		Self-learning Topics: Hands-on with HDFS and YARN using a Local Hadoop Setup, Introduction to MongoDB Compass / Atlas for NoSQL Practice, Data Distribution Strategies in NoSQL Systems		
3.0		**MapReduce Paradigm**	**06**	
	3.1	MapReduce: The Map Tasks, Grouping by Key, The Reduce Tasks, Combiners, Details of MapReduce Execution, Coping With Node Failures.		**CO3**
	3.2	Matrix Multiplication, Matrix Multiplication with One MapReduce Step. Illustrating the use of MapReduce with use of real-life databases and applications.		
	3.3	Relational-Algebra Operations, Computing Selections b		

		MapReduce, Computing Projections by MapReduce, Union, Intersection, and Difference by MapReduce, Computing Natural Join by MapReduce, Grouping and Aggregation by MapReduce		
		Self-learning Topics: Optimization Techniques in MapReduce, MapReduce vs Spark: Conceptual Comparison, Writing and Running MapReduce Jobs in Python (MRJob / Hadoop Streaming)		
4.0		**Mining Big Data Streams**	**07**	
	4.1	The Stream Data Model: A DataStream-Management System, Examples of Stream Sources, Stream Queries, Issues in Stream Processing.		**CO4**
	4.2	Sampling Data in a Stream: Sampling Techniques. Filtering Streams: The Bloom Filter Counting Distinct Elements in a Stream: The Distinct Count Problem, The Flajolet-Martin Algorithm.		
	4.3	Combining Estimates and Space Requirements. Counting Ones in a Window: The Cost of Exact Counts, The Datar-Gionis- Indyk - Motwani Algorithm, Query Analysis, Decaying Windows.		
		Self-learning Topics: Reservoir sampling, uniform sampling, and trade-offs in accuracy vs memory. Analyze Space-Efficient Approximate Counting		
5.0		**Finding Similar Items and Clustering**	**07**	
	5.1	Distance Measures: Definition of a Distance Measure, Euclidean Distances, Jaccard Distance, Cosine Distance, Edit Distance, Hamming Distance.		**CO5**
	5.2	Frequent Pattern Mining: Handling Larger Datasets in Main Memory Basic Algorithm of Park, Chen, and Yu. The SON Algorithm and MapReduce.		
	5.3	Clustering Algorithms: CURE Algorithm. Canopy Clustering, Clustering with MapReduce		
		Self-learning Topics: Sliding-Window Models in Streaming Analytics, Introduction to Apache Flink or Spark Structured Streaming		
6.0		**Real-Time Big Data Models**	**08**	
	6.1	Link Analysis: PageRank Definition, Structure of the web, dead ends, Using Page rank in a search engine, Efficient computation of Page Rank: PageRank Iteration Using MapReduce.		**CO6**
	6.2	Hubs and Authorities, HITS Algorithm. Mining Social Network Graphs: Social Networks as Graphs, Types, Clustering of Social Network Graphs,		

		Total	**39**	

Textbooks:

1. Radha Shankarmani, M Vijayalakshmi,"Big Data Analytics", Wiley Publications
2. Anand Rajaraman and Jeff Ullman "Mining of Massive Datasets", Cambridge University Press.
3. Alex Holmes ―Hadoop in Practice‖, Manning Press, Dreamtech Press.
4. Dan Mcary and Ann Kelly ―Making Sense of NoSQL‖ – A guide for managers and the rest of us, Manning Press.

Reference books:

1. Bill Franks , ―Taming The Big Data Tidal Wave: Finding Opportunities In Huge Data Streams With Advanced Analytics‖, Wiley
2. Jared Dean, ―Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners‖, Wiley India Private Limited, 2014.
3. Jiawei Han and Micheline Kamber, ―Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 3rd ed, 2010.
4. Ronen Feldman and James Sanger, ―The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, 2006.

Online References:

1. NPTEL Course on Big Data Computing: https://youtu.be/r5k-_RLIpuA
2. NPTEL Course on Data Science & Big Data https://youtu.be/Ue8bXCDRqYM

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practica	Tutorial	Theory	Practical	Tutorial	Total
AIDSC603	Software Engineering and Project Management	2	--	--	2	--	--	2

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessmen		ESE$			
		ISE	MSE				
AIDSC603	Software Engineering and Project Management	15	15	45	--	--	75

Pre-requisite:

1. Basic Programming Skills C, C++.

Program Outcomes Addressed

1 PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO6: The Engineer and the World
6. PO7: Ethics
7. PO8: Individual and Collaborative Team work
8. PO9: Communication
9. PO10: Project Management and Finance
10. PO11: Life-long Learning.

Course Objectives:

1. To provide the knowledge of software engineering discipline.
2. To understand software requirements and analysis.
3. To do planning and apply scheduling in software engineering problem.
4. To apply analysis, and develop software solutions for the given problem.
5. To understand need of project management and project management life cycle.
6. To demonstrate and evaluate real-time projects using software engineering principles and ensure quality through testing.

Course Outcomes:

1. Identify requirements, analyze and prepare models.
2. Apply cost estimation techniques and fundamental design engineering principles to support project planning and decision-making.

3. Apply testing and assure quality in software solutions.
4. Analyze need of project management and project management life cycle.
5. Schedule and track the progress of the projects.
6. Apply project implementation, monitoring, and closure techniques to ensure successful execution and completion of software projects.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction to Software Engineering and Requirements**	**05**	
	1.1	Nature of Software and Software Engineering, SDLC.		**CO1**
	1.2	Generic Process Model, Prescriptive Process Models: V-Model, Incremental, Agile Process and Agility		
	1.3	Software Requirements: Functional and Non-Functional Requirements. Requirement Engineering Process, Feasibility Studies, Validation and Management. Software Prototyping, Documentation and Software Requirement		
		Self-learning Topics: Extreme Programming (XP),		
2.0		**Cost Estimation and Design Engineering**	**05**	
	2.1	Project Estimation: LOC, FP, Empirical Estimation Models - COCOMO II Model.		
	2.2	Design Concepts- The Design Model, Modularity- Cohesion, and coupling. Architectural Design: Patterns, Design Decisions, Application Architectures.		
	2.3	Designing Class-based Components, The Golden Rules of UI Design, Interface Design Steps & Analysis, Design Evaluation.		
		Self-learning Topics. Behavioral UML Diagrams- State diagram, Activity diagram, Use-case diagram,		
3.0		**Software Testing and Software Risk, Configuration Management**	**04**	
	3.1	Software testing Strategies: A strategic approach, Test strategies for Conventional software, Unit Testing, Integration testing, Validation Testing, System testing.		**CO3**
	3.2	Testing Conventional Applications: White-box testing: Basis path, Control structure testing, black-box testing: Graph based, Equivalence, Boundary Value.		
	3.3	Software Configuration Management (SCM) repositories & processes, Software Quality Assurance (SQA) task and plan, Formal Technical Review (FTR) & Walkthrough.		
		Self-learning Topics: Version control using Github, Green SCM Practices.		

4.0		**Project Management Basics**	**04**	
	4.1	Definition, characteristics, and types of projects, Need for project management, 4Ps, W5HH principle.		**CO4**
	4.2	Project lifecycle, ITPM overview, Project feasibility (technical, economic, social, environmental), RFP, PMBOK knowledge areas.		
	4.3	Project identification, idea generation, stakeholder analysis, Organizational structures (Functional,		
		Self-learning Topics: Foundations of Strategic Decision-Making in Complex Initiatives, Methods for Evaluating Organizational Value, Benefits, and Long-Term Outcomes, Project charter and project scope.		
5.0		**Project Planning & Scheduling**	**04**	
	5.1	Work Breakdown Structure (WBS), task definition, action plan, timeline charts, baseline plan, and Gantt charts for schedule representation.		**CO5**
	5.2	Network diagrams (AON & AOA), CPM and PERT (basic numerical), feasibility and project appraisal (technical, economic, financial), cost estimation (direct and indirect), Risk management: Identification, assessment, and mitigation strategies with qualitative risk analysis. RMMM.		
		Self-learning Topics: Principles of Effective Project Planning and Creating Realistic Schedules, Basics of Project Risk Culture, Resource Optimization, and Performance Tracking, schedule tracking and Earned Value Analysis.		
6.0		**Project Implementation, Monitoring & Closure**	**04**	
	6.1	Project team roles: PI, CO-PI, associates, team members; resource mobilization; procurement basics; vendor selection; quality basics including QC, QA, and an introduction to		**CO6**
	6.2	Project execution involving communication, coordination, and conflict management; project monitoring including progress reports, KPIs, and review meetings.		
		Self-learning Topics: Understanding Project Team Roles and Quality Management, Project Execution, Monitoring, and Control Techniques, Ethics, Risk Response, and Social–Environmental Impact in Projects. Evaluation and audits; risk response basics; code of ethics in project management; dilemmas and decision making.		
		Total	**26**	

Textbooks:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach", 9th edition , McGraw-Hill Publications, 2019.
2. Ian Sommerville, "Software Engineering", 9th edition, Pearson Education, 2011.
3. Ali Behfrooz and Fredeick J. Hudson, "Software Engineering Fundamentals", Oxford University Press, 1997.
4. Grady Booch, James Rambaugh, Ivar Jacobson, "The unified modeling language user guide", 2nd edition, Pearson Education, 2005.
5. Meredith, J. R., & Mantel, S. J. , "Project Management: A Managerial Approach" (11th ed.). Wiley India, 2014.
6. E. W. Larson and C. F. Gray, "Project Management: The Managerial Process", 8th ed. New York: McGraw Hill Education, 2020.
7. Highsmith, J, Agile Project Management: Creating Innovative Products (2nd ed.). Pearson Education, 2009.

Reference books:

1. bJalote, "An integrated approach to Software Engineering", 3rd edition, Springer, 2005.
2. Rajib Mall, "Fundamentals of Software Engineering", 5th edition, Prentice Hall India, 2014.
3. Ugrasen Suman, "Software Engineering – Concepts and Practices", Cengage Learning, 2013.
4. P. Chandra, Projects: Planning, Analysis, Selection, Financing, Implementation, and Review. New Delhi: Tata McGraw Hill Education, 2009.
5. M. P. Spinner, Elements of Project Management. USA: Prentice Hall, 1997.

Online References:

1. Software Engineering - Course
2. nptel.ac.in/courses/106101061

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 15 marks.
- ISE 15 marks = 05 marks for attendance + 10 marks for activities.

MSE:

- To be conducted as a written examination for 15 marks (on 50% syllabus)

End Semester Examination:

ESE of duration 02 hours is 60 marks and scaled to 45.

1. Question paper will comprise 03 questions.
2. Question 1 (15 marks):- Solve any 03 out of 04. All questions carry 05 marks each.
3. Question 2 (30 marks):- Solve any 03 out of 05. All questions carry 10 marks each.
4. Question 3 (15 marks):- Solve any 03 out of 04. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theo	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 6011	Robotics and Applications	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessmen		ESE$			
		ISE	MSE				
AIDSPEC 6011	Robotics and Applications	20	20	60	--	--	100

Pre- requisite:

1. Basic knowledge of mathematics.
2. Programming skills: C, C++.

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO9: Communication
6. PO11: Life-long Learning.

Course Objectives:

1. To introduce the fundamental concepts of robotics.
2. To develop understanding of robot kinematics.
3. To provide knowledge of robot dynamics and actuation systems.
4. To familiarize learners with robotic sensors and perceptions.
5. To build competence in robot control, motion planning, and mobile robotics.
6. To develop hands-on understanding of robot programming.

Course Outcomes: Learners will be able to:

1. Explain the fundamentals of robotics, robot classification, anatomy, and diverse application areas across industries.
2. Apply kinematic modeling techniques including transformation matrices, forward and inverse kinematics, and DH parameters for robotic manipulators.
3. Analyze robot dynamics and actuation mechanisms using principles of force, torque, Newton–Euler, and Lagrangian methods and evaluate different actuator technologies.
4. Evaluate and integrate various robotic sensors and perception systems, including vision processing and basic sensor fusion concepts.
5. Design and analyze robot control strategies and motion planning algorithms and interpret mobile robot kinematics and navigation concepts.
6. Develop and implement basic robot programming using ROS, simulation tools, and AI/ML techniques for robotic applications.

Module No.	Unit no.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction to Robotics**	**06**	
	1.1	Definition, scope, importance of robotics, History and evolution of robotic systems		**CO1**
	1.2	Classification of robots: industrial, service, mobile, humanoid, Anatomy of a robot: links, joints, end-effectors, DOF, workspace		
	1.3	Industrial applications: manufacturing, healthcare, agriculture, defense, logistics, Overview of automation & Industry 4.0		
		Self-learning Topics: Soft robotics: materials, applications, emerging trends		
2.0		**Robot Kinematics**	**07**	
	2.1	Coordinate frames and transformation matrices		**CO2**
	2.2	Forward kinematics, Inverse kinematics (analytical concepts)		
	2.3	Denavit–Hartenberg (D–H) parameters, Homogeneous transformations, Kinematic redundancy, singularities		
		Self-learning Topics: Jacobian matrix: derivation, geometric vs analytical Jacobian, Velocity kinematics & differential kinematics		
3.0		**Robot Dynamics & Actuation**	**06**	
	3.1	Concepts of force, torque, inertia, Dynamic modeling of manipulators		
	3.2	Newton–Euler and Lagrangian formulations (conceptual)		

	3.3	Actuation systems: Electric- DC motors, servo motors, stepper motors Pneumatic & Hydraulic Systems.		**CO3**
	3.4	Power transmission: gears, belts, harmonic drives.		
		Self-learning Topics: Dynamic simulation using MATLAB / Gazebo		
4.0		**Sensors, Perception & Vision Systems**	**07**	
	4.1	Sensors: Proximity, IR, ultrasonic, tactile, force/torque sensors.		**CO4**
	4.2	Encoders, IMU, gyroscope, LIDAR and Radar basics.		
	4.3	Perception & Vision: Image acquisition and preprocessing, Feature extraction, segmentation.		
	4.4	Object detection and recognition, Basics of sensors, Sensor types and working.		
		Self-learning Topics: 3D perception: depth cameras (Intel RealSense), stereo vision.		
5.0		**Control system & Mobile Robotics**	**07**	
	5.1	Introduction to control systems, Mathematical modelling basics, Standard test signals, Time domain response characteristics, frequency domain basics.		**CO5**
	5.2	Robot: Open-loop and closed-loop control, PID controller: concept and tuning, Joint space vs task space control.		
	5.3	Mobile Robotics: Wheeled robot models – differential drive, skid steer, Kinematics of mobile robots, Basics of SLAM (mapping & localization), Autonomous navigation architecture.		
		Self-learning Topics: Motion Planning: Trajectory generation, Path planning algorithms: BFS, DFS, A*, Dijkstra, RRT (conceptual).		
6.0		**Robot Programming, AI in Robotics & Applications**	**06**	
	6.1	Robot Programming: Basics of Robot Operating System (ROS), ROS nodes, topics, publishers/subscribers.		**CO6**
	6.2	Simulation tools: Gazebo, Webots, CoppeliaSim, Programming robots using Python/C++.		
	6.3	AI & Applications: Role of AI & ML in robotics, Reinforcement learning for robot decision-making.		
	6.4	Collaborative robots (CoBots), Use cases: smart factories, autonomous vehicles, agriculture, logistics, defense.		

		Self-learning Topics: Ethics in robotics & AI, I-based motion planning (Neural Motion Planner basics)		
		Total	**39**	

Textbooks:

1. S. K. Saha, Introduction to Robotics, 2nd ed. New Delhi, India: McGraw-Hill Education, 2014.
2. M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. Hoboken, NJ, USA: Wiley, 2005.
3. Robert Shilling, "Fundamentals of Robotics - Analysis and control, Prentice Hall of India, 2009
4. Saeed Benjamin Niku, "Introduction to Robotics – Analysis, Control, Applications", Wiley India Pvt. Ltd., Second Edition, 2011

Reference books:

1. John J. Craig, "Introduction to Robotics – Mechanics & Control", Third Edition, Pearson Education, India, 2009.
2. Mark W. Spong, Seth Hutchinson, M. Vidyasagar, "Robot Modeling & Control", Wiley India Pvt. Ltd., 2006.
3. Mikell P. Groover et.al, "Industrial Robots-Technology, Programming & applications", McGraw-Hill, New York, 2008.
4. S. R. Deb and Sankha Deb, "Robotics Technology and Flexible Automation" , Second Edition. TMH

Online References:

1. ROS Tutorials – Official Documentation https://wiki.ros.org/ROS/Tutorials
2. NPTEL Courses: Robotics by IIT Kanpur, Robot Kinematics and Dynamics by IIT Kharagpur
3. Coursera / edX (Free audit mode): Robotics Specialization (University of Pennsylvania), Modern Robotics (Northwestern University)
4. MATLAB Robotics Toolbox (Peter Corke) https://petercorke.com/toolboxes/robotics-toolbox/
5. Gazebo & Webots Official Documentation https://gazebosim.org, https://cyberbotics.com

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC 6012	Blockchain Technology	3	-	-	3	-	-	3

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSPEC6012	Blockchain Technology	20	20	60	-	-	100

Pre- requisite:

1. Cryptography and Distributed Systems

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions:
4. PO4: Conduct Investigations of Complex Problems:
5. PO5: Engineering Tool Usage
6. PO11: Life-Long Learning
7. PSO1: Problem solving

Course Objectives:

1. To get acquainted with the concept of Distributed ledger system and Blockchain.
2. To learn the concepts of consensus and mining in Blockchain through the Bitcoin network
3. To understand Ethereum and develop-deploy smart contracts using different tools and frameworks.
4. To understand permissioned Blockchain and explore Hyperledger Fabric.
5. To understand different types of crypto assets.

Course Outcomes: Learners will be able to

1. Describe the basic concept of Blockchain and Distributed Ledger Technology-
2. Interpret the knowledge of the Bitcoin network, nodes, keys, wallets and transactions
3. Implement smart contracts in Ethereum using different development frameworks.
4. Develop applications in permissioned Hyperledger Fabric network.
5. Interpret different Crypto assets and Crypto currencies.
6. Analyze the use of Blockchain with AI, IoT and Cyber Security using case studies.

Module No.	Unit No.	Topics	Hrs	Mapped to Course Outcome
1.0		**Introduction to Blockchain**	**05**	
	1.1	Distributed Ledger Technologies: Introduction to Blockchain: History, evaluation, fundamentals concept, components, types.		**CO1**
	1.2	Block in a Blockchain: Structure of a block, The Genesis Block, Linking Blocks in the Blockchains, Merkle tree, Tokenized blockchain, need of sidechain		
		Self-learning Topics: Working and example of Sidechain		
2.0		**Consensus Protocol and Bitcoin blockchain**	**06**	
	2.1	Consensus: Byzantine, Generals problem, consensus algorithm: PoW, PoS, PoA, PoET, LPoS, pBFT, Proof-of- Burn(PoB), Life of a miner, Mining difficulty, Mining pool & its methods		**CO2**
	2.2	Bitcoin: Bitcoin mining, hashcash, Block propagation and relay, bitcoin scripts, transaction in the bitcoin network		
		Self-learning Topics. Syntax and examples of bitcoin scripts		
3.0		**Ethereum and Smart Contracts**	**08**	
	3.1	Ethereum: History, components, Architecture of Ethereum, Consensus, Miner and mining node, Ethereum virtual machine, Ether, Gas, Transaction, Accounts, Patricia Merkle Tree, Swarm, Whisper and IPFS, complete transaction working and steps in Ethereum, Case study of Ganache for Ethereum blockchain, Exploring etherscan.io and ether block structure, Comparison between Bitcoin and Ethereum		**CO3**
	3.2	Smart Contracts: history, characteristics, working of smart contracts, types, Oracles, structure and limitations Solidity programming: setup tools and installation, Basics functions, Visibility and Activity Qualifier, Ethereum networks, Solidity files and structure of Contracts, data types, storages, array, functions, Developing and executing smart in Ethereum, Smart Contracts Use Cases, Opportunities and Risks		
		Self-learning Topics: Other programming languages for smart contract		
4.0		**Private and Consortium blockchains**	**09**	
	4.1	Introduction to Private and Consortium: Key Characteristics, needs, Examples of Private and Consortium blockchains, smart contracts in Private blockchains		**CO4**

	4.2	Introduction to Hyperledger, Tools and Frameworks, Hyperledger Fabrics, Comparison between Hyperledger		
		Fabric & other Technologies, Hyperledger Platform, Paxos and Raft consensus, Ripple and Corda blockchains, Byzantine Faults: Byzantine Fault Tolerant (BFT) and Practical BFT		
		Self-learning Topics: Recent Hyperledger tools		
5.0		**Cryptocurrencies and digital tokens**	**06**	
	5.1	Cryptocurrency basics, types, usage, ERC20 and ERC721 Tokens, comparison between ERC20 and ERC721, ICO: basics and related terms, launching an ICO, pros and cons, evaluation and platforms, STO, Different Crypto currencies, Defi, Metaverse, Types of cryptocurrencies, Bitcoin, Altcoins and Tokens (Utility and Security), Cryptocurrency wallets: Hot and Cold wallets, Cryptocurrency usage, Transaction in Blockchain, UTXO and double spending problem		**CO5**
		Self-learning cryptocurrency Topics: Security and attacks on cryptocurrency		
6.0		**Blockchain applications, Tools and case studies**	**05**	
	6.1	Applications of Blockchain: Various domains including Education, Energy, Healthcare, real estate, logistics, supply chain. Tools: Corda, Ripple, Quorum and other Emerging Blockchain Platforms, Case Study on any of the Blockchain Platforms		**CO6**
		Self-learning Topics: Solana blockchain tools		
		Total	**39**	

Textbooks:

1. C. Subramanian, A. A. George, A. K. A. Abhillash, and M. Karthikeyan, Blockchain Technology, 1st ed. Hyderabad, India: Universities Press, 2020.
2. R. Modi, Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, 2nd ed. Birmingham, U.K.: Packt Publishing, 2022.
3. A. Kumar, Hyperledger Fabric In-Depth: Learn, Build and Deploy Blockchain Applications Using Hyperledger Fabric, 1st ed. New Delhi, India: BPB Publications, 2020.
4. C. Burniske and J. Tatar, Cryptoassets: The Innovative Investor's Guide to Bitcoin and Beyond, 1st ed. New York, NY, USA: McGraw-Hill Education, 2017.
5. A. M. Antonopoulos and G. Wood, Mastering Ethereum: Building Smart Contracts and DApps, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2025.

Reference books:

1. A. M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.
2. A. M. Antonopoulos and G. Wood, Mastering Ethereum: Building Smart Contracts and DApps, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2025.
3. K. Saurabh and A. Saxena, Blockchain Technology: Concepts and Applications, 1st ed. Hoboken, NJ, USA: Wiley, 2018.
4. A. Lewis, The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology That Powers Them, 1st ed. Coral Gables, FL, USA: Mango Publishing, 2018.

Online References:

1. NPTEL courses: Blockchain and its Applications, Blockchain Architecture Design and Use Cases.
2. https://ethereum.org/en/
3. https://www.truffl esuite.com/tutorials
4. https://hyperledger-fabric.readthedocs.io/en/release -2.2/
5. Blockchain demo: https://andersbrownworth.com/blockchain/
6. Blockchain Demo: Public / Private Keys & Signing: https://andersbrownworth.com/blockchain/public -private-keys/

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPE6013	Image and Video Processing	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE[$]			
		ISE	MSE				
AIDSPE6013	Image and Video Processing	20	20	60	--	--	100

Pre-requisite:
AIDSC301 -Applied Mathematics III, AIDSC401 -Applied Mathematics IV

Program Outcomes addressed:
1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design / Development of Solutions.
4. PO11: Life- Long Learning
5. PSO2: Implement Artificial Intelligence and Data Science techniques such as data analysis, machine learning, data mining, neural networks and design efficient algorithms using emerging technologies in the field.

Course Objectives:
1. To understand fundamentals of digital image and video formation, representation, and standards.
2. To study various image enhancement techniques in spatial and frequency domains.
3. To learn techniques for image segmentation and morphological processing.
4. To understand and apply different image transform techniques.
5. To explore lossless and lossy image compression methods.
6. To study motion estimation techniques and their applications in video processing.

Course Outcomes: Learners will be able to
1. Describe the fundamentals of digital images, videos, and their standard formats.
2. Apply spatial and frequency domain techniques for image enhancement.
3. Apply segmentation and morphological operations for feature extraction.
4. Apply various transforms for image analysis and representation.
5. Compare different image compression techniques for efficient storage and transmission.
6. Analyze motion estimation methods and their role in video coding applications.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Digital Image and Video Fundamentals**	**05**	**CO1**
	1.1	Introduction to Digital Image, Digital Image Processing System, Sampling and Quantization, Feature-extraction, Representation of Digital Image, Connectivity, Image File Formats: BMP, TIFF and JPEG. Color Models (RGB, HSI, YUV)		
	1.2	Fundamentals of Digital Video, Chroma Sub-sampling, Video Standards, Composite and Component Video Systems, Digital Video Formats and Application.		
		Self-Learning: Applications of Digital Image and Video Processing in Real Life, Evolution of Image and Video Storage Formats		
2.0		**Image Enhancement**	**08**	**CO2**
	2.1	Point Processing: Digital Negative, contrast stretching, thresholding, gray level slicing, bit plane slicing, log transform and power law transform. Spatial Domain: Neighborhood Processing: Averaging filters, order statistics filters, high pass filters and high boost filters		
	2.2	Frequency Domain: DFT for filtering, Ideal, Gaussian and Butterworth filters for smoothening and sharpening		
	2.3	Histogram Modeling: Histogram equalization and histogram specification		
		Self-Learning: Real-Time Image Enhancement Techniques Used in Mobile Cameras, Effect of Different Filters on Image Sharpness and Noise Reduction		
3.0		**Image Segmentation & Morphological Operations**	**07**	**CO3**
	3.1	Segmentation: Discontinuities (point, Line, Edge) Image Edge detection using Robert, Sobel, Prewitt and Laplacian Mask, Deep Segmentation, Evaluation Metrics		
	3.2	Region Oriented Segmentation: Region growing by pixel Aggregation, Region Splitting and Merging		
	3.3	Morphology: Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transformation.		
		Self-Learning: Real-Time Applications of Image Segmentation in Medical Imaging and Object Detection, Role of Morphological Processing in Document Image Analysis		
4.0		**Image Transforms**	**08**	**CO4**
	4.1	Introduction to Unitary Transforms, Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT)		

	4.2	Discrete Hadamard Transform (DHT), Fast Hadamard Transform (FHT), Discrete Cosine Transform (DCT), Inverse DCT		
		Self-Learning: Applications of DCT in JPEG Image Compression, Comparison of DFT, DHT, and DCT in terms of energy compaction and computation time		
5.0		**Image Compression**	**07**	**CO5**
	5.1	Introduction, Redundancy, Fidelity Criteria		
	5.2	Lossless Compression Techniques: Run length Coding, Arithmetic Coding, Huffman Coding		
	5.3	Lossy Compression Techniques: Improved Gray Scale Quantization, Vector Quantization, MPEG		
		Self-Learning: Applications of Image Compression in Real-Time Systems, Role of Transform Coding (DCT/DWT) in Image Compression		
6.0		**2D Motion Estimation in Videos**	**04**	**CO6**
	6.1	Motion Estimation Techniques: Optical Flow, Pixel-based Motion Estimation, Region-based Motion Estimation, Multi-resolution Motion Estimation		
	6.2	Applications of Motion Estimation: Application of Motion Estimation in Video Coding: Compression Efficiency, Temporal Prediction		
		Self-Learning: Multi-Resolution / Pyramid-Based Motion Estimation, Region-Based Motion Estimation and Object-Level Tracking, Applications of Motion Estimation in Video Coding		
		Total	**39**	

Textbooks:

1. Gonzalez, Rafael C. & Woods, Richard E. “Digital Image Processing “. Pearson, 4th Edition (2018).
2. Wang, Yao; Ostermann, Jörn; Zhang, Ya-Qin, “Video Processing & Communications”, Prentice Hall, 1st Edition (2002).
3. Jayaraman, S.; Esakkirajan, S.; Veerakumar, T. “Digital Image Processing”, Tata McGraw-Hill Education, 3rd Edition, 2009. Central University of Punjab Catalog
4. Jain, Anil K., “Fundamentals of Digital Image Processing”, Prentice Hall (India), 1st Edition 1989
5. Sridhar, S. “Digital Image Processing”, Oxford University Press, 2nd Edition 2012 .

Reference books:

1. Tekalp, A. Murat, “Digital Video Processing”, Prentice Hall, 2nd Edition (2015).
2. Forsyth, David A.; Ponce, Jean, “Computer Vision: A Modern Approach”, Pearson, 2nd Edition (2011).
3. Pakhira, Malay K. “Digital Image Processing and Pattern Recognition”, Prentice Hall of India, 1st Edition 2011.
4. Chanda, Bhabatosh; Majumder, Dwijesh Dutta, “Digital Image Processing & Analysis”, PHI Learning / Prentice Hall of India, 2nd Edition 2011.
5. Sayood, Khalid, “Introduction to Data Compression”, Morgan Kaufmann, 5th Edition (2017).

Online References:

Useful Links

1. https://onlinecourses.nptel.ac.in/noc22_ee116/preview
2. https://www.coursera.org/learn/introduction-image-processing
3. https://www.edx.org/course/digital-video-processing-fundamentals
4. https://onlinecourses.nptel.ac.in/noc21_ee20/preview

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1 (20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3 (20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEC6014	Augmented Reality and Virtual Reality	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course		ESE$			
		ISE	MSE				
AIDSPEC 6014	Augmented Reality and Virtual Reality	20	20	60	--	--	100

Pre- requisite:

1. Computer Graphics

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering Tool Usage

Course Objectives:

1. To understand the need and significance of Virtual Reality.
2. To explore the concepts of Virtual reality and develop 3D virtual environments.
3. To understand the technical and engineering aspects of virtual reality systems.
4. To analyze various techniques for applying virtual reality.
5. To provide a foundation to the fast-growing field of AR and make the students aware of the various AR devices.

Course Outcomes: Learners will be able to

1. Describe how VR systems work and list the applications of VR
2. Elaborate geometric presentation of the virtual world and its operations.
3. Explain the concepts of motion and tracking in VR systems.
4. Design and implementation of the hardware that enables VR systems to be built.
5. Describe how AR systems work and analyze the hardware requirements of AR.
6. Analyze and understand the working of various state of the art AR devices.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0		**Introduction to Virtual Reality**	**05**	
	1.1	What is virtual reality? The beginnings of VR, VR paradigms, Collaboration, Virtual reality systems, Representation, User interaction		**CO1**
		Self-learning Topics: Feedback mechanisms in VR environment		
2.0		**The Geometry of Virtual Worlds**	**06**	
	2.1	Geometric Models, Changing Position and Orientation, Axis-Angle Representations of Rotation, Viewing Transformations, Chaining the Transformations		**CO2**
		Self-learning Topics: Light House approach, Case Study on various models		
3.0		**Motion in Real and Virtual Worlds**	**06**	
	3.1	Velocities and Accelerations, The Vestibular System, Physics in the Virtual World, Mismatched Motion and Vection		**CO3**
		Self-learning Topics: Rendering for VR application		
4.0		**Applying Virtual Reality**	**07**	
	4.1	Virtual reality: the medium, Form and genre, What makes an application a good candidate for VR, Promising application fields, Demonstrated benefits of virtual reality, More recent trends in virtual reality application development, A framework for VR application development		**CO4**
		Self-learning Topics: Explore tools for UI in VR		
5.0		**Augmented Reality**	**08**	
	5.1	Terminology, Simple augmented reality, Augmented reality as an emerging technology, Augmented reality applications, Marker detection, Marker pose, Marker types and identification: Template markers, 2D bar-code markers, Imperceptible markers: Image markers, Infrared markers,		**CO5**
		Self-learning Topics: Indoor Tracking, Full Body Tracking		
6.0		**AR Development & Applications**	**07**	
	6.1	Edutainment, Medical, Military,		**CO6**
	6.2	Production and Manufacturing, Navigation, Astronomical Observation, E-commerce: Human Factors		
	6.3	Physical Visual Side Effects, Legal Considerations		
	6.4	Moral and Ethical Considerations.		
		Self-learning Topics: Applications of AR/MR in Civil Construction and Architecture		
		Total	**39**	

Textbooks:

1. Steven M. LaVelle," Virtual Reality", Cambridge University Press,1st edition,2015
2. Grigore Burdea, Philippe Coiffet, "Virtual Reality Technology", Wiley India,2nd edition,2018
3. Dieter Schmalsteig and Tobias Hollerer, "Augmented Reality- Principles and Practice", Pearson Education,1st edition, 2016

Reference books:

1. Tony Parisi, "Learning Virtual Reality", 1st edition, O'Reilly,1st edition, 2015
2. Alan Craig and William Sherman," Understanding virtual reality: Interface,2nd edition, 2018 application and design", 2nd Edition, Morgan Kaufmann Publisher.
3. Borko Furht, "Handbook of Augmented Reality", Springer,2011 edition
4. Erin Pangilinan, Steve Lukas, and Vasanth Mohan, "Creating Augmented and Virtual Realities- Theory and Practice for Next-Generation Spatial Computing", O'Reilly Media,1st edition, 2019

Online References:

1. www.nptel.ac.in
2. www.coursera.org

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are 80 marks and scaled to 60.

1. Question paper will comprise 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question 3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC6033	Sensor Technology	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC6033	Sensor Technology	20	20	60	--	--	100

Pre-requisite:

1.FEC 204- Digital Signal Design

Program Outcomes Addressed

1. PO1: Engineering knowledge.
2. PO2: Problem analysis
3. PO11: Lifelong learning.

Course Objectives:

1. To understand the basics of sensors.
2. To categorize various sensors based on physical parameters.
3. To familiarize with MEMS sensors and Actuators
4. To introduce wireless sensing technologies
5. To develop an understanding of signal conditioning using ADC and DAC
6. To provide insight into various sensor applications

Course Outcomes: Learners will be able to

1. Explain the transduction principle of various sensors.
2. Select sensors suitable for the required application
3. Describe the working of MEMS sensors and Actuators
4. Analyze wireless sensing techniques
5. Design the data acquisition system and identify various signal conditioning methods.
6. Create an application using various sensor technologies

Module No.	Unit No.	Topics	Hrs	Mapped to Course Outcome
1.0		**Introduction**	**03**	
	1.1	**Definition:** Need and Role of Sensors		**CO1**
	1.2	Basic characteristics: sensitivity, resolution, accuracy, linearity, drift, range, hysteresis		
	1.3	Criteria to choose a Sensor: Accuracy, Environmental condition, Range Calibration,		
		Repeatability		
	1.4	Classification of Sensors: Active and Passive Sensors, Contact and Non contacted sensors		
		Self-learning Topics: Introduction to Instrumentation Systems: Block diagram (Sensor, Signal Conditioner, Display/Recorder)		
2.0		**Types of Sensors**	**09**	
	2.1	Temperature Sensors: RTD, Thermocouple and Thermistors sensor		**CO2**
	2.2	Proximity Sensors: Inductive (LVDT),Capacitive, Photoelectric and Ultrasonic sensors		
	2.3	Chemical Sensors: Gas, Smoke, Conductivity and pH sensor		
	2.4	Other Sensors: Optical, Infrared (IR), Sound, Motion, Pressure, Level, Moisture, Humidity, Laser, Image and GPS sensor		
		Self-learning Topics: LDR, Photo Diode and its applications		
3.0		**MEMS Sensors and Actuators**	**06**	
	3.1	MEMS SENSORS: General design methodology, techniques for sensing, Pressure sensor, Mass Flow sensor, Acceleration sensor, Gyroscopes, Micro machined microphones.		**CO3**
	3.2	MEMS ACTUATORS: Techniques for actuation, Digital Micro mirror Devices, Micro Machined		
	3.3	Smart Sensors: Architecture of Smart Sensor and Applications		
		Self-learning Topics: Chemical sensors, Taguchi gas sensor, Combustible gas sensor		
4.0		**Wireless Sensing Technologies**	**05**	
	4.1	Bluetooth: Concepts of Pico net, Scatter net, Link types, Network Connection Establishments.		**CO4**
	4.2	ZigBee: Components, Architecture, Network Topologies.		
	4.3	Ultra-Wide Band (UWB), Near Field Communication (NFC) and RFID: technical requirements, components and characteristics		
	4.4	WLAN (Wi-Fi): WLAN Equipment, WLAN topologies, IEEE 802.11 Architecture		

		Self-learning Topics: Wireless Sensor Network		
5.0		**Data Acquisition and Signal Conditioning**	**08**	
	5.1	Fundamentals of Data Acquisition: Analog and Digital data acquisition system with different configurations, Data loggers, Noise and interference		**CO5**
	5.2	Signal Conditioning: Wheatstone Bridge, Flash ADC, R2R DAC		
	5.3	Utilization of Signal conditioning circuits for Temperature, Pressure Optical, Strain gauges, Displacement and piezoelectric Transducers		
		Self-learning Topics: Interfacing and Communication with sensors		
6.0		**Sensor Applications**	**08**	
	6.1	Onboard Automobile sensing system, home appliances sensors, Aerospace Sensors, Sensors for Environmental Monitoring, Biomedical Sensing Applications		**CO6**
	6.2	Radio sensors for industrial applications, Radio Astronomy, Remote Sensing, Ground Penetrating Radars, Underwater sensing, LiDAR		
		Self-learning Topics: Case study on Sensor		
		Total	**39**	

Textbooks:

1. D.V.S. Murthy, "Transducers and Instrumentation", PHI Learning, 2nd Edition, 2013.
2. D. Patranabis, "Sensor and Transducers Prentice Hall", New Delhi, 2013,3rd edition.
3. Ramon Pallas-Areny, John G. Webster, Sensors and Signal Conditioning, 2012, 3rd edition, John Wiley & Sons, New York.
4. Vijay K. Garg, Wireless Communications and Networking, 2015, 2nd edition, Morgan Kaufmann (Elsevier), USA

Reference books:

1. An Introduction to Microelectromechanical Systems Engineering, Nadim Maluf, Kirt Williams, Artech House, 2004.
2. A.K. Sawhney, "A Course in Electrical and Electronic Measurements and Instrumentation", Dhanpatrai & Co., 2015, 20th edition, Dhanpat Rai & Co, New Delhi.
3. Nathan Ida, "Sensors, Actuators and their Interfaces: A Multidisciplinary Introduction", Second Edition, IET Control, Robotics and Sensors Series 127, 2020
4. Jacob Fraden, "Handbook of Modern Sensors Physics, Designs, and Applications, Fifth Edition, Springer, 2015, Springer, New York.

Online References:

1. https://onlinecourses.nptel.ac.in/noc23_ee95/preview

Course Assessment:
ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study, etc., of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
MDMC6043	Cloud Computing and Security	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
MDMC 6043	Cloud Computing and Security	20	20	60	--	--	100

Course pre-requisite: Computer Networks and Operating Systems.
Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design and development of solutions
4. PO6: The Engineer and Society
5. PO7: Environment and Sustainability
6. PO11: Lifelong learning

Course Objectives:

1. To conceptualize cloud computing fundamentals.
2. To gain familiarity with the concept of virtualization.
3. To explore various cloud computing services.
4. To investigate various opensource clouds.
5. To identify risks and provide cloud security.
6. To understand several applications and recent trends in cloud computing.

Course Outcomes:

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

1. Analyze cloud computing service models, deployment models, and architecture.
2. Analyze virtualization techniques and implementation levels.
3. Apply different cloud computing services.
4. Analyze opensource cloud platforms.
5. Apply security measures in cloud computing.
6. Investigate cloud computing applications and recent trends.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1		**Introduction to Cloud Computing**	**04**	**CO1**
	1.1	Definition of Cloud Computing, Need for Cloud computing, Characteristics of Cloud computing, NIST model and cloud cube model.		
	1.2	Cloud deployment models (Public Cloud, Private Cloud, Community Cloud, Hybrid Cloud) and Cloud Service Models (Function, advantages, disadvantages): IaaS, PaaS, SaaS.		
	1.3	Advantages and Disadvantages of Cloud computing.		
		Self-Learning: Comparison between cloud computing and traditional computing (on premise IT infrastructures)		
2		**Virtualization**	**06**	**CO2**
	2.1	Introduction and benefits of virtualization, Virtualization Structures and Implementation Levels, Virtual Machine Monitor (VMM).		
	2.2	Hypervisors- Type I, II (Xen, KVM, VMware ESXi, Hyper-V), Virtualization Mechanisms and Tools.		
	2.3	Binary translation with full virtualization, para virtualization with Compiler Support and hardware assisted virtualization. CPU virtualization, memory virtualization and I/O virtualization, virtualization in multicore processors.		
		Self-Learning: Hypervisor Security Issues & VM Escape Attacks, Virtualization vs Containerization (Docker/Kubernetes), Basics of GPU Virtualization.		
3		**Cloud Web Services**	**12**	**CO3**
	3.1	Introduction to the AWS Cloud, AWS core services by categories. Compute service: Introduction to EC2, Amazon Machine Images, Instance Types. Instance Lifecycle. Storage service: Introduction to S3, working with Buckets, setting bucket security, S3 event and notifications, bucket properties, S3 lifecycle management, working with Elastic Block Store Volumes, and Object Storage Vs Block Storage. Security service: AWS Identity and Access Management (IAM).		
	3.2	Database as a Service: Introduction to Amazon Relational Database Service (RDS), Database Engines, Database Instance Classes, Backup and Recovery, Non-relational (No-SQL) Databases, Types of Non-relational Databases, Introduction to DynamoDB, Features, Partition and Hash keys. Network as a Service: Introduction to virtual private cloud, Subnet, Elastic Network Interfaces, Internet Gateways, Route Tables, Security Groups and network access control lists.		

	3.3	Serverless Computing: Introduction, Working with Serverless environment, Basics of serverless events and functions, AWS Lambda. Monitoring as a service: Introduction to AWS CloudWatch, CloudWatch Metrics, CloudWatch Alarms. Content Delivery Services - Amazon CloudFront Deployment & Management Services: Amazon Elastic Beanstalk, Amazon CloudFormation. Application services: AWS Simple Notification Service (SNS), Simple Queue Service (SQS), and Simple Email Service (SES).		
		Self-learning: Comparison of AWS services with other cloud service platforms like Azure and GCP. Analytics Services - Amazon Elastic MapReduce		
4		**Open-Source Cloud Implementation of OpenStack and Eucalyptus**	**05**	**CO4**
	4.1	OpenStack Cloud Architecture, Features of OpenStack, Components of OpenStack and Mode of Operations of OpenStack		
	4.2	Eucalyptus Architecture, Features of Eucalyptus, Components of Eucalyptus and Mode of Operations of Eucalyptus.		
		Self-learning: Explore different open-source cloud computing platforms and compare them based on different services.		
5		**Cloud Security**	**06**	**CO5**
	5.1	Security overview, cloud security and shared responsibility model, cloud security challenges, Cloud Security Risks and countermeasures, Host security in IaaS, PaaS and SaaS.		
	5.2	Data Security, Data Security Concerns, Data Confidentiality and		
		Encryption, Data Availability, Data Integrity, Cloud Storage Gateways, Cloud Firewall, Introduction to AWS Key Management Service (KMS)		
	5.3	AAA Administration for Clouds, AAA model, SSO for Clouds, Authentication management and Authorization management in clouds, Accounting for Clouds Resource utilization.		
		Self-learning: Comparison of Cloud Security in AWS/Microsoft Azure/Google Cloud Platform, Basics of AWS security services AWS WAF, AWS Shield, etc.		
6		**Cloud Applications and recent trends**	**06**	**CO6**
	6.1	Scientific Applications Healthcare: ECG analysis in cloud, Maintaining electronic health records. IoT-enabled Cloud Applications: Smart Agriculture Business and Consumer Applications: Social Networking and media applications.		

	6.2	Recent Trends: Mobile Cloud Computing: Definition, architecture, benefits and challenges of mobile cloud computing, Edge computing. Case Studies - Dew computing, Edge Computing, FOG Computing.		
		Self-learning: Quantum computing, Cloud native applications.		
		Total	**39**	

Textbooks:

1. R. Buyya, C. Vecchiola, S. T. Selvi, S. Poojara, and S. N. Srirama, Mastering Cloud Computing: Foundations and Applications Programming, 2nd ed. Amsterdam, The Netherlands: Morgan Kaufmann (Elsevier), 2026.
2. A. Srinivasan and J. Suresh, Cloud Computing: A Practical Approach for Learning and Implementation, 1st ed. New Delhi, India: Pearson Education, Jan. 2014.
3. R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2010.
4. B. Sosinsky, Cloud Computing Bible, 1st ed. Hoboken, NJ, USA: Wiley Publishing, Jan. 2011.

Reference books:

1. K. Jayaswal, J. Kallakurichi, D. J. Houde, and D. Shah, Cloud Computing Black Book, 2nd ed. New Delhi, India: Dreamtech Press, Jan. 2014.
2. M. Wittig and A. Wittig, Amazon Web Services in Action, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2018.
3. P. Arora, R. Biyani, and S. Dave, To the Cloud: Cloud Powering an Enterprise, 1st ed. New Delhi, India: Tata McGraw-Hill Education, Jan. 20, 2012.
4. K. Hwang, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, 1st ed. Burlington, MA, USA: Morgan Kaufmann Publishers, Nov. 22, 2011.

Online References:

1) NPTEL: https://onlinecourses.nptel.ac.in/noc22_cs20/preview
2) OpenStack Installation Guide: https://docs.openstack.org/install-guide/
3) Eucalyptus Installation: https://docs.eucalyptuscloud.org/eucalyptus/4.4.4/install-guide-4.4.4.pdf
4) AWS Management Console: https://aws.amazon.com/console/
5) https://ndl.iitkgp.ac.in NOC: Cloud Computing https://rb.gy/wyjtjx
6) https://ndl.iitkgp.ac.in NOC :Cloud Computing and Distributed Computing –Virtualization https://rb.gy/uuyzq3

Course Assessment:

ISE:

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study etc. of 15 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 50% syllabus)

End Semester Examination

$ ESE of duration 03 hours is of 80 marks and scaled to 60.

Question paper will comprise of 03 questions.
Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
Question3(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
All COs should be mapped as per the weightage in the syllabus.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practica	Tutoria	Theory	Practical	Tutoria	Total
MDMC 6063	Organizational Behavior and Human Resource Management	03	-	02	03	-	01	04

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessme		ESE$			
		ISE	MSE				
MDMC 6063	Organizational Behavior and Human Resource	20	20	60	25	--	125

Pre-requisite:

1. Basic understanding of communication and teamwork in project environments
2. Curiosity to learn about human behavior and roles in organizations

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO7: Global and Ethical Practices
4. PO8: Collaborative Teamwork
5. PO9: Communication
6. PO10: Project Management and Finance
7. PO11: Lifelong Learning

Course Objectives:

1. To help learners understand human behavior at work and factors influencing performance, motivation, and teamwork.
2. To provide foundational knowledge of HRM functions and leadership skills relevant to engineering careers.

Course Outcomes:

1. Explain fundamental concepts of Organizational Behaviour and HRM.
2. Analyse personality, perception, motivation, and attitudes affecting workplace behaviour.
3. Demonstrate effective interpersonal and leadership skills required for engineering teams.
4. Apply HRM concepts like recruitment, teamwork, performance, and conflict resolution in real situations.

Module No.	Unit No.	Topics	Hrs.	Mapped to Course Outcome
1.0	**1**	**Introduction to Organizational Behavior**	**3**	
		Meaning and significance of OB, functions, roles and skills related to management; contributing disciplines to the field of OB; challenges and opportunities for OB.		**CO1**
2.0	**2**	**Interpersonal Skills**	**3**	
		Importance of interpersonal skills for engineers; verbal and non-verbal communication; active listening; giving and receiving feedback; building trust and rapport; handling difficult conversations		**CO3**
3.0	**3**	**Personality**	**3**	
		Meaning and determinants of personality; major personality traits and attributes; Big Five Model; Job Fit Theory; measuring personality; managing emotions at the workplace.		**CO2**
4.0	**4**	**Perception and Decision Making**	**3**	
		Meaning of perception; factors influencing perception and attitude; Attribution Theory; errors in attribution; link between perception and individual decision making.		**CO2**
5.0	**5**	**Values, Attitudes, Job Satisfaction and Learning**	**3**	
		Definition and types of values; definition and types of attitudes; definition and measurement of job satisfaction; meaning of learning; methods of learning at work.		**CO2**
6.0	**6**	**Motivation**	**2**	
		Definition of motivation; theories of motivation (Theory X & Theory Y, Maslow's Need Hierarchy, Two-factor theory, ERG theory, Goal-setting theory, Reinforcement theory, Equity theory, Expectancy theory); motivation from theory to application; link between skill-based plans and motivation theories.		**CO2**
7.0	**7**	**Leadership Theories, Styles & Team Influence**	**3**	
		Concept of leadership; leader vs manager; leadership traits and competencies; overview of major leadership theories (trait, behavioural, contingency/situational); leadership styles and their relevance in technical teams.		**CO3**
8.0	**8**	**Applied Leadership Skills for Engineering Teams**	**3**	
		Leading teams; delegation and empowerment; decision-making and problem-solving as a leader; power and influence; ethical leadership; developing one's own leadership plan.		**CO3, CO4**

9.0	**9**	**Understanding Group Behaviour and Teams**	**3**	
		Definition of groups and teams; Stages of group and team development; Group structure: roles, norms, status; Group tasks and decision making; Types of teams - problem-solving, self-managed, cross-functional, virtual etc.; Features of effective teams; Creating and managing effective teams in engineering projects		**CO3**
10.0	**10**	**Conflicts in the Organisation**	**3**	
		Definition of conflict; transitions in conflict thought; functional and dysfunctional conflict; the conflict process; negotiation process; bargaining strategies (distributive & integrative).		**CO3**
11.0	**11**	**Organizational Culture and Stress**	**2**	
		Organizational Culture and Stress – Definition of organizational culture; types of culture; functions of culture; culture as an asset and liability; creating and sustaining culture; learning organizational culture; Hofstede's model of culture; Organizational Stress – definition of stress; understanding stress and its consequences; sources of stress; managing stress.		**CO4**
12.0	**12**	**Organizational Change**	**2**	
		Meaning of change; forces of change; levels of change; resistance to change; process of change; role of managers in		**CO4**
13.0	**13**	**Introduction to Human Resource Management & Talent**	**3**	
		Meaning, scope and objectives of HRM; role of HR in organizations and specifically in engineering / project-based firms; HR planning; job analysis and job descriptions; recruitment and		**CO1, CO4**
14.0	**14**	**Performance Management, Development & Employee**	**3**	
		Training and development; performance management and feedback; basics of compensation and rewards; employee engagement and welfare; HR's role in supporting career development of engineers.		**CO1, CO4**
15.0	**15**	**Tutorials**		**CO4**
		Case Study discussions, Role Play and quiz		
		Total	**39**	

Textbooks:

1. S. P. Robbins and T. A. Judge, Organizational Behavior, 19th ed. Harlow, U.K.: Pearson, 2022.
2. K. Aswathappa, Organizational Behaviour, 12th ed. New Delhi, India: McGraw-Hill Education, 2018.
3. G. Dessler, Human Resource Management, 16th ed. Harlow, U.K.: Pearson, 2020.
4. M. Armstrong, Armstrong's Handbook of Human Resource Management Practice, 16th ed. London, U.K.: Kogan Page, 2021.

Reference books:

1. S. P. Robbins and T. A. Judge, Organizational Behavior, 19th ed. Harlow, U.K.: Pearson, 2022.
2. K. Aswathappa, Organizational Behaviour, 12th ed. New Delhi, India: McGraw-Hill Education, 2018.
3. G. Dessler, Human Resource Management, 16th ed. Harlow, U.K.: Pearson, 2020.
4. M. Armstrong, Armstrong's Handbook of Human Resource Management Practice, 16th ed. London, U.K.: Kogan Page, 2021.

Course assesment

ISE

- To be conducted in any of these forms - Assignment/ Quiz/ Presentation/ Class Test/ Case study, etc., of 20 marks.
- ISE 20 marks = 05 marks for attendance + 15 marks for activities.

MSE:

- To be conducted as written examination for 20 marks (on 40% - 50% syllabus)

End Semester Examination:

$ ESE of duration 03 hours are of 80 marks and scaled to 60.

1. Question paper will comprise of 03 questions.
2. Question1(20 marks): - Solve any 04 out of 06. All questions carry 05 marks each.
3. Question 2 (40 marks): - Solve any 04 out of 06. All questions carry 10 marks each.
4. Question3(20 marks):- Solve any 04 out of 06. All questions carry 05 marks each.
5. All COs should be mapped as per the weightage in the syllabus.

CIAP (25-Marks):

25-Marks = Tutorial 20Marks + Attendance 5Marks
Tutorials will be class Wise and shall include real-world Organizational Behavior and Human Resource Management using excel. Hands-on Excel-based exercises to be conducted in the computer laboratory.

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL601	Intelligent Data Analytics Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSL601	Intelligent Data Analytics Lab	--	--	--	25	25	50

Pre- requisite:

1. Hadoop/Spark ecosystem basics
2. Programming in Java or Python
3. NoSQL databases
4. Big data analytics concepts
5. Skill Lab (Python Programming)

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO4: Conduct investigation of complex problem
4. PO3: Design/Development of Solutions
5. PO4: Conduct Investigations of Complex Problems
6. PO5: Engineering Tool Usage
7. PO7: Ethics
8. PO10: Project Management and Finance

Lab Objectives:

1. To understand and implement big data ecosystem tools such as Hadoop (HDFS), Spark, Hive, Pig, Sqoop, and Flume for distributed data storage, ingestion, and processing.
2. To design and analyze NoSQL and graph-based data models (HBase, MongoDB, Neo4j/GraphX) and evaluate their suitability for scalable and real-world applications.
3. To develop and optimize distributed data processing algorithms using MapReduce and Apache Spark, including batch and streaming data analytics techniques.
4. To perform data preprocessing, transformation, feature engineering, and visualization using Python and relevant libraries for effective data analysis.
5. To implement and evaluate supervised machine learning algorithms such as Decision Trees and

Support Vector Machines using appropriate performance metrics.

6. To apply advanced machine learning techniques including dimensionality reduction (PCA), ensemble methods (Random Forest, Boosting), neural models, and recommender systems for solving real-world problems.

Lab Outcomes:

1. Explain and implement the Hadoop/Spark ecosystem, perform distributed data storage (HDFS), and execute data ingestion using tools like Sqoop, Flume, Hive, and Pig.
2. Design, implement, and query NoSQL (wide-column and document) and graph data models, and analyze their suitability for large-scale applications.
3. Implement and optimize distributed and streaming data processing algorithms using MapReduce and Spark, including joins, matrix operations, PageRank, and streaming techniques.
4. Perform data preprocessing, feature engineering, dimensionality reduction, and visualization using Python and relevant libraries.
5. Implement, train, and evaluate supervised machine learning models, including classification algorithms, and analyze their performance using appropriate metrics.
6. Apply advanced machine learning techniques, including ensemble methods, neural models, and recommender systems, to solve real-world big data problem

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
Big Data Analytics: (Perform any 6 experiments out of 1 to 8)		
1	Explore and demonstrate HDFS basics and the Hadoop ecosystem components; perform common HDFS operations and explain roles of YARN, MapReduce, Hive, Pig, HBase, Sqoop, Flume, Kafka, Spark. Key Tasks: ▯ Install/configure single-node Hadoop (or use a VM/cluster image). ▯ Create HDFS directories, copy local → HDFS, list, display, move and delete files. ▯ Demonstrate HDFS replication and show how to read file blocks. ▯ Install/configure Sqoop & connect to a sample MySQL/Postgres DB. ▯ Use Sqoop import/export to move a table to Hive/HDFS and back. ▯ Configure Flume to capture log lines from a local file tail and write to HDFS. ▯ Produce a one-page diagram describing where Hive, Pig, HBase,	**LO1**

2	Install, model and query wide-column and document databases; show tradeoffs (consistency, schema). Key Tasks: ▯ Install HBase and create a table; insert/read rows using shell and Java/Python client. ▯ Install MongoDB; design a document schema for a simple dataset (e.g., user events) and perform CRUD and aggregation queries. ▯ Compare a use case (time-series or event logging) and justify chosen NoSQL store. Tools / Datasets: HBase, HBase shell, MongoDB, mongo shell, small	**LO2**
3	Implement two non-trivial MapReduce algorithms — sparse matrix multiplication and PageRank — to learn data partitioning and iterative MapReduce patterns. Key Tasks: ▯ Write MapReduce jobs (Java or Python streaming) for word count and numeric aggregates. ▯ Implement map-side and reduce-side join for two small tables (e.g., userID→name and transactions). ▯ Implement sparse matrix multiplication using MapReduce (matrix stored as coordinate list). ▯ Implement PageRank on a small web graph (iterative MR jobs with convergence criterion).	**LO3**
4	Learn Spark basics, perform ETL and analytics using RDD and DataFrame APIs; compare performance to MapReduce. Key Tasks:	**LO3,**
	▯ Set up PySpark session (local or cluster). ▯ Load a medium CSV, perform cleaning, transformations and aggregations with DataFrame API. ▯ Compare same job implemented in MapReduce (Lab 4) vs Spark — wall-time and code complexity. ▯ Demonstrate caching, partitioning, and simple tune-up (coalesce, persist). Tools / Datasets: Apache Spark (PySpark), Jupyter notebook, sample	
5	Use Hive and Pig to run SQL-style queries, table creation, partitions, and basic descriptive statistics/visualization. Key Tasks: ▯ Create Hive databases/tables; load and query data (joins, window functions, partitions). ▯ Implement same set of transformations in Pig Latin. ▯ Produce descriptive statistics (count, mean, median approx, group by) and export results for plotting. Tools / Datasets: Hive, Pig, HDFS, sample dataset (sales / logs).	**LO1**

6	Implement streaming algorithms (space-efficient) to process high-velocity data and measure accuracy. Key Tasks: ▯ Implement Bloom Filter for membership testing and evaluate false positive rate. ▯ Implement DGIM (counting 1's in sliding windows) to estimate recent counts and compare with exact counts. ▯ Integrate with a simple stream source (Flume/Kafka → consumer) to process live messages. Tools / Datasets: Python (or Java), Kafka/Flume, test stream	**LO3**
7	Run basic social graph analytics: degree distribution, connected components, centrality and community detection. Key Tasks: ▯ Load a small social graph into GraphX (Spark) or Neo4j. ▯ Compute degree, PageRank, connected components and top central nodes. ▯ Run a basic community detection (Label Propagation) and visualize subgraph. ▯ Discuss storage and modeling tradeoffs for graph analytics on massive graphs	**LO2**
8	Build and evaluate item recommendation systems (ALS collaborative filtering and simple content-based) on a dataset. Key Tasks: ▯ Preprocess ratings data, split into train/test. ▯ Train an ALS model (Spark MLlib), tune hyperparameters, compute RMSE and top-k recommendations. ▯ Implement a simple content-based recommender and compare coverage/accuracy. ▯ Discuss cold start and scalability strategies.	**LO4**
Machine Learning Lab: (Perform any 6 experiments out of 9 to 15)		
9	Implement the Decision Tree Classification algorithm in Python using a suitable dataset (like Play Tennis Dataset) by performing the following steps: a) Load and explore the dataset b) Split the dataset into training and testing sets c) Train the Decision Tree model on the training data d) Visualize the decision tree (if possible) e) Use the model to make predictions on the test data f) Evaluate the model using performance metrics such as accuracy, confusion matrix, precision, recall, and F1-score	**LO5**
10	Implement the Support Vector Machine (SVM) classification algorithm in Python using a suitable dataset (like Iris Dataset), and evaluate its performance using metrics such as accuracy, confusion matrix, precision, recall, and F1-score.	**LO5**
11	Implement Principal Component Analysis (PCA) in Python on a suitable dataset (like Iris dataset) to reduce its dimensionality by extracting the first n principal components (e.g., n = 1, 2, …, 10). Perform the following tasks: a) Apply PCA to compute principal components b) Select different values of n and observe the variance explained c) Visualize the data using the reduced components (e.g., 2D plot)	**LO4**

12	Implement ensemble learning using the bagging technique with the Random Forest algorithm in Python on a suitable dataset (Iris dataset) by performing the following tasks: a) Train the Random Forest model on the training data b) Use the model to make predictions on the test data c) Evaluate the model's performance using metrics such as accuracy, etc.	**LO6**
13	Implement ensemble learning using boosting techniques (AdaBoost or XGBoost) in Python for a classification or regression task using a suitable dataset [Breast Cancer Dataset (Best for Classification) and California Housing Dataset (Best for Regression)]. Perform the following tasks: a) Load and split the dataset into training and testing sets b) Train the boosting model (AdaBoost or XGBoost) on the training data c) Use the model to make predictions on the test data d) Evaluate the model's performance using appropriate metrics (e.g., accuracy for classification or mean squared error for regression)	**LO6**
14	Design and implement the McCulloch–Pitts neuron model in Python and demonstrate how it performs basic logical operations such as AND, OR, and NOT by analyzing its outputs for different input combinations.	**LO6**
15	Design and implement the Single Layer Perceptron learning algorithm to simulate basic logic gates such as AND, OR, and NOT, and analyze its output for different input combinations.	**LO6**

Textbooks:

7. T. White, Hadoop: The Definitive Guide, 4th ed. Sebastopol, CA: O'Reilly Media, 2015.
8. M. Guller, Big Data Analytics with Spark, New York, NY: Springer, 2015.
9. R. Kamal and P. Saxena, Big Data Analytics, New Delhi, India: McGraw Hill Education, 2018.
10. A. C. Müller and S. Guido, Introduction to Machine Learning with Python, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2016.
11. F. Chollet, Deep Learning with Python, 3rd ed. Shelter Island, NY, USA: Manning Publications, 2025.
12. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, 1st ed. Cambridge, MA, USA: MIT Press, 2016.

Reference books:

1. B. Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and Its Applications. Hoboken, NJ, USA: Wiley, 2014.
2. V. Prajapati, Big Data Analytics with R and Hadoop, 2nd ed. Birmingham, UK: Packt Publishing, 2016.
3. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Indianapolis, IN, USA: Wiley, 2015.
4. P. J. Sadalage and M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston, MA, USA: Addison-Wesley, 2012.
5. C. M. Bishop, Neural Networks for Pattern Recognition, 1st ed. Oxford, U.K.: Oxford University Press, 1995.

Online Resources:

1. Official Project Site of Apache Spark — documentation, tutorials, quick start guides, examples. Apache Spark+1
   Link: https://spark.apache.org/ Apache Spark
2. BigData Applications on HPC / Batch & Stream Analytics Tutorial — shows how to run analytic engines (Hadoop, Spark, Flink) on clusters: gives handson, practical perspective. UL HPC Tutorials
3. Big Data Analytics with Spark and Hadoop Download
4. NPTEL course: Introduction to Machine Learning (in Hindi) by Prof. Anubha Gupta, IIT Delhi, available at: NPTEL
5. Google's Machine Learning Crash Course: https://developers.google.com/machine- learning/crash-course
6. Fast.ai Practical Deep Learning: https://www.fast.ai/
7. DeepLearning.AI (Andrew Ng): https://www.deeplearning.ai/
8. Kaggle Learn: https://www.kaggle.com/learn
9. Machine Learning Mastery (Jason Brownlee): https://machinelearningmastery.com/
10. Towards Data Science (Medium Blog): https://towardsdatascience.com/
11. Scikit-learn (ML Library for Python): https://scikit-learn.org/
12. TensorFlow End-to-End ML Platform: https://www.tensorflow.org/

Software Tools

1. Apache Hadoop (HDFS + MapReduce + YARN) — for distributed storage and processing. Apache Hadoop+1
2. Apache Spark — for fast in-memory analytics, streaming, ML, and graph processing. Apache Spark+1
3. NoSQL databases: MongoDB, HBase / wide-column stores — for unstructured/semi- structured data labs. (These are covered in the textbook and ecosystem resources above.) McGraw Hill India+1
4. Programming & scripting languages: Java (for Hadoop MapReduce), Python (for Spark, streaming, ML), Scala optionally (for Spark).
5. Data ingestion & streaming tools (if needed later): e.g. connectors, shell scripts, or other tools — as part of bigger pipelines.

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment Journal) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical & oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL602	Software Engineering and Project Management Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSL602	Software Engineering and Project Management Lab	--	--	--	25	25	50

Pre- requisite:

1. Basic Programming Skills : C, Java

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO6: The Engineer and the World
6. PO7: Ethics
7. PO8: Individual and Collaborative Team work.
8. PO9: Communication
9. PO10: Project Management and Finance
10. PO11: Life-long Learning.

Lab Objectives:

1. To develop the ability to apply various software engineering process models, especially Agile, using industry-standard tools.
2. To enable students to prepare software documentation such as SRS, DFDs, UML diagrams, and project planning artifacts using professional software tools.
3. To train students in designing and executing software test cases using white-box and black-box testing techniques.
4. To provide hands-on experience in collaborative software development using Git and GitHub, and to introduce DevOps practices through Docker-based containerization for building, managing, and deploying applications.
5. To enhance project planning and management skills by creating Gantt charts, WBS, sprint plans, and risk registers.
6. To improve students' analytical, teamwork, collaboration, and project closure skills through mind mapping, stakeholder analysis, and lessons-learned exercises.

Lab Outcomes:

1. Apply Agile methodology and manage project tasks effectively using tools like Trello, Jira, and Zoho.
2. Prepare IEEE-standard SRS documents and create analytical diagrams such as DFDs, Use Case Diagrams, and Activity Diagrams.
3. Design and execute white-box and black-box test cases for validating software functionalities.
4. Demonstrate Git proficiency, collaborate via GitHub, and apply DevOps practices using Docker for building and deploying applications.
5. Create project management artifacts including mind maps, stakeholder maps, Gantt charts, WBS, and risk registers.
6. Plan, track, monitor, and close software projects by documenting sprint activities, risks, and lessons learned using modern project management tools.

Suggested List of Experiments: Students are required to complete at least 12 experiments.

Sr. No.	Title of Experiments	LO
1	Application of Agile Process model using Trello framework. To implement Agile practices such as backlog creation, sprint planning, task allocation, and progress tracking to demonstrate iterative development and team collaboration. Tools / Datasets: Trello (project management tool), sample project	**LO1**
2	Design SRS document in IEEE format for the given case study. To develop a Software Requirements Specification (SRS) document for a given case study using IEEE format. It should define functional and non-functional requirements, scope, and constraints for proper analysis. It provides a structured specification for system development. Tools / Datasets: IEEE SRS template, given case study.	**LO2**
3	Construct DFD for the case study using Dia. To develop Data Flow Diagrams (DFDs) for a given case study using Dia. It should represent data flow, processes, data stores, and external entities at context and detailed levels. This helps in understanding system functionality clearly. Tools / Datasets: Dia tool, given case study.	**LO2**
4	Construct a use case diagram for the case study using Dia. To develop a Use Case Diagram for a given case study using Dia to represent system functionalities and user interactions. The diagram should clearly identify actors, use cases, and their relationships to effectively model system behavior from the user's perspective. Tools / Datasets: Dia tool, given case study.	**LO2**
5	Construct UML diagram for the case study using Dia. To construct a UML Diagram for a given case study using Dia to illustrate key system components, relationships, and interactions. The diagram should clearly depict appropriate UML elements (such as classes, objects, or interactions) to effectively model and communicate the system design. Tools / Datasets: Dia tool, given case study.	**LO2**

6	Creation of test cases using white box testing and black box testing.	**LO3**

	To create test cases using white box and black box testing techniques. It covers inputs, outputs, and execution paths to validate system functionality and logic. This ensures effective software testing. Tools / Datasets: Test case templates, given problem statement/system.	
7	To execute essential Git commands for project version control and collaborate using GitHub repositories. To perform essential Git operations using GitHub for version control and collaboration, including repository creation, commits, branching, merging, and managing version history. Tools/Datasets: Git, GitHub, sample project repository.	**LO4**
8	To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers To gain an understanding of Docker architecture and container lifecycle through basic commands for creating, running, and managing containers and images.	**LO4**
9	To learn Dockerfile instructions, build an image for a sample web application using Dockerfile. To build a Docker image for a sample web app using a Dockerfile with key instructions like FROM, RUN, COPY, and CMD. Tools / Datasets: Docker, Dockerfile, sample web application source code.	**LO4**
10	Virtual Project Initiation and Stakeholder Mapping Using Lucidchart/ Draw.io. To create a stakeholder mapping diagram using Lucidchart or Draw.io. It maps stakeholders, their roles, and relationships in a project. This helps visualize project initiation clearly. Tools / Datasets: Lucidchart / Draw.io, given project scenario.	**LO5**
11	Gantt Chart Scheduling using GanttProject. To create a project schedule using GanttProject by defining tasks, durations, dependencies, and milestones. The Gantt chart should clearly represent the project timeline and task sequencing to support effective project planning and tracking.	**LO5**
12	Creating a Work Breakdown Structure (WBS) and Assigning Tasks Using Project Management Software (OpenProject). To create a Work Breakdown Structure (WBS) for a given project using OpenProject. It decomposes the project into smaller tasks, defines hierarchies, and assigns responsibilities. This ensures structured planning and execution. Tools/ Datasets: OpenProject software, given project scenario.	**LO5**
13	Creating a Risk Register and Performing Group Risk Assessment Using Risk Register Software (RiskyProject Lite).To create a risk register and perform risk assessment using RiskyProject Lite. It identifies risks, analyzes their probability and impact, and prioritizes them. This helps define mitigation strategies and control measures. Tools / Datasets: RiskyProject Lite software, given project scenario.	**LO6**

14	Agile Sprint Planning and Professional Project Scheduling in Software Tools use Jira or Zoho Projects (for Agile sprint planning and tracking tasks/milestones). To plan and manage Agile sprints using Jira or Zoho Projects by defining user stories, creating sprint backlogs, assigning tasks, and tracking progress. The activity should include scheduling milestones, monitoring task completion, and managing iterative development cycles effectively. Tools / Datasets: Jira / Zoho Projects, given project scenario and user stories.	**LO6**
15	Documenting Project Closure and Interactive Lessons Learned Review {Jira or Trello (for Agile project boards and workflow visualization)} or Kahoot or Menti meter (for interactive lessons learned and feedback sessions). To document the project closure by summarizing project deliverables, evaluating outcomes, and conducting feedback sessions to identify improvements for future projects. Tools / Datasets: Jira / Trello, Kahoot / Mentimeter, given project scenario.	**LO6**
16	Case study on any emerging tools on DevOps. Study a selected emerging DevOps tool and examine its features, workflow integration, and impact on software development and deployment processes. The analysis should highlight how the tool supports continuous integration, continuous delivery, automation, and collaboration in modern DevOps practices.	**LO4**

Textbooks:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach", 9th edition, McGraw-Hill Publications, 2019.
2. I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2020.
3. P. Jalote, An Integrated Approach to Software Engineering, 3rd ed. New Delhi, India: Springer, 2005.
4. C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2020.

Reference books:

1. M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. Boston, MA, USA: Addison-Wesley, 2022.
2. G. J. Myers, C. Sandler, and T. Badgett, The Art of Software Testing, 3rd ed. Hoboken, NJ, USA: Wiley, 2020.

Online Resources:

1. Trello Guides: Help Getting Started With Trello | Trello
2. Free Online Design Tools & Diagram Makers
3. Software Testing – Beginner's Guide

Software Tool

1. Trello (open source): https://trello.com/
2. Dia http://dia-installer.de/
3. GitHub for Open-Source Projects: https://github.comhttps://github.com

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) = 20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSL603	Skill Lab (Data Handling & Visualization) Lab	--	2*+2	--	--	02	-	02

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSL603	Skill Lab(Data Handling & Visualization) Lab	--	--	--	25	25	50

Pre- requisite:

1. Basic statistics and Maths
2. Python programming

Program Outcomes addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions:
4. PO4: Conduct Investigations of Complex Problems:
5. PO5: Engineering Tool Usage
6. PO6: The Engineer and the World
7. PO8: Ethics
8. PO11: Life-Long Learning

Lab Objectives:

1. To Introduce the concept of Data Analytics Lifecycle.
2. To Develop Mathematical concepts required for advance regression.
3. To Understand data modeling in time series and its process.
4. To create awareness about Text analytics and its applications.
5. To provide overview of Data analytics and visualization with R/ Python.
6. To understand data visualization using power BI and tableau.

Lab Outcomes:

1. Comprehend basics of data analytics and visualization.
2. Apply various regression models on a given data set and perform prediction.
3. Demonstrate advanced understanding of Time series concepts and analysis of data using various time series models.
4. Design Text Analytics Application on a given data set
5. Experiment with different analytics techniques and visualization using R / Python
6. Develop interactive Power BI dashboards integrating data transformation and visualization for professional reporting.

Module No.	Unit	Topics	Hrs.	LO
1.0		**Introduction to Data analytics and Life cycle**	**03**	
	1.1	Data Analytics Lifecycle Overview: Key roles, project background, business understanding and identification of data		**LO1**
	1.2	Data Preparation & Modeling: Data conditioning, visualization		
	1.3	Results & Deployment: Communicating insights effectively and operationalizing the analytics solution for real-world implementation.		
		Self-learning Topics: Case studies of end-to-end analytics projects		
2.0		**Regression Models**	**05**	
	2.1	Multiple Linear Regression: Assessing model fit and assumptions, Cross-validation and model selection, Stepwise regression methods, Prediction using regression models.		**LO2**
	2.2	Logistic Regression and Comparison: Logistic Response function and logit, Logistic regression and Generalized Linear model, ROC, AUC, and confusion matrix for model evaluation		
		Self-learning Topics: Introduction to GLM families beyond logistic (Poisson, Gamma)		
3.0		**Time Series**	**05**	
	3.1	Overview of Time Series Analysis Box-Jenkins Methodology, ARIMA Model Autocorrelation Function (ACF), Autoregressive Models, Moving Average Models, ARMA and ARIMA Models, Building and Evaluating an ARIMA Model,		**LO3**
		Self-learning Topics: Understanding time series decomposition, GARCH models for volatility forecasting (finance)		
4.0		**Text Analytics**	**04**	

	4.1	History of text mining, Roots of text mining overview of seven practices of text analytic, Application and use cases for Text mining		**LO4**
	4.2	Text Analysis Steps, A Text Analysis Example, Collecting Raw Text, Representing Text, Term Frequency—Inverse Document Frequency (TFIDF)		
		Self-learning Topics: Case studies on healthcare (clinical notes), Ecommerce (reviews, rating analysis)		
5.0		**Data analytics and visualization with R / Python**	**03**	
	5.1	Introduction to R: Visualization and Analysis using R.		**LO5**
	5.2	Essential Data Libraries for data analytics and Visualization in Python		
		Self-learning Topics: Case studies on simple analytical reports using Python or R		
6.0		**Power BI / Tableau**	**06**	
	6.1	Principles of data visualization, Power BI / Tableau interface & dashboards, Parameter controls in Tableau, LOD (Level of Detail) expressions in Tableau, DAX basics, calculated measures,		LO6
		Storytelling with data, Interactive reporting.		
		Self-learning Topics: Introduction to connecting different data sources in dashboards		
		Total	**26**	

Textbooks:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, EMC Education services Wiley Publication.
2. Data Analytics using Python: Bharati Motwani, Wiley Publications.
3. Data Analytics using R, Bharati Motwani, Wiley Publication.

Reference books:

1. Data Mining, Concepts and Techniques: 3rd edition, Jiawei Han, Micheline Kamber and Jian Pei
2. Python for Data Analysis: 3rd Edition, Wes McKinney, Publisher(s): O'Reilly Media, Inc.

Online References:

1. Introduction to Data Analytics, IIT Madras,Prof. Nandan Sudarsanam, Prof. Balaraman Ravindran -https://nptel.ac.in/courses/110106072
2. Explorefundamentalsof datavisualizationwith Power BI: https://microsoftlearning.github.io/DP-900T00A-Azure-Data- Fundamentals/Instructions/Labs/dp900 -pbi-06-

lab.html

3. Data visualization with python- https://www.coursera.org/learn/python-for-data-visualization
4. Microsoft Power BI Learning Path -https://learn.microsoft.com/power-bi/
5. Tableau Learning Resources-https://www.tableau.com/learn
6. Google Data Studio (Looker Studio) Tutorials-https://support.google.com/looker-studio

Software Tools:

1. R Studio
2. Python
3. Power BI

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	LO
1	To study and implement basic operations using popular data analytics and machine learning libraries, including Scikit-Learn, TensorFlow, Keras, NumPy, Matplotlib, and Seaborn.	**LO1**
2	To build and evaluate a linear/multiple regression model using Python on a given dataset by performing the following steps: a. Split the dataset into training and testing sets. b. Train the regression model and determine model parameters (intercept and coefficients/slope). c. Visualize the training and testing datasets along with the regression line. d. Predict the output for the test dataset. e. Compare the actual and predicted values to assess model performance.	**LO2**
3	To develop and implement a Logistic Regression model using Python on a suitable dataset for binary classification. The experiment includes training the model to predict class labels and evaluating its performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.	**LO2**
4	To develop and implement an ARIMA (Autoregressive Integrated Moving Average) model using Python or R to analyze and forecast time series data.	**LO3**
5	To perform time series analysis using Python or R to explore, model, and interpret temporal data.	**LO3**
6	Text analytics: To develop and implement a Text Analytics system using Python or R to perform tasks such as spam detection or sentiment analysis on textual data.	**LO4**
7	To design and implement two data visualization experiments in R using different libraries to effectively represent and analyze a given dataset	**LO5**
8	To design and implement two data visualization experiments in python using different libraries to effectively represent and analyze a given dataset	**LO5**
9	Geospatial Data Analytics: Visualizing Vector Data, Raster Data using open-source tools (e.g.: QGIS)	**LO5**
10	Geospatial Data Analytics: To perform supervised classification on Sentinel images to identify Land Use / Land Cover (Google Earth Engine)	**LO5**

11	A)To Implement simple and effective visualizations in power BI to create column chart, Bar chart, Pie chart, Table and card visual B)Use power BI to connect excel/csv file, create visualization and apply filters and slicers	**LO6**
12	To explore and implement advanced visualization techniques in Power BI by creating chart types (waterfall charts, tree maps, funnel charts, and maps).	**LO6**
13	To build one complete interactive dashboard (Power BI) Perform tasks: i)import business dataset ii)clean and transform data iii)create multiple visuals iv) Add slicers & KPI	**LO6**
14	To Implement data visualization techniques using Tableau by connecting external data sources and creating interactive visual representations.	**LO6**
15	Mini Project- Analyze application data and create interactive dashboards using charts to identify trends, patterns, and business insights.	**LO6**

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and a minimum passing marks in term work. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Practical Exam: (2 hours/ 25 Marks)

- End-semester Practical /Oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

Course Code	Course Name	Teaching Scheme (Hrs.)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEL 6011	Robotics and Applications Lab	--	2	--	--	1	--	1

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE$			
		ISE	MSE				
AIDSPEL 6011	Robotics and Applications Lab	--	--	--	25	--	25

Pre- requisite:

1. Basic programming in Python/C/C++.
2. Fundamentals of mathematics
3. Kinematics, and electronics.

Program Outcomes Addressed:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO5: Engineering tool Usage
5. PO6: The Engineer and the World
6. PO8: Ethics
7. PO11: Life-long Learning.

Lab Objectives:

1. To understand fundamentals of robotics including kinematics, dynamics, and control.
2. To gain hands-on experience with sensors, actuators, and robotic mechanisms.
3. To perform robot simulation using ROS, Gazebo, Webots or MATLAB.
4. To implement motion planning, path planning, and mobile robot navigation.
5. To develop robot programming skills using ROS, Python and microcontrollers.
6. To integrate hardware and software systems in a robotic platform.

Lab Outcomes: Learners will be able to:

1. Apply fundamental and advanced concepts of robot kinematics and dynamics to analyze and solve robotics problems.
2. Design and implement embedded control systems by interfacing sensors and actuators with microcontrollers.
3. Develop and test perception algorithms using vision and sensor data for environment interaction and obstacle avoidance.
4. Formulate and implement path planning and control algorithms for mobile and manipulator robots.
5. Integrate AI techniques and ROS-based software tools for developing and deploying

advanced robotic applications.

6. Develop interdisciplinary problem-solving skills by integrating mechanical design, electronics, programming, and data analysis in robotics projects.

Suggested List of Experiments: Students are required to complete at least 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
1	Installation & setup of ROS, Gazebo, CoppeliaSim, and Python libraries. To install and configure robotics software environment with proper dependencies to enable simulation and programming. Tools/Datasets: ROS, Gazebo, CoppeliaSim, Python libraries.	**LO1**
2	Implementation of rotation & translation matrices in Python/MATLAB. To apply transformation matrices to perform 2D/3D rotation and translation operations in robotic systems. ToolsDatasets: Python / MATLAB, NumPy (if Python is used).	**LO1**
3	Design and implement the forward kinematics of a 2-DOF/3-DOF robotic manipulator on a physical hardware setup and validate the results by observing and measuring the actual end-effector position through real-time visualization tools. To compute end-effector position using forward kinematics and verify it using a physical setup with real-time visualization tools. Tools/Datasets: Robotic manipulator setup (2-DOF/3-DOF), ROS / Gazebo / CoppeliaSim, Python / MATLAB, real-time visualization tools.	**LO2**
4	Design and implement the Inverse kinematics of a 2-DOF/3-DOF robotic manipulator on a physical hardware setup and validate the results by observing and measuring the actual end-effector position through real-time visualization tools. To compute joint angles from a desired end-effector position and validate results using hardware and real-time visualization tools. Tools/Datasets: Robotic manipulator setup (2-DOF/3-DOF), ROS / Gazebo / CoppeliaSim, Python / MATLAB, real-time visualization tools.	**LO2**

5	Guided Dynamic Modeling and Torque Calculation for a 2-DOF Planar Arm. To perform dynamic modeling of a 2-DOF planar robotic arm and compute joint torques for given motion conditions. The task focuses on deriving basic equations of motion and analyzing the relationship between applied torque and robot movement. Tools/Datasets: Python / MATLAB, ROS / Gazebo / CoppeliaSim.	**LO3**
6	Design of a Simple Servo-Based Robotic Arm for Pick-and-Place. To design a simple robotic arm using servo motors and implement pick-and-place operations. The experiment should demonstrate basic robotic movement control, object handling, and coordinated joint motion for performing simple tasks. Tools/Datasets: Servo motors, microcontroller (Arduino/ESP32), robotic arm kit, programming software.	**LO3**
7	Basic Mobile Robot Programming using Arduino/ESP32. To program a mobile robot using Arduino or ESP32 to perform basic movements such as forward, backward, left, and right navigation. The experiment focuses on motor control, sensor interfacing, and basic embedded programming for robotic motion. Tools/Datasets: Arduino / ESP32, motor driver module, mobile robot kit, sensors.	**LO4**
8	Simple Robot Vision using OpenCV: Color-Based Object Detection. To develop a simple robot vision system using OpenCV to detect and track objects based on color segmentation. The experiment focuses on image processing techniques such as filtering, color space conversion, and object localization for robotic perception. Tools/ Datasets: Python, OpenCV, webcam or camera module.	**LO4**
9	Obstacle Avoidance Robot using Ultrasonic/IR Sensors. To design and implement a mobile robot that detects and avoids obstacles using ultrasonic or infrared sensors. The system should process sensor inputs to control motor actions for safe navigation in the environment. Tools/Datasets: Arduino / ESP32, ultrasonic/IR sensors, motor driver, mobile robot kit.	**LO3**
10	Implementation of A* Path Planning in 2D Grid World. To develop and execute the A* search algorithm to find the optimal path between a start and goal position in a 2D grid world. The experiment focuses on heuristic-based path optimization and obstacle-aware navigation. Tools/Datasets: Python, NumPy, grid map dataset / simulated environment.	**LO4**

11	Design of line Following Robot using IR Sensors with Basic P and PD Control. To develop a line-following robot that uses IR sensors to detect path lines and applies P and PD control algorithms for smooth and stable navigation. The experiment focuses on real-time error correction and motor speed adjustment based on sensor feedback. Tools/ Datasets: Arduino / ESP32, IR sensors, motor driver, robotic kit.	**LO5**
12	Basic ROS Programming with Preconfigured TurtleBot World. To develop and execute basic ROS nodes to control and simulate robot movement in a preconfigured TurtleBot world. The experiment focuses on ROS topics, publishers, subscribers, and simple navigation commands in a simulated environment. Tools /Datasets: ROS, TurtleBot simulation, Gazebo.	**LO5**
13	Mobile Robot Control Using Smartphone Bluetooth/Wi-Fi Application. To design and implement a mobile robot that can be controlled using a smartphone via Bluetooth or Wi-Fi. The experiment focuses on wireless communication, motor control, and real-time command execution for robot navigation. Tools /Datasets:Arduino /ESP32,Bluetooth/Wi-Fi module, mobile app (Blynk/Arduino IoT app), robot kit.	**LO5**
14	Deep Learning-Based Real-Time Object Recognition for Robotic Applications. To develop a real-time object recognition system using deep learning models to detect and classify objects for robotic vision. Tools/Datasets: Python, TensorFlow / PyTorch, OpenCV, camera module, COCO/ ImageNet.	**LO6**
15	Autonomous Robot Navigation using SLAM and ROS. To develop an autonomous navigation system using SLAM in ROS to map an unknown environment and enable robot localization and path planning. Tools/Datasets: ROS, SLAM packages, Gazebo / TurtleBot simulation, LiDAR or sensor data.	**LO6**
16	Mini-project on any robotic application. To design and implement a small-scale robotic system for a chosen application such as automation, navigation, or object handling. The project should demonstrate integration of sensors, actuators, and control logic to achieve a functional robotic solution.	**LO6**

Textbooks:

1. H. D. Taghirad, Fundamentals of Robotics: Applied Case Studies with MATLAB® and Python, Boca Raton, FL, USA: CRC Press, 2025.
2. Python Programming Handbook for Robotics Development.

Reference books:

1. Modern Robotics: Mechanics, Planning, and Control by Kevin Lynch and Frank Park (2018).
2. Robot Operating System (ROS) for Absolute Beginners: Robotics Programming Made Easy by Lentin Joseph

Online Resources:

1. https://www.mathworks.com/products/robotics.
2. ROS Tutorials and Documentation: https://docs.ros.org
3. OpenCV Documentation: https://opencv.org
4. https://github.com/aras-labs/Fundamentals_of_Robotics.
5. https://www.mathworks.com/academia/books/matlab-with-python-debray.html.

Software Tools:

1. MATLAB with Robotics System Toolbox and Simulink (including ROS Toolbox)
2. Python with robotics libraries (Robotics Toolbox for Python, OpenCV)
3. Robot Operating System (ROS / ROS 2)
4. Gazebo or CoppeliaSim for robot simulation
5. Arduino IDE and ESP32 development environment

Hardware Tools:

1. Microcontroller boards: Arduino Uno, ESP32, Raspberry Pi
2. Servo motors (for robotic arm actuation)
3. Ultrasonic and IR sensors (for obstacle detection and line following)
4. Camera or webcam (for vision-based tasks)
5. Mobile robot chassis/platform or TurtleBot (for ROS experiments)
6. Communication modules: Bluetooth/Wi-Fi modules

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theor	Practic	Tutoria	Theor	Practica	Tutori	Total
AIDSPEL 6012	Blockchain Technology Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessme		ESE$			
		ISE	MSE				
AIDSPEL 6012	Blockchain Technology Lab	-	-	-	25	-	25

Pre- requisite:

1. Cryptography and Cryptocurrency

Program Outcomes:

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. PO4: Conduct Investigations of Complex Problems:
5. PO5: Engineering Tool Usage
6. PO7: Ethics
7. PO8: Individual and Collaborative
8. PO9: Communication:
9. PO10: Project Management and Finance:
10. PO11: Life-Long Learning
11. PSO1: Problem solving

Lab Objectives:

1. To develop and deploy smart contracts on local Blockchain.
2. To deploy the smart contract on test networks.
3. To deploy and publish smart contracts on Ethereum test network.
4. To design and develop crypto currency.
5. To understand permissioned Blockchain.
6. To design and develop a Full-fledged DApp using Ethereum/Hyperledger.

Lab Outcomes: Learners will be able to:

1. Develop and test smart contract on local Blockchain.
2. Develop and test smart contract on Ethereum test networks.
3. Write and deploy smart contract using Remix IDE and Metamask.
4. Design and develop Cryptocurrency.
5. To develop application using solidity and Ganache.
6. Develop and test DApp using Ethereum/Hyperledger.

Suggested List of Experiments: Students are required to complete at least 12 experiments.

Sr. No.	Title of Experiments	LO
1	Study Truffle framework and implement a local blockchain, analyzing cryptography and Merkle Tree hashing with a real-time application in secure transaction verification used in digital payment systems and cryptocurrencies	**LO1**
2	Configure and create a Genesis Block to initialize a private blockchain network and study its role through a real-time case study of Ethereum blockchain network initialization	**LO1**
3	Develop a Solidity Smart Contract for a calculator application to understand contract execution logic with real-time applications in decentralized computation and automated financial calculations	**LO1**
4	Develop a Smart Contract using Solidity for employee database management, demonstrating real-time blockchain-based HR record systems used for tamper-proof employee records	**LO2**
5	Implement blockchain wallet integration and transaction handling using Solidity, demonstrating real-time digital wallet systems used for secure cryptocurrency transfers	**LO2**
6	Deploy and test Ethereum smart contracts using the Ganache blockchain platform, demonstrating real-time blockchain testing environments used by developers before production deployment	**LO2**
7	Develop and deploy smart contracts using MetaMask and Remix IDE, demonstrating real- time decentralized application interaction with blockchain networks and browser wallets	**LO2**
8	Design and develop a simple cryptocurrency token, demonstrating real-time blockchain token systems used in crowdfunding, digital assets, and decentralized finance (DeFi)	**LO3**
9	Create and configure a network in Hyperledger Fabric, demonstrating real-time enterprise blockchain applications in supply chain management and logistics tracking	**LO4**

10	Case Study on Hyperledger Fabric architecture, analyzing real-time enterprise use cases such as product traceability in supply chains	**LO5**
11	Case Study on other blockchain platforms (Ethereum, Corda, EOS) analyzing real-time applications in finance, healthcare data sharing, and government systems	**LO6**
12	Develop a blockchain-based application for secure data storage and verification, demonstrating real-time applications in academic certificate verification systems	**LO6**
13	Design and develop a decentralized application (DApp) using Ethereum or Hyperledger Fabric, demonstrating real-time decentralized marketplace or digital asset management systems	**LO6**
14	Develop an e-Voting application using Ganache blockchain, demonstrating real-time secure digital voting systems ensuring transparency and tamper-proof results	**LO6**
15	Case Study on cryptocurrency development, analyzing real-time cryptocurrency ecosystems including token generation, mining, and blockchain transactions	**LO3**

Textbooks:

1. Ethereum Smart Contract Development, Mayukh Mukhopadhyay, Packt publication.
2. Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, Ritesh Modi, Packt publication.
3. Hands-on Smart Contract Development with Hyperledger Fabric Y2, Matt Zand, Xun Wu and Mark Anthony Morris, O'Reilly.

Referencebooks:

1. Mastering Blockchain, Imran Bashir, Packt Publishing
2. Introducing Ethereum and Solidity, Chris Dannen, APress
3. Hands-on Blockchain with Hyperledger, Nitin Gaur, Packt Publishing

Online Resources:

1. https://trufflesuite.com/
1. https://metamask.io/
2. https://remix.ethereum.org/
3. https://www.hyperledger.org/use/f abric

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
AIDSPEL 6013	Image and Video Processing Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course		ESE			
		ISE	MSE				
AIDSPEL 6013	Image and Video Processing Lab	--	--	--	25	-	25

Pre-requisite:
1. ECC301-Applied Mathematics III
2. ECC401-Applied Mathematics IV

Program Outcomes:
1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design of Solutions
4. PO5: Modern tool usage
5. PO9: Individual and team work.
6. PSO1: To analyze and apply logical, mathematical, and computational skills to solve problems in Artificial Intelligence and Data Science
7. PSO2: To implement Artificial Intelligence and Data Science techniques such as data analysis, machine learning, data mining, neural networks, and algorithmic design using emerging tools and technologies.

Lab Objectives:
1. To understand the fundamental concepts of digital images and videos, including sampling, quantization, colour models, video standards, and various image/video file formats.
2. To implement and analyze image enhancement techniques in both spatial and frequency domains for improving image quality.
3. To apply image segmentation and morphological operations for extracting meaningful information from images and to understand real-time applications.
4. To explore image transforms, image compression techniques, and basic motion estimation methods used in video processing, focusing on their role in efficient representation, transmission, and analysis of visual data.

Lab Outcomes:

1. Explain digital image/video basics including sampling, quantization, colour models, and file formats.
2. Apply spatial and frequency-domain enhancement techniques to improve image quality.
3. Analyze the impact of various filters and camera techniques on image sharpness and noise.
4. Perform image segmentation and morphological operations for feature extraction.
5. Implement and compare image transforms such as DFT, DCT, DHT, and FFT.
6. Apply lossless and lossy compression and basic motion estimation techniques used in video processing.

Suggested List of Experiments: Students are required to complete at least 12 experiments.

Sr. No.	Title of Experiments	LO
1	Perform sampling and quantization on an image and convert between RGB, HSI, and YUV color models. Task :Students are required to perform sampling and quantization on a given digital image by reducing its spatial resolution and varying the number of intensity levels, and analyze the impact on image quality. Further, they must convert the image between RGB, HSI, and YUV color models, extract individual components, reconstruct the image, and compare the results to understand differences between color representations.	**LO1**
2	Analyze header structure, storage requirements, and compression properties of BMP, TIFF, and JPEG formats. Task :Students are required to study and compare the image formats BMP, TIFF, and JPEG by analyzing their header structure, storage requirements, and compression techniques. They should examine sample images of each format, estimate file sizes, identify the type of compression used (lossy/lossless), and present a comparative analysis highlighting advantages and limitations of each format.	**LO1**
3	Implement digital negative, contrast stretching, intensity slicing, bit-plane slicing, log and power-law transforms. Task :Students are required to implement gray-level transformation techniques including digital negative, contrast stretching, intensity level slicing, bit-plane slicing, logarithmic, and power-law transformations on a given image. They should display and compare the results of each method and analyze their effects on image enhancement and feature visibility	**LO2**
4	Apply smoothing and sharpening filters on an image and study their effects on noise reduction and image sharpness. Task :Students are required to apply smoothing and sharpening filters on a given image using appropriate techniques (e.g., averaging, Gaussian, Laplacian). They should compare the outputs and analyze the effects on noise reduction and image sharpness with suitable observations.	**LO3**
5	Apply sharpening filters to enhance image details and edges and analyze their effect on noise amplification. Task :Students are required to apply different sharpening filters (such as Laplacian, unsharp masking, and high-pass filters) on a given image to enhance edges and fine details. They should compare the results and analyze how sharpening improves detail while	**LO3**

	potentially amplifying noise, with appropriate observations.	
6	Implement DFT-based filtering using Ideal, Gaussian, and Butterworth filters for smoothing and sharpening. Task :Students are required to implement DFT-based filtering on a given image by applying Ideal, Gaussian, and Butterworth filters in the frequency domain for both smoothing and sharpening. They should compare the results, observe effects such as blurring and ringing, and analyze the performance of each filter.	**LO5**
7	Perform histogram equalization and histogram specification and analyze enhancement results. Task :Students are required to perform histogram equalization and histogram specification on a given image to enhance its contrast. They should compare the original and processed images, plot the corresponding histograms, and analyze the improvement in image quality and intensity distribution.	**LO2**
8	Detect edges using Roberts, Prewitt, Sobel, and Laplacian operators and compare performance. Task: Students are required to detect edges in a given image using Roberts, Prewitt, Sobel, and Laplacian operators. They should compare the outputs and analyze the performance of each operator in terms of edge detection accuracy, noise sensitivity, and computational complexity.	**LO4**
9	Implement region growing, region splitting, and merging for segmentation. Task: Students are required to implement image segmentation techniques including region growing, region splitting, and region merging on a given image. They should compare the segmented outputs and analyze the effectiveness of each method in terms of accuracy, continuity of regions, and sensitivity to noise and seed selection.	**LO4**
10	Perform erosion, dilation, opening, and closing operations and study their effect on object shape, size, and noise removal. Task: Students are required to perform morphological operations such as erosion, dilation, opening, and closing on a given binary or grayscale image. They should compare the results and analyze the effect of each operation on object shape, size, and noise removal.	**LO4**
11	Perform the Hit-or-Miss operation for detecting specific features and patterns in a binary image. Task :Students are required to implement the Hit-or-Miss transformation on a given binary image to detect specific patterns or features. They should design appropriate structuring elements, apply the operation, and analyze the effectiveness of detection with suitable observations.	**LO4**
12	Implement DFT, FFT, DCT, DHT, and FHT and compare energy compaction and complexity. Task :Students are required to implement DFT, FFT, DCT, DHT, and FHT on a given signal or image and analyze their performance. They should compare the transforms in terms of energy compaction, computational complexity, and suitability for image processing applications, with appropriate observations.	**LO5**
13	Implement Run Length Coding, Huffman Coding, and Arithmetic Coding for grayscale images. Task :Students are required to implement Run Length Coding, Huffman Coding, and Arithmetic Coding for a given grayscale image. They should compare the compression efficiency, compression ratio, and reconstruction quality, and analyze the performance of	**LO6**

	each technique with suitable observations	
14	Implement scalar/vector quantization and DCT-based JPEG-like compression. Task : Students are required to implement scalar and vector quantization techniques along with DCT-based JPEG-like compression on a given image. They should compare the compression performance, reconstruction quality, and analyze the trade-offs between compression ratio and image distortion with suitable observations.	**LO5**
15	Implement optical flow, block matching (pixel-based), and region-based motion estimation. Task :Students are required to implement motion estimation techniques including optical flow, pixel-based block matching, and region-based methods on a given video sequence or image frames. They should compare the results and analyze the performance of each technique in terms of motion accuracy, computational complexity, and robustness to noise and illumination changes.	**LO6**

Textbooks:

1. Gonzalez, Rafael C. & Woods, Richard E. "Digital Image Processing ". Pearson, 4th Edition (2018).
2. Wang, Yao; Ostermann, Jörn; Zhang, Ya-Qin, "Video Processing & Communications", Prentice Hall, 1st Edition (2002).
3. Jayaraman, S.; Esakkirajan, S.; Veerakumar, T. "Digital Image Processing", Tata McGraw-Hill Education, 3rd Edition, 2009. Central University of Punjab Catalog
4. Jain, Anil K., "Fundamentals of Digital Image Processing", Prentice Hall (India), 1st Edition 1989
5. Sridhar, S. "Digital Image Processing", Oxford University Press, 2nd Edition 2012

Reference books:

1. Tekalp, A. Murat, "Digital Video Processing", Prentice Hall, 2nd Edition (2015). PTG Media
2. Forsyth, David A.; Ponce, Jean, "Computer Vision: A Modern Approach", Pearson, 2nd Edition (2011).
3. Pakhira, Malay K. "Digital Image Processing and Pattern Recognition", Prentice Hall of India, 1st Edition 2011
4. Chanda, Bhabatosh; Majumder, Dwijesh Dutta, "Digital Image Processing & Analysis", PHI Learning / Prentice Hall of India, 2nd Edition 2011.
5. Sayood, Khalid, "Introduction to Data Compression", Morgan Kaufmann, 5th Edition (2017).

Online References:

1. https://onlinecourses.nptel.ac.in/noc22_ee116/preview
2. https://www.coursera.org/learn/introduction-image-processing
3. https://onlinecourses.nptel.ac.in/noc21_ee20/preview

Term Work:

- Term work should consist of at least 12 experiments. (10 hardware + 02 simulation experiments)

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and a minimum passing mark in term work. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiments) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practical			Theory	Practic	
AIDSPEL 6014	Augmented Reality and Virtual Reality Lab	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course		ESE			
		ISE	MSE				
AIDSPEL 6014	Augmented Reality and Virtual Reality Lab	--	--	--	25	-	25

Pre- requisite:

1. Computer Graphics

Program Outcomes Addressed

1. PO1: Engineering Knowledge
2. PO2: Problem Analysis
3. PO3: Design/Development of Solutions
4. **PO5: Engineering Tool Usage**

Lab Objectives:

1. To perform installation of Unity.
2. To explore working of VR Gadget.
3. To develop scene VR application.
4. To track objects in a virtual environment.

Lab Outcomes: Learners will be able to:

1. Setup VR development environment.
2. Use HTC Vive/ Google Cardboard/ Google Daydream and Samsung gear VR.
3. Develop VR scene and place object.
4. Work with Augmented Faces features.

Suggested List of Experiments: Students are required to complete the 12 experiments.		
Sr. No.	**Title of Experiments**	**LO**
1	Installation of Unity and Visual Studio, setting up Unity for VR development, understanding documentation of the same.	**LO1**
2	Build a VR Experience Using the Unity XR Interaction Toolkit (XRI) / Demonstration of the working of HTC Vive, Google Cardboard, Google Daydream and Samsung gear VR.	**LO2**
3	Develop a scene in Unity that includes: i. a cube, plane and sphere, apply transformations on the 3 game objects. ii. add a video and audio source	**LO2**
4	Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the colour, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the colour and material/texture of the game objects dynamically on the button click.	**LO3**
5	Develop a scene in Unity that includes a sphere and plane. Apply Rigid body component, material and Box collider to the game Objects. Write a C# program to grab and throw the sphere using vr controller.	**LO4**
6	Develop a simple UI (User interface) menu with images, canvas, sprites and buttons. Write a C# program to interact with UI menu through the VR trigger button such that on each successful trigger interaction display a score on scene.	**LO4**
7	Place a three-dimensional ARCore pawn on detected AR plane surfaces	**LO4**
8	Implement Augmented Faces feature to overlay 3D masks or objects on detected human faces.	**LO4**
9	Implement teleportation locomotion in VR using XR Interaction Toolkit. Users move by pointing to a location and teleporting there.	**LO2**
10	Create interactive objects that can be picked, rotated, and dropped using VR controllers.	**LO3**
11	Implement hand gesture recognition to trigger actions like scaling or rotating objects.	**LO3**
12	Use Vuforia or AR Foundation to detect image markers and display a 3D model on top of the marker.	**LO4**
13	Develop an AR application that measures distance between two real-world points using ARCore plane detection.	**LO3**
14	Create VR scenes demonstrating different lighting types (directional, point and spot light) and shadows.	**L03**

15	Create a simple networked VR environment where multiple users interact using Photon networking.	**LO3**

Textbooks/ Reference books:

1. Dieter Schmalsteig and Tobias Hollerer, "Augmented Reality- Principles and Practice", Pearson Education,1st editon,2016
2. Chetankumar G Shetty, "Augmented Reality- Theory, Design and Development", Mc Graw Hill,1st editon,2020
3. Alan B. Craig, "Understanding Augmented Reality – Concepts and Applications", Morgan Kaufmann, Elsevier,1st edition,2013

Online Resources:

1. https://nptel.ac.in/
2. http://msl.cs.uiuc.edu/vr/
3. http://lavalle.pl/vr
4. http://nptel.ac.in
5. www.coursera.org

Software Tools

1. Unity
2. Python

Term Work:

- Term work should consist of at least 12 experiments.

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theo	Practical	Tutorial	Theory	Practica	Tutorial	Total
MDML 6032	Sensor Technology Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 6032	Sensor Technology Lab	--	--	--	25	--	25

Pre-requisite:

Engineering
Physics-II Digital
Signal Design
Microprocessor and Microcontroller

Program Outcomes Addressed

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions.
4. PO5: Engineering Tool Uses
5. PO11: Life-long learning

Lab Objectives:

1. To study fundamental sensor interfacing circuits
2. To study the working principles and characteristics of various sensors and transducers
3. To develop skills to interface microcontroller with various sensors.
4. To implement practical sensor-based systems.
5. To verify sensor circuits to understand behavior, calibration, and performance under different conditions.
6. To apply data processing and basic machine learning techniques.

Lab Outcomes:

Upon completion of this course, Learners will be able to:

1. Apply to the principle of voltage division in analog circuits.
2. Measure physical parameters using sensors and plot input–output characteristics.
3. Demonstrate real-time sensing and communication using microcontrollers.
4. Analyze angular velocity and vibration using gyroscope and accelerometer
5. Design and develop automation circuits

6. Simulate sensors and circuits using software tools.

Suggested List of Experiments: Students are required to complete the 12 experiments.

Sr. No.	Title of Experiments	LO
1	Basics of the Voltage Divider Circuit and the Wheatstone Bridge: Design and demonstrate how a voltage divider circuit and a Wheatstone bridge can be used to detect and measure small changes in resistance in strain gauges. Analyze which method provides higher sensitivity and justify your choice.	**LO1**
2	Displacement measurement using an Inductive Transducer (LVDT): Illustrate how an LVDT (Linear Variable Differential Transformer) can be used to measure piston displacement. Examine its working, output characteristics, and suitability for industrial environments.	**LO2**
3	Characterization of temperature sensors, such as thermocouples, thermistors, and resistance temperature detectors. Compare and evaluate thermocouples, thermistors, and RTDs for this application. Determine which sensor is most appropriate based on accuracy, response time, and stability, and justify your selection with relevant characteristics.	**LO2**
4	Interfacing Arduino / Raspberry Pi /ESP32 with sensors.: Develop and demonstrate how an Arduino / Raspberry Pi / ESP32 can be interfaced with multiple sensors to collect and transmit environmental data. Illustrate the system architecture and explain data acquisition and communication methods.	**LO3**
5	Distance measurement using an ultrasonic sensor: Design and implement a system using an ultrasonic sensor to measure distance in real time. Analyze the working principle, calculate distance using echo time, and evaluate factors affecting accuracy in a real-world parking scenario.	**LO3**
6	Interfacing a Gas Sensor with Arduino/ESP32 for Real-Time Air Quality Monitoring: Design and demonstrate a gas sensor interfaced with an Arduino or ESP32 to monitor air quality in real time. Illustrate the system architecture, including sensor calibration, data acquisition, and wireless transmission. Analyze how the system can be used to trigger alerts when pollutant levels exceed safe thresholds in a real-world urban environment.	**LO3**
7	Angular velocity measurement using a Gyroscope sensor: Develop and implement a system to measure and analyze vibrations using an accelerometer sensor. Interface an accelerometer with a microcontroller. Record acceleration data along different axes. Convert acceleration signals into vibration parameters (frequency, amplitude). Plot vibration signals and identify patterns. Evaluate system performance under different vibration sources.	**LO4**

8	Vibration measurement using an Accelerometer: Design a system to measure and analyze vibrations using an accelerometer sensor. Interface an accelerometer with a microcontroller. Record acceleration data along different axes. Convert acceleration signals into vibration parameters (frequency, amplitude). Plot vibration signals and identify patterns. Evaluate system performance under different vibration sources. Display or log the angular velocity readings in real time. Analyze the accuracy of the sensor under different motion conditions.	**LO4**
9	Develop a sensor-based system to measure force/weight using a load cell: Interface a load cell with an amplifier (e.g., HX711) and microcontroller. Calibrate the system using known weights. Measure unknown forces accurately. Display readings on an LCD or serial monitor. Analyze linearity, sensitivity, and error in measurements.	**LO4**
10	Design and implement a circuit to detect day and night conditions using a Light Dependent Resistor (LDR): Build a voltage divider circuit using an LDR and a resistor. Set a threshold to distinguish between light and dark conditions. Use a transistor or microcontroller to control an output (LED or relay). Analyze how the resistance of LDR changes with light intensity. Test the circuit under different lighting conditions.	**LO4**
11	Simulate a voltage divider circuit using Python and verify theoretical and simulated results: Write a Python program to calculate the output voltage for given resistor values. Compare computed results with theoretical calculations. Analyze how the output voltage varies with different resistor combinations.	**LO5**
12	Introduction to Sensor Simulation Platforms Tools: To understand the fundamentals of sensor interfacing by creating and testing basic sensor circuits using the Tinkercad simulation platform, and to observe how simulated sensor inputs interact with microcontroller-based systems without requiring physical hardware.	**LO5**
13	Simulated Temperature & Humidity using Measurement Tools: To simulate temperature and humidity measurement using virtual sensors in Tinkercad , interface them with a microcontroller (e.g., ESP 430, Arduino, etc.), and display the sensed temperature and humidity for a weather monitoring station	**LO5**
14	Simulate Light & Proximity Sensors using Measurement Tools: To design and simulate circuits using light and proximity sensors in Tinkercad, observe sensor response variations with changing environmental conditions, and analyze their role in Automatic Street lighting systems or Touchless switching systems	**LO5**
15	Sensor Data Processing, Analysis & Noise Filtering Using Python (NumPy, Pandas): To process simulated sensor datasets using Python libraries such as NumPy and Pandas, perform statistical analysis, and apply filtering techniques to reduce noise and improve sensor data accuracy for signal conditions in IOT applications.	**LO6**

16	Sensor Data Classification using Machine Learning Algorithms: To apply machine learning algorithms for classification of sensor data, train models using labeled datasets, and evaluate their performance in identifying patterns for Activity recognition using wearable sensors or Smart healthcare monitoring systems	**LO6**

Textbooks:

1. Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs and Applications", 2016, 5th edition, Springer, New York.
2. John H. Davies, "MSP430 Microcontroller Basics", 2011, 2nd ed., Newnes Publishing, New York.
3. Jonathan W. Valvano, Embedded Systems: Introduction to Arm Cortex-M Microcontrollers, 2017, 5th edition.

Reference books:

1. Sergey Y. Yurish," Digital Sensors and Sensor Systems: Practical Design", 2011, 1st ed., IFSA Publishing, New York.
2. Zach Shelby, Carsten Bormann, "6LoWPAN: The Wireless Embedded Internet", 2009, 1st ed., John Wiley & Sons, New Jersey.
3. Sergey Y. Yurish, Digital Sensors and Sensor Systems: Practical Design, 2011, 1st edition, IFSA Publishing, New York.

Online Resources:

1. https://www.arduino.cc/en/software/

Software Tools

1. Arduino IDE, Tinker Cad, Python / Jupiter

Term Work:

- Term work should consist of atleast 12 experiments. (10 hardware + 02 simulation experiments)

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and a minimum passing mark in term work. Term work will be assessed as a Continuous Internal Assessment Practical (CIAP).

Term work Marks (CIAP):

- 25 Marks (Total Marks) =20 Marks (Experiments) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Teaching Scheme			Credits Assigned			
		Theory	Practica	Tutorial	Theory	Practical	Tutoria	Total
MDML 6042	Cloud Computing and Security Lab	--	02	--	--	01	-	01

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
MDML 6042	Cloud Computing and Security Lab	--	--	--	25	--	25

Pre-requisite: Computer Networks and Operating Systems

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design and development of solutions
4. PO5: Modern tool usage
5. PO9: Individual and teamwork
6. PO11: Lifelong learning

Lab Objectives:

1. To make students familiar with key concepts of virtualization.
2. To make students familiar with various service models such as IaaS, SaaS, and PaaS.
3. To make students familiar with Security as a Service (SECaaS).
4. To make students familiar with the concept of open cloud platforms, containerization, and orchestration.

Lab Outcomes:

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

1. Implement different types of virtualization techniques.
2. Analyze various cloud computing service models and implement them to solve the given problems.
3. Identify major security issues in the cloud and provide appropriate security solutions.
4. Explore various open cloud platforms and recommend the appropriate one for the given application.
5. Implement Serverless architecture and configure workflows.
6. Implement the concept of containerization and orchestration.

Suggested List of Experiments:		
Sr. No.	**Title of Experiments**	**LO**
1	Title: Introduction to Virtualization AIM: To study and implement Hosted Virtualization using VirtualBox& KVM. Objectives: • To know the concept of Virtualization along with their types, structures and mechanisms. • To create and run Virtual machines inside hosted hypervisors like VirtualBox and KVM with their comparison based on various virtualization parameters.	**LO1**
2	Title: To study and implement Bare Metal Hypervisor using Xen Server AIM: To study and implement Bare Metal Hypervisor using Xen Server. Objectives: • Understand Xen architecture and components • Install Xen Hypervisor and related packages • Configure GRUB to boot Xen • Launch Xen Manager and create/manage VMs	**LO1**
3	Title: Introduction to AWS IAM AIM: To understand and manage AWS Identity and Access Management (IAM) by exploring users, groups, and policies. Objectives: • Explore IAM Users and Groups • Inspect IAM policies applied to groups • Add users to groups • Locate IAM sign-in URL • Experiment with policy effects	**LO3**
4	Title: Introduction to Amazon EC2 AIM: To understand and manage Amazon EC2 instances. Objectives: • Launch EC2 instance • Configure security groups • Monitor and resize EC2 instance	**LO2**
5	Title: Working with Amazon EBS AIM: To manage Amazon EBS volumes. Objectives: • Create and attach EBS volume • Format and mount volume • Create snapshot and restore volume	**LO2**
6	Title: Scale and Load Balance Your Architecture AIM: To implement ELB and Auto Scaling.	**LO2**

	Objectives: • Create load balancer • Configure Auto Scaling group • Monitor performance using CloudWatch	
7	Title: Build Your VPC and Launch a Web Server AIM: To design and deploy a custom VPC and launch EC2 instance. Objectives: • Create VPC and subnets • Configure security group • Launch EC2 instance in VPC	**LO2**
8	Title: Introduction to Amazon S3 AIM: To learn basic features of Amazon S3. Objectives: • Create S3 bucket • Upload objects • Manage permissions and bucket policies	**LO2**
9	Title: Build Your DB Server and Interact With Your DB Using an App AIM: To deploy and configure Amazon RDS. Objectives: • Launch RDS instance • Configure DB connectivity • Interact with DB using web app	**LO2**
10	Title: Securing Applications by Using Amazon Cognito AIM: To secure applications using Amazon Cognito. Objectives: • Create user pool • Configure identity pool • Integrate authentication in app	**LO3**
11	Title: Encrypting Data at Rest by Using AWS Encryption Options AIM: To implement encryption using AWS KMS. Objectives: • Review default encryption • Create KMS key • Encrypt EBS and S3 data	**LO3**
12	Title: Automating Infrastructure Deployment with AWS CloudFormation AIM: To automate deployment using CloudFormation. Objectives: • Deploy VPC and app layers • Explore templates in Designer • Delete stack with retention policy	**LO2**
13	Title: Building Decoupled Applications by Using Amazon SQS AIM: To build decoupled architecture using SQS and SNS. Objectives:	**LO2**

	• Configure S3 events to SNS • Subscribe SQS to SNS • Implement message polling	
14	Title: Implementing a Serverless Architecture on AWS AIM: To design and implement serverless architecture using AWS Lambda. Objectives: • Invoke Lambda from S3/DynamoDB • Configure SNS notifications • Build serverless workflow	**LO5**
15	Title: Installation and Configuration of OpenStack AIM: To install and configure OpenStack using DevStack. Objectives: • Understand OpenStack architecture • Install OpenStack using DevStack • Access Horizon dashboard and launch instances	**LO4**
16	Title: To study and Implement Containerization using Docker AIM: To know the basic differences between Virtual machine and Container. Objectives: • Demonstrate creating, finding, building, installing, and running containers • Work with Linux/Windows application containers on local or cloud platform	**LO5**
17	Title: To study and implement container orchestration using Kubernetes AIM: To understand steps to deploy Kubernetes Cluster and manage applications. Objectives: • Deploy Kubernetes Cluster on local systems • Deploy applications on Kubernetes • Create a Service in Kubernetes • Develop Kubernetes configuration files in YAML • Create a deployment in Kubernetes using YAML	**LO6**
18	Title: To Train a Machine Learning Model using Amazon SageMaker AIM: To train a machine learning model using Amazon SageMaker by applying the XGBoost algorithm on a biomechanical vertebral column dataset. Objectives • Split data into training, validation, and test datasets • Train an XGBoost model in Amazon SageMaker	**LO2**
19	Title: To develop Machine Learning Pipeline with Amazon SageMaker AIM: To deploy a trained machine learning model using Amazon SageMaker, perform predictions, and execute batch transformations on the test dataset. Objectives:	**LO2**

	• Deploy a machine learning model • Use the test dataset to perform a batch transformation with the	

Text books

1. R. Buyya, C. Vecchiola, and S. T. Selvi, Mastering Cloud Computing, 1st ed. New Delhi, India: McGraw-Hill Education, 2017.
2. A. Vithal and B. Jadhav, Cloud Computing and Services. Mumbai, India: SYBGEN Learning India Pvt. Ltd., Star Edu Solutions.
3. A. Srinivasan and J. Suresh, Cloud Computing: A Practical Approach for Learning and Implementation, 1st ed. New Delhi, India: Pearson, 2014.
4. R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, 1st ed. Hoboken, NJ, USA: Wiley, 2010.
5. B. Sosinsky, Cloud Computing Bible, 1st ed. Hoboken, NJ, USA: Wiley Publishing, 2011.

Reference books

1. K. Jayaswal, J. Kallakurchi, D. J. Houde, and D. Shah, Cloud Computing Black Book, 2nd ed. New Delhi, India: DreamTech Press, 2014.
2. M. Wittig and A. Wittig, Amazon Web Services in Action, 2nd ed. Shelter Island, NY, USA: Manning Publications.
3. P. Arora, R. Biyani, and S. Dave, To the Cloud: Cloud Powering an Enterprise, 1st ed. New Delhi, India: Tata McGraw-Hill Education, 2012.
4. K. Hwang, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, 1st ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

Term Work:

- The term work should include 12 experiments from the list.

The experiments can also be performed using Microsoft Azure and Google cloud platform for similar kind of services. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

Term work Marks:

- 25 Marks (Total Marks) =20 Marks (Experiment) + 05 Marks (Attendance)
- The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work.

Course Code	Course Name	Credits Assigned			
		Theory	Practical	Tutorial	Total
AIDSP601	Major Project I	--	2	--	2

Course Code	Course Name	Examination Scheme					
		Theory Marks			CIAP	ESEP	Total
		Course Assessment		ESE			
		ISE	MSE				
AIDSP601	Major Project I	--	--	--	25	25	50

Program Outcomes addressed:

1. PO1: Engineering knowledge.
2. PO2: Problem Analysis.
3. PO3: Design/Development of Solutions.
4. PO4: Conduct Investigations of Complex Problems.
5. PO5: Engineering Tool Usage.
6. PO6: The Engineer and The World.
7. PO7: Ethics.
8. PO8: Individual and Collaborative Team work.
9. PO9: Communication.
10. PO10: Project Management and Finance.
11. PO11: Life-Long Learning.

Objectives:

1. Investigate and evaluate prominent literature to come with application-oriented project topics in connection with the curriculum.
2. Study and develop an outline for thinking and practice that illuminates and brings insight to the design and implementation aspects with respect to the project topic.
3. Design and create practical resources and solution aspect for the design and implementation.
4. Present an organised exploratory framework, while understanding the documentary deliverables within established academic practices and/ or ideas.
5. Offer inquiry-based argumentation / presentation along with project implementation
6. To promote self-learning and research-oriented thinking.

Outcomes: Upon completion of this course, learners will be able to:

1. Perform extensive Review of Literature from diverse knowledge banks or through interactions with Industry experts.
2. Developing or Creating ideas capable of addressing industrial or social solutions to identified problem domains.
3. Acquire knowledge of tools & technologies and application of their expertise in creating project implementation and deliverables.
4. Implement solutions using appropriate technologies, engineering practices, and version control.
5. Test and validate the system through structured test cases, datasets, and simulations.
6. Compose technical documentation, present project outcomes, and demonstrate teamwork, ethics, and self-learning.

1. Guidelines for Project topic selection Process to be defined and followed:

1.1 General Guidelines

a. Project orientation can be given at the end of fifth semester.
b. Students should be informed about the domain and domain experts whose guidance can be taken before selecting projects.
c. Students can select the problem statement from the Industry and provide technology solution as capstone project. Capstone projects in engineering study are considered important as it allow students to integrate and apply the knowledge and skills acquired throughout their academic program and effectively demonstrating their learning of programme by tackling a real-world problem, ultimately keeping them well prepared for the job market
d. Student‘s should be recommended to refer papers from reputed conferences/ journals like IEEE, Elsevier, ACM etc. which are not more than 3 years old for review of literature.
e. Students can certainly take ideas from anywhere, but be sure that they should evolve them in the unique way to suit their project requirements. Students can be informed to refer Digital India portal, SIH portal or any other hackathon portal for problem selection.

1.2 Topics can be finalized with respect to following criterion:

a. Topic Selection: The topics selected should be novel in nature (Product based, Application based or Research based) or should work towards removing the lacuna in currently existing systems.
b. Technology Used: Use of latest technology or modern tools can be encouraged.
c. Students should not repeat work done previously (work done in the last three years).
d. Project work must be carried out by the group of at least 2 students and maximum 4.
e. The project work can be undertaken in a research institute or organization/Industry/any business establishment. (out-house projects)
f. The Capstone Project- problems must be related to the programme or may be interdisciplinary, based on the industry expected outcomes.The individual students have different aptitudes and strengths. Project work, therefore, should match the strengths of students. For this purpose, students should be asked to identify the type of project work they would like to execute.
g. Project shall address National Thrust area such as Environment, Digitization, Automation, sustainability and predefined Sustainable development goals(SDGs).
h. The project proposal presentations can be scheduled according to the domains and should be judged by faculty who are expert in the domain.
i. Head of department and senior staff along with project coordinators will take decision regarding final selection of projects.
j. Guide allocation should be done and students have to submit weekly progress report to the internal guide.
k. Internal guide has to keep track of the progress of the project and also has to maintain attendance report. This progress report can be used for awarding term work marks.
l. In case of industry/ out-house projects, visit by internal guide will be preferred and external members can be called during the presentation at various levels.

1.3 Deployment Requirement

a. Each project must include a deployment phase, ensuring the solution is usable in a real-world or simulated environment.

Examples of deployment include:

- Web application hosted on cloud/server
- Mobile application (APK/Test version/Play Store)
- Industrial automation system implemented or simulated
- AI/ML model deployed via API or dashboard
- IoT system with real-time monitoring and control

1. Guidelines for Assessment of Major Project I:

1.1 Term Work Evaluation (CIAP)

a. A Review/Progress Monitoring Committee shall be constituted by the Head of Department. **Major Project I** shall be evaluated continuously, with a minimum of two reviews per semester.
b. Continuous assessment should also emphasize individual performance, including each student's contribution, understanding of the project, and responses during review interactions.
c. Project report to be submitted as per the guidelines issued by the department.
d. Term work will be assessed as Continuous Internal Assessment Practical (CIAP).

2.2 Distribution of marks

The distribution of Term Work marks for the semester shall be as follows:

a. Marks by Guide/Supervisor based on Logbook: 10
b. Marks by Review Committee: 10
c. Quality of Project Report: 05

2.3 Evaluation Rubrics CIAP (Term work);

a. Quality of literature survey and need identification
b. Clarity of problem definition
c. Innovation and creativity
d. Feasibility and justification of approach
e. Cost effectiveness and societal/environmental impact (SDGs)
f. Quality of system design and use of engineering norms
g. Implementation quality and functioning of prototype
h. Effective use of technical skill sets and tools
i. Teamwork, leadership, and individual contribution
j. Clarity in documentation, presentation, and communication
k. Deployment readiness and implementation feasibility
l. Usability, scalability, and real-world applicability of the deployed solution

2.4 ESEP (Practical/Oral) Examination:

a. **Major Project I** shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organizations having experience of more than five years approved by the head of Institution.
b. Students shall be motivated to publish a paper based on the work in Conferences/students

competitions.

c. End-semester Practical / oral exam will be held based on the above syllabus and will be conducted as End Semester Examination Practical (ESEP).

2.5 Evaluation Rubrics ESEP (Practical / Oral):

a. Problem identification and clarity
b. Innovation in solution
c. Feasibility and societal impact
d. Working model functionality
e. Effective use of engineering tools
f. Adherence to standards and norms
g. Teamwork and individual contribution
h. Communication skills
i. Deployment quality (hosting, execution, usability, and performance)