Shri Virle Parle Kelwani Mandal's

NMIMS Global University, Dhule

School of Technology, Management & Engineering

Survey No. 499, Plot No 02, Behind Gurudwara,
Mumbai Agra National Highway,
Dhule -424 001, Maharashtra, India

Curriculum Structure and Syllabus
of
Master of Technology (M. Tech) (Regulation 2025)

Effective from Academic Year 2025-26

Vision

"SVKM NMIMS Global University aspires to be a world-renowned institution of higher learning, dedicated to fostering excellence in education, research and innovation with social responsibility"

Mission

1. To foster a dynamic and inclusive learning environment that nurtures the educational and research aspirations of students from diverse backgrounds.
2. To deliver state-of-the-art pedagogy, emphasizing interdisciplinary collaboration and innovation.
3. To uphold the values of ethics and community engagement to cultivate global citizens and leaders who actively contribute to the advancement of society and the world.

INDEX

CURRICULUM FRAMEWORK

(Regulation 2025)

LIST OF ABBREVIATIONS

Sr. No.	Abbreviation	Type of Course
1	PCC	Programme Core Course (PCC)
2	PEC	Programme Elective Course (PEC)
3	VSEC	Vocational and Skill Enhancement Course (VSEC)
4	ELC	Experiential Learning Courses (ELC)

COURSE WISE CREDIT DISTRIBUTION

Sr. No.	Type of Course	No. of Courses	Credits	
			No.	%
1	Programme Core Course (PCC)	8	22	27.5
2	Programme Elective Course (PEC)	4	12	15
3	Vocational and Skill Enhancement Course (VSEC)	2	4	5
4	Experiential Learning Courses (ELC)	7	42	52.5
	TOTAL	**21**	**80**	**100**

SEMESTER-WISE COURSE DISTRIBUTION

Course Distribution : Semester Wise						
Sr. No.	**Type of Course**	**No. of Courses / Semester**				**Total**
		1	**2**	**3**	**4**	
1	Programme Core Course (PCC)	5	3			8
2	Programme Elective Course (PEC)	1	3			4
3	Vocational and Skill Enhancement Course (VSEC)	1	1			2
4	Experiential Learning Courses (ELC)	1	1	3	2	7
	Total	**8**	**8**	**3**	**2**	**21**

SEMESTER-WISE CREDIT DISTRIBUTION

Credit Distribution : Semester Wise						
Sr. No.	**Type of Course**	**No. of Credits / Semester**				**Total**
		1	**2**	**3**	**4**	
1	Programme Core Course (PCC)	14	8			22
2	Programme Elective Course (PEC)	3	9			12
3	Vocational and Skill Enhancement Course (VSEC)	2	2			4
4	Experiential Learning Courses (ELC)	1	1	20	20	42
	Total	**20**	**20**	**20**	**20**	**80**

Curriculum Structure

First Year

M. Tech.

Computer Engineering

CURRICULUM STRUCTURE

First Year M. Tech Semester – I

First Year M. Tech. (Regulation 2025 - With effect from Academic Year 2025-26)

Semester-I

Sr. No.	Course Code	Course	Teaching Scheme				End Semester Examination						Continuous Assessment			Aggregate (A+B)	Category	Mode of Examination	Template
			Theory (hrs.)	Practical (hrs.)	Tutorial (hrs.)	Credits	Duration (hrs.)	Theory	Oral	Practical	Oral & Practical	ESE Total (A)	Term Test	Term Work	CA Total (B)				
1	T25COS501	Advance Operating System	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25COS501L	Advance Operating System Laboratory	-	2	-	1	2	-	-	-	25	25	-	25	25	50	Lab	Lab	
2	T25CDS502	Advance Database System	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CDS502L	Advance Database System Laboratory	-	2	-	1	2	-	-	-	25	25	-	25	25	50	Lab	Lab	
3	T25CPS503	Probability and Statistics	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
5	T25CML504	Machine Learning	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
6	T25CCN505	Computer Network Design and Analysis	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CAI505	Artificial & Computational Intelligence	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CSD505	Software Development and Version Control	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
7	T25CDV506	Machine Learning and Data Analytics & Visualization Laboratory	-	4	-	2	2	-	-	-	50	50	-	50	50	100	Lab	Lab	
8	T25CSM507	Seminar-I	-	2	-	1	2	-	-	-	-	-	-	50	50	50	Term Work	-	
Total			**15**	**10**	**-**	**20**	**18**	**300**	**-**	**-**	**100**	**400**	**200**	**150**	**350**	**750**			

First Year M. Tech Semester – II

First Year M. Tech (Regulation 2025) (With effect from Academic Year 2025-26)

Semester-II

Sr. No.	Course Code	Course	Teaching Scheme				End Semester Examination						Continuous Assessment			Aggregate (A+B)	Category	Mode of Examination	Template
			Theory (hrs.)	Practical (hrs.)	Tutorial (hrs.)	Credits	Duration (Hrs)	Theory	Oral	Practical	Oral & Practical	ESE Total (A)	Term Test	Term Work	CA Total (B)				
1	T25CAC 551	Algorithms and Complexity Theory	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CAC 551L	Algorithms and Complexity Theory Laboratory	-	2	-	1	2	-	-	-	25	25	-	25	25	50	Lab	Lab	
2	T25CSC 552	Security in Computing	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CSC 552L	Security in Computing Laboratory	-	2	-	1	2	-	-	-	25	25	-	25	25	50	Lab	Lab	
3	T25CDL 553	Deep Learning	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CBT 553	Blockchain Technology	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CCI 553	Cloud Infrastructure and System Design	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
4	T25CBD 553	Big Data Analytics	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CNL 553	Natural Language Processing	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CAC 553	Advanced Computer Vision	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
5	T25CIT 554	Internet of Things	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CHP 554	High Performance Computing	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
	T25CSC 554	Soft Computing	3	-	-	3	2	60	-	-	-	60	40	-	40	100	Theory	Written	
6	T25CSI5 55	SmartTech	-	4	-	2	2	-	-	-	50	50	-	50	50	100	Lab	Lab	

		Integration Laboratory																	
7	T25CSM556	Seminar-II	-	2	-	1	2	-	-	-	-	-	-	50	50	50	Term Work	-	
Total			**15**	**10**	**-**	**20**	**18**	**300**	**-**	**-**	**100**	**400**	**200**	**150**	**350**	**750**			

Curriculum Structure
Second Year
M. Tech.
Computer Engineering

Second Year M. TECH Semester – III

First Year M. Tech (Regulation 2025) (With effect from Academic Year 2025-26)																	
Semester-III																	
Sr. No.	**Course Code**	**Course**	**Teaching Scheme**				**End Semester Examination**						**Continuous Assessment**			**Aggregate (A+B)**	**Credits earned**
			Theory (hrs.)	**Practical (hrs.)**	**Tutorial (hrs.)**	**Credits**	**Duration (Hrs)**	**Theory**	**Oral**	**Pract**	**Oral & Pract**	**ESE Total (A)**	**Term Test**	**Term Work**	**CA Total (B)**		
1	T25CMEL601	Seminar-III : Synopsis	0	4	0	2	2	0	0	0	0	0	0	100	100	100	2
2	T25CMEL602	Research Paper Publication -I	0	4	0	2	2	0	0	0	0	0	0	100	100	100	2
3	T25CMEL603	Dissertation Phase -I	0	32	0	16	2	0	0	0	200	200	0	300	300	500	16
Total			**0**	**40**	**0**	**20**	**6**	**0**	**0**	**0**	**200**	**200**	**0**	100	**500**	**700**	**20**

Second Year M. Tech Semester – IV

First Year M. Tech (Regulation 2025) (With effect from Academic Year 2025-26)																	
Semester-IV																	
Sr. No.	**Course Code**	**Course**	**Teaching Scheme**				**End Semester Examination**						**Continuous Assessment**			**Aggregate (A+B)**	**Credits earned**
			Theory (hrs.)	**Practical (hrs.)**	**Tutorial (hrs.)**	**Credits**	**Duration (Hrs)**	**Theory**	**Oral**	**Pract**	**Oral & Pract**	**ESE Total (A)**	**Term Test**	**Term Work**	**CA Total (B)**		
1	T25CMEL651	Research Paper Publication -II	0	4	0	2	2	0	0	0	0	0	0	100	100	100	2
2	T25CMEL652	Dissertation Phase -II	0	36	0	18	2	0	0	0	300	300	0	300	300	600	18
Total			**0**	**40**	**0**	**20**	**4**	**0**	**0**	**0**	**300**	**300**	**0**	**400**	**400**	**700**	**20**

Semester – I

Program:	**M. Tech. (Computer Engineering)**	**Semester: I**
Course:	**Advance Operating System**	**Code: T25COS501**

Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

This course aims at enabling students:

1. To learn the design principles of Operating System.
2. To learn the Problems of Synchronization in Operating System.
3. To learn concepts of Memory Management in Operating System.
4. To learn concepts of Distributed Computing Environment.

Course Outcomes:

After learning the course, the students will be able to:

CO1 To Comprehend and Use basic concepts of Operating System with its structure.

CO2 To Illustrate concepts of Process as well as Thread Management along with Implement concepts of CPU Scheduling algorithms.

CO3 To Illustrate concepts of Process Synchronization as well as deadlock along with Implement concepts of Synchronization primitives and banker’s algorithms.

CO4 To Comprehend concept of Memory Management along with Implement concepts of page replacement algorithms and memory allocation algorithms.

CO5 To Describe the Distributed Computing Environment and use various distributed algorithm related to resource management.

Detailed Syllabus

Unit	**Description**	**Duration (H)**
I	Types of Operating Systems, Operating System Structures. Operating- System Services, System Calls, Virtual Machines, Operating System Design and Implementation. Process Management: Process Concepts, Operations On Processes, Cooperating Processes, Threads, Inter Process Communication, Process Scheduling, Scheduling Algorithms, Multiple-Processor Scheduling. Thread Scheduling.	8
II	The Critical Section Problem, Semaphores, And Classical Problems Of Synchronization, Critical Regions, Monitors, Deadlocks,-System Model, Deadlocks Characterization, Methods For Handling Deadlocks, Deadlock- Prevention, Avoidance, Detection and Recovery from Deadlocks.	8

III	Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Continuous Memory Allocation, Fixed and variable partition, Internal and external fragmentation and compaction, Paging: Principle of operation, Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging; Segmentation. Virtual Memory: Basics of Virtual Memory –Page fault, Demand paging, Page Replacement algorithms: Optimal, first in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used(LRU).	8
IV	Introduction to Distributed Computing Environment (DCE), Desirable Features of a Good Message-Passing System, Issues in IPC by Message-Passing, RPC model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC messages, Marshaling arguments and Results, Server Management, Parameter Passing Semantics, Call Semantics, Communication Protocols for RPCs.	8
V	Desirable Features of a Good Global Scheduling Algorithm, Task assignment Approach, Load-Balancing Approach, load Sharing Approach, Process Migration, Threads.	7
Total		**39**

Text Books:

1. Silberschatz & Galvin, ‘Operating System Concepts’, Wiley.
2. DISTRIBUTED SYSTEMS, Second edition, Andrew S.Tanenbaum, Maarten Van teen.

Reference Books:

1. William Stallings-“Operating Systems”- 5th Edition - PHI
2. Charles Crowley, ‘Operating Systems: A Design-Oriented Approach’, Tata Hill Co., 1998 edition.
3. Andrew S.Tanenbaum, ‘Modern Operating Systems’, 2nd edition, 1995, PHI.
4. Advanced Concepts in Operating systems.Distributed, Database and Multiprocessor operating systems, Mukesh singhal, Niranjan G.Shivaratri, Tata McGraw Hill Edition.

e-sources: NPTEL/SWAYAM Course (if any)

1. https://training.linuxfoundation.org/resources/free-courses/

2. https://training.linuxfoundation.org/training/introduction-to-linux/

Program:	**M. Tech. (Computer Engineering)**				**Semester: I**	
Course:	**Advance Operating System Laboratory**				**Code: T25COS501L**	
Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**		
	Lecture	**Practical**	**Tutorial**	**ISE**	**ESE** **(OR & PR)**	**Total**
01	**-**	**02**	**-**	**25**	**25**	**50**

Prior knowledge of

Course Objectives:

This course aims at enabling students:

1. To familiarize students with UNIX/Linux system calls and inter-process communication mechanisms
2. To learn the Problems of Synchronization in Operating System.
3. To Cultivate analytical and problem-solving skills to evaluate system performance metrics and handle concurrency, deadlocks, and resource management issues.
4. To Enable students to design and implement algorithms for CPU scheduling, synchronization, and memory allocation.

Course Outcomes:

After learning the course, the students will be able to:

CO1: Apply UNIX system calls to create, manage, and synchronize processes effectively.

CO2: Implement and analyze CPU scheduling, memory management, and file allocation algorithms.

CO3: Demonstrate the use of inter-process communication mechanisms such as pipes, shared memory, and message queues.

CO4: Develop and test solutions to classical synchronization problems using semaphores and monitors.

CO5: Evaluate and compare the performance of different operating system mechanisms in terms of efficiency and resource utilization.

Detailed Syllabus

Sr. No.	**Practical Description**
1	Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, stat, opendir, readdir
2	Write programs using the I/O System calls of UNIX operating system (open, read, write, etc.).
3	Write C programs to simulate UNIX commands like ls, grep, etc.

4	Given the list of processes, their CPU burst times and arrival times. Display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
5	Given the list of processes, their CPU burst times and arrival times. Display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
6	Develop application using Inter-Process Communication (using shared memory, pipes or message queues).
7	Implement the Producer-Consumer problem using semaphores (using UNIX system calls)
8	Implement Memory management schemes like paging and segmentation.
9	Implement Memory management schemes like First fit, Best fit and Worst fit.
10	Implement any file allocation techniques (Contiguous, Linked or Indexed).

Text Books:

3. Silberschatz & Galvin, ‘Operating System Concepts’, Wiley.

Reference Books:

5. William Stallings “Operating Systems”- 5th Edition - PHI
6. Charles Crowley, ‘Operating Systems: A Design-Oriented Approach’, Tata Hill Co., 1998 edition.
7. Andrew S.Tanenbaum, ‘Modern Operating Systems’, 2nd edition, 1995, PHI.
8. Advanced Concepts in Operating systems.Distributed, Database and Multiprocessor operating systems, Mukesh singhal, Niranjan G.Shivaratri, Tata McGraw Hill Edition.

e-sources:

3. https://training.linuxfoundation.org/resources/free-courses/

4. https://spoken-tutorial.org/BASH

5. https://spoken-tutorial.org/LINUX

Program:	**M. Tech. (Computer Engineering)**					**Semester: I**	
Course:	**Advance Database Systems**					**Code: MCOT25PC02**	
Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

This course aims at enabling students:

1. To understand the underlying principles of Relational Database Management Systems
2. To understand and apply secure database design and advanced data models.
3. To build and maintain efficient database systems that handle different types of data using current trends and tools.

Course Outcomes:

After learning the course, the students will be able to:

CO1 To Design a database depending on the business requirements, considering various design issues.

CO2 To Analyze and compare how databases use indexing, transactions, access control, and recovery to improve performance and reliability.

CO3 To Analyze and compare how parallel and distributed databases support real-time applications for better speed and reliability.

CO4 To Analyze how databases are used on the web and how they handle semi-structured data.

CO5 To Analyze database security and advanced data models for modern applications.

Detailed Syllabus

Unit	Description	Duration (H)
I	**Relational Databases and Design Theory** Integrity Constraints revisited, Extended ER diagram, Problems Caused by Redundancy, Decompositions, Problems Related to Decomposition, Reasoning About FD's, FIRST, SECOND, THIRD Normal Form, BCNF, Fourth Normal Form, Lossless Join Decomposition, Dependency Preserving Decomposition, Schema Refinement in Data Base Design, Multi Valued Dependencies.	8
II	**Query Processing and Transaction Management** Storage and File Structure, Indexing, Query processing, and Query Optimization, Transaction Management, formalization of a transaction, ACID properties, classification of transaction. Concurrency Control: definition, execution schedules, examples, locking based algorithms, timestamp	8

	ordering algorithms, deadlock management.	
III	**Parallel Databases and Distributed Databases** Parallel Database Architecture, Data partitioning strategy, Inter-Query, and Intra-Query Parallelism, Distributed Database Features, Distributed Database Architecture, Fragmentation, Replication, Distributed Query Processing, Distributed Transactions Processing	8
IV	**Databases On the Web Domain and Semi-Structured Data** Web interface, XML, structure of XML data, querying XML data, storage of XML data, XML applications, semi-structured data model, indexes for text data.	8
V	**Database Security** Database Security Issues, Security Models, Different threats to databases, Challenges to maintaining database security. **Enhanced Data Models for Advanced Applications** Active database concepts, temporal database concepts, spatial databases: concept and architecture, mobile databases, Geographic Information Systems (GIS)	8
Total		**40**

Text Books:

1. Elmasri and Navathe, Fundamentals of Database Systems
2. Ramakrishnan and Gehrke, Database Management Systems

Reference Books:

1. Abraham Silberschatz, Henry F. Korth, and S. Sudharsan, "Database System Concepts", 7th Edition, McGraw Hill, 2019.
2. Fawcett, Joe, Danny Ayers, and Liam RE Quin. "Beginning XML" Wiley India Private Ltd., 5th Edition, 2012
3. Rigaux, Ph, Michel Scholl, and Agnes Voisard. "Spatial databases: with application to GIS" Morgan Kaufmann, 2002.

e-sources: NPTEL/SWAYAM Course (if any)

1. https://www.coursera.org/specializations/database-systems

Program:	**M. Tech. (Computer Engineering)**	**Semester: I**
Course:	**Probability & Statistics**	**Code: T25CPS503**

Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

After learning the course, the students will be able to:

1. Calculate and interpret the correlation coefficient between two variables.
2. Find probability of given events using addition and multiplication theorem. Apply Bayes theorem Translate real-world problems into probability models.
3. Find expectation and variance of discrete and continuous random variable. Find probability using Binomial, Poisson and Normal distribution.
4. Calculate the simple linear regression equation for a set of data and fit the different curves for given data.
5. Apply the concept of testing of hypothesis for large and small sampling to the engineering problems.

Course Outcomes:

After learning the course, the students will be able to:

1. Calculate and interpret the correlation coefficient between two variables.
2. Find probability of given events using addition and multiplication theorem. Apply Bayes theorem Translate real-world problems into probability models.
3. Find expectation and variance of discrete and continuous random variable. Find probability using Binomial, Poisson and Normal distribution.
4. Calculate the simple linear regression equation for a set of data and fit the different curves for given data.
5. Apply the concept of testing of hypothesis for large and small sampling to the engineering problems

Detailed Syllabus

Unit	**Description**	**Duration (H)**
I	**Introduction to measures of central tendency, dispersion and Skewness:** Central tendency of raw data, Discrete and grouped frequency data, Absolute measures and relative measures of dispersion, Karl Pearson's coefficient of skewness and Bowley coefficient of skewness **Self-Learning Topics:** Graphical representation of data and find various central tendencies of Real data	8

	Correlation Introduction, Types of correlation, Correlation and causation, Methods of studying correlation, Karl Pearson's correlation coefficient, Spearman‘s rank correlation, Coefficient, Properties of Karl Pearson's correlation coefficient and Spearman‘s rank correlation coefficient, Probable errors.	
II	**Probability Theory** Definition of probability: classical, empirical and axiomatic approach of probability, Addition theorem of probability, Multiplication theorem of probability, Bayes’ theorem of inverse probability, Properties of probabilities with proofs, Examples.	8
III	**Random Variable and Probability Distributions** Random variables, Probability distributions, Probability mass function, Probability density function, Mathematical expectation, Properties of expectation and variance with proofs. Theoretical Probability Distributions : Binomial distribution, Poisson distribution, Normal distribution, Fitting of binomial distributions, Fitting of Poisson distributions, Importance of normal distribution, Examples.	8
IV	**Linear Regression Analysis & Curve fitting** Introduction, Linear and non-linear regression, Lines of regression, Derivation of regression lines of y on x and x on y, Angle between the regression lines, Coefficients of regression, Theorems on regression coefficient, Properties of regression coefficient ,least square method, fitting of linear and parabolic curve	8
V	**Testing of Hypothesis** Estimation, types of errors, Hypothesis, Critical Region, Level of Significance, Testing of hypothesis for Small and Large sample, z-test, student’s t-distribution test, Non parametric test- Chi square test, Independence of attributes, goodness of fit.	8
	Total	**40**

Text Books

1. S. C. Gupta, Fundamentals of Statistics, Himalaya Publishing House, 7th Revied and Enlarged Edition, 2016.
2. G. V. Kumbhojkar, Probability and Random Processes, C. Jamnadas and Co., 14th Edition, 2010.
3. Veerarajan T., Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi, 2010.

Reference Books:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. G. Haribaskaran, Probability, Queuing Theory and Reliability Engineering, Laxmi Publications, 2nd Edition, 2009.
3. Murray Spiegel, John Schiller, R. ALU Srinivasan, Probability and Statistics, Schaum's Outlines, 4th Edition, 2013.
4. Kishor S. Trivedi, Probability, Statistics with Reliability, Queuing and Computer Science Applications, Wiley India Pvt. Ltd, 2nd Edition, 2001.
5. Vijay K. Rohatgi, A. K. Md. Ehsanes Saleh, An Introduction to Probability And Statistics, Wiley Publication, 2nd Edition, 2001.
6. Roxy Peck, Chris Olsen, Jay Devore, Introduction to Statistics and Data Analysis, Third Edition, Thomson Books/Cole.
7. Ronald Walpole; Raymond Myers; Sharon Myers; Keying Ye, Probability & statistics for engineers & scientists, 9th edition, Prentice Hall

e-sources: NPTEL/SWAYAM Course (if any)

Program:	**M. Tech. (Computer Engineering)**					**Semester: I**	
Course:	**Machine Learning**					**Code: T25CML504**	
Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

This course aims to enable students:

1. To learn the Types of Machine Learning and to use different Classification and Regression models and their evaluation methods.
2. To learn the Feature Selection and Recommender system.
3. To learn concepts of Artificial Neural Network and Statistical Learning Framework.
4. To Use Ensemble Learning and Clustering Techniques.
5. To Learn Reinforcement Learning.

Course Outcomes:

After learning the course, the students will be able to:

CO1 To explain key concepts of machine learning, differentiate types of learning, and evaluate models using appropriate validation techniques while understanding the bias-variance tradeoff and construct, interpret, and evaluate regression techniques.

CO2 To construct, interpret, and evaluate classification tasks and apply feature selection and dimensionality reduction techniques, and develop basic recommender systems.

CO3 To apply probabilistic models, and support vector machines, to solve classification and regression problems.

CO4 To apply artificial neural networks to solve classification and regression problems using appropriate algorithms and tools

CO5 To analyze and apply advanced machine learning concepts such as ensemble methods, clustering, and reinforcement learning to address complex data-driven problems.

Detailed Syllabus

Unit	**Description**	**Duration (H)**
I	**Introduction:** Introduction to Machine Learning, Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance. **Linear Regression:** Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.	8
II	**Decision tree learning:** Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning,	8

	inductive bias in decision tree learning, issues in decision tree learning. **Instance based Learning:** K nearest neighbor, the Curse of Dimensionality, **Feature Selection:** forward search, backward search, univariate , multivariate feature selection approach, Feature reduction (Principal Component Analysis). **Recommender System:** Content based system, Collaborative filtering based.	
III	**Probability and Bayes Learning:** Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression. **Support Vector Machine:** Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem.	8
IV	**Artificial Neural Networks:** Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm, Loss functions and hyper parameter tuning. **Statistical Learning Framework:** PAC learning, Agnostic PAC learning, Bias-complexity tradeoff, No free lunch theorem, VC dimension, Structural risk minimization, Mistake Bound Find S-algorithm	8
V	**Ensembles:** Introduction, Bagging and boosting, Random forest, AdaBoost, Stacking, Voting. **Clustering:** Introduction, K-mean clustering, agglomerative hierarchical clustering, Model Based Clustering, GMM Clustering, EM algorithm. **Reinforcement Learning**: Exploration vs Exploitation, MDP, Policy, Q-Learning	8
Total		**40**

Text Books:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020.

Reference Books:

1. K.P. Soman, R. Longonathan and V. Vijay, Machine Learning with SVM and Other Kernel Methods, PHI-2009
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006.
3. Shalev-Shwartz,S., Ben-David,S., (2014), Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press
4. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
5. Kishan Mehrotra, Chilukuri Mohan and Sanjay Ranka, Elements of Artificial Neural Networks, Penram International
6. Rajjan Shinghal, Pattern Recognition, Techniques and Applications, OXFORD

e-sources: NPTEL/SWAYAM Course (if any)

1. Introduction to Machine Learning, By Prof. Balaraman Ravindran , IIT Madras https://nptel.ac.in/courses/106106139

2. Introduction to Machine Learning, by Prof. S. Sarkar, IIT Kharagpur https://nptel.ac.in/courses/106105152
3. Machine Learning CS229 Stanford School of Engineering https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU

Program:	**M. Tech. (Computer Engineering)**	**Semester: I**
Course:	**Programme Elective Course – I** **Computer Network Design and Analysis**	**Code: T25CCN505**

Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

This course aims at enabling students:

1. Understand the principles and process of network design and architecture.
2. Analyze user, application, and performance requirements for networks.
3. Model network flows and develop system-level architectures.
4. Design addressing, routing, and performance strategies.
5. Apply network management and evaluate design using metrics.

Course Outcomes:

After learning the course, the students will be able to:

CO1. Design, deploy, and maintain networks using modular and hierarchical principles.

CO2. Analyze and gather user, application, device, and network requirements, develop service metrics, and create specifications for network design.

CO3 Identify, prioritize, and model network flows while designing network architectures using component, reference, and system models.

CO4 Design and implement addressing and routing mechanisms, applying strategies and architectural considerations for efficient network communication.

CO5 Design network management and performance architectures, incorporating management mechanisms, performance metrics, and network layout considerations.

Detailed Syllabus

Unit	**Description**	**Duration (H)**
I	**FUNDAMENTALS OF NETWORK DESIGN** Design Principles - Determining Requirements - Analyzing the Existing Network - Preparing the Preliminary Design - Completing the Final Design Development - Deploying the Network - Monitoring and Redesigning – Maintaining - Design Documentation - Modular Network Design - Hierarchical Network Design.	8

II	**Requirements Analysis**: user-, application-, device-, network- and performance requirements. Requirement Analysis: process; gathering and listing requirements, service metrics development, behavior characterization, RMA -, delay-, capacity-, supplemental performance requirements development. Requirement mapping. Specifications development.	8
III	**Flow analysis**: Basics, flow identification and development, Flow models, flow prioritization, flow specifications. Network architecture: component architectures, reference architecture, architectural models, systems and network architectures.	8
IV	**Addressing and routing architecture**: Fundamentals, Addressing mechanisms, Routing mechanisms, Addressing strategies, Routing strategies. Architectural considerations for addressing and routing.	8
V	**Network Management Architecture:** Objectives and basics, Defining Network Management, Network Management Mechanisms, Architectural considerations for network management architecture. **Performance Architecture**: Objectives and basics, Performance Mechanisms, Architectural considerations for Performance mechanisms. Network layout, Design traceability and Design metrics.	8
Total		**40**

Text Books:

1. James D. McCabe, "Network Analysis, Architecture, and Design" (2/e) Morgan Kaufmann 2003.
2. Gil Held, "Network Design: Principles and Applications (Best Practices)", Auerbach Publications, 1st edition, 2000.

Reference Books:

1. Andrew S. Tenanbaum "Computer Networks", 4th Ed., Pearson Education.
2. James F. Kurose, Keith W. Ross "Computer Networking: A Top-Down Approach" TMH.
3. Diane Tiare and Catherine Paquet, "Campus Network Design Fundamentals", Pearson Education, 1st edition, 2006.

e-sources: NPTEL/SWAYAM Course (if any)

1. https://onlinecourses.nptel.ac.in/noc22_cs114/preview

2. https://onlinecourses.nptel.ac.in/noc22_ee61/preview

3. https://www.apsit.edu.in/network-design-analysis

Program:	M. Tech. (Computer Engineering)	Semester: I
Course:	Programme Elective Course – I Artificial & Computational Intelligence	Code: T25CAI505

Credit	Teaching Scheme (Hrs./Week)			Evaluation Scheme and Marks			
	Lecture	Practical	Tutorial			ESE	Total
				ISE	MSE		
03	03	-	-	20	20	60	100

Prior knowledge of

Course Objectives:

This course aims at enabling students:

To introduce the fundamental concepts, principles, and methodologies of Artificial Intelligence (AI) and Computational Intelligence (CI).

To develop the ability to analyze, design, and implement AI algorithms using heuristic and search-based approaches.

To understand and evaluate various knowledge representation, reasoning, and planning mechanisms in intelligent systems.

To apply computational intelligence techniques such as neural networks, fuzzy logic, for real-world problem solving.

Course Outcomes:

After learning the course, the students will be able to:

CO1 Explain the foundational concepts, historical evolution, and problem-solving paradigms of Artificial and Computational Intelligence.

CO2 Design and implement AI algorithms using heuristic and search-based methods.

CO3 Utilize computational intelligence methods such as neural networks, fuzzy logic, a solve engineering problems.

CO4 Analyze and evaluate real-world problems using appropriate AI and CI approaches, including knowledge representation and probabilistic reasoning.

CO5 Demonstrate research orientation and critical understanding of modern AI topics such as NLP, learning, and planning systems.

Detailed Syllabus

Unit	Description	Duration (H)
I	**Foundations of Artificial Intelligence:** Introduction to Artificial Intelligence, History, and Philosophy of AI. Intelligent agents: structure, environment, types of agents.AI Problem formulation, State-space representation, Problem-solving frameworks. The Concept of Rationality	8

II	**Problem-solving:** Solving Problems by Searching, Problem-Solving Agents, Uninformed search: Depth-first, Breadth-first, Uniform cost search. Informed (Heuristic) search: Greedy search, A* search, AO* algorithm, Local search algorithms: Hill Climbing, Constraint Satisfaction Problems (CSPs), Backtracking algorithms, heuristics for CSPs. Game Playing: Minimax Algorithm, Alpha-Beta Pruning.	8
III	**Knowledge Representation and Reasoning:** Knowledge representation issues, Representation & mapping, Approaches to knowledge representation, Issues in knowledge representation. Using predicate logic: Representing simple fact in logic, Representing instant & ISA relationship, Computable functions & predicates, Procedural verses declarative knowledge, Logic programming, Forward verses backward reasoning	8
IV	**Probabilistic Reasoning:** Representing knowledge in an uncertain domain, The semantics of Bayesian networks, Dempster-Shafer theory, fuzzy logics, Planning: Overview, Components of a planning system, Goal stack planning, Hierarchical planning and other planning techniques.	8
V	**Natural Language Processing:** Introduction, Syntactic processing, Semantic analysis, Discourse & pragmatic processing. Learning: Forms of learning, Inductive learning, Learning decision trees, explanation based learning, Learning using relevance information, Neural net learning & genetic learning.	8
Total		**40**

Text Books:

1. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson Education, 2020.
2. Rich, E. and Knight K.: Artificial Intelligence, Tata McGraw- Hill
3. Andries Engelbrecht, Computational Intelligence: An Introduction, 2nd Edition, Wiley, 2007.

Reference Books:

1. Kevin Warwick, Artificial Intelligence Techniques in Power Systems, IET Publications.
2. Timothy J. Ross, Fuzzy Logic with Engineering Applications, 4th Edition, Wiley, 2016.
3. Melanie Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.

e-sources: NPTEL/SWAYAM Course (if any)

1. https://onlinecourses.nptel.ac.in/noc25_cs07/

2. https://onlinecourses.swayam2.ac.in/nou25_cs07

Program:	**M. Tech. (Computer Engineering)**	**Semester: I**
Course:	**Software Development and Version Control**	**Code: T25CSD505**

Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**			
	Lecture	**Practical**	**Tutorial**			**ESE**	**Total**
				ISE	**MSE**		
03	**03**	**-**	**-**	**20**	**20**	**60**	**100**

Prior knowledge of

Course Objectives:

This course aims to enable students:

1. To develop the ability to choose and apply suitable process models for efficient and adaptive software development.
2. To study the principles of software reuse, component-based software engineering, and service-oriented software engineering.
3. To learn project management techniques, including risk management, teamwork, and project planning.
4. To Explore version control systems, configuration management, and tools for effective software development.

Course Outcomes:

After learning the course, the students will be able to:

CO1 To Apply software engineering practices used in real-world development.

CO2 To Analyze system requirements, models, and designs to identify key components, evaluate development processes, and assess dependability and security for reliable and safe systems

CO3 To Evaluate software components and services for reuse in various types of software systems.

CO4 Analyze project management and software quality processes to identify risks, plan projects, manage teams, and ensure software meets quality standards.

CO5 Analyze and compare version control systems and tools to manage changes and releases in software development.

Detailed Syllabus

Unit	**Description**	**Duration (H)**
I	**Introduction to software Engineering** The Nature of software, The unique nature of WebApps, Software engineering practice, Process Models. Agile Software Development: Extreme Programming (XP), Other Agile Process Models.	8
II	**Requirement Engineering and System Security** Requirement Engineering, System Modeling, Architectural Design, Design and Implementation. System Dependability and Security: Dependable systems, Safety engineering and Security Engineering.	8

III	**Software Engineering: Reuse, Components** Software Reuse, Component based Software Engineering, Distributed Software Engineering, service oriented Software Engineering. Real Time Software Engineering.	8
IV	**Software Management** Project Management: Risk Management, Team Work, Project Planning, Quality Management: Software Quality Management: Software Quality, software standards, reviews and Inspections, Quality Management and agile development, software measurement.	8
V	**Software Configuration Management and Version Control** Version Management, System Building, Change Management, Releases Management. Types of version control, Popular version control systems. Use of version control system. Version control Tools.	8
Total		**40**
Text Books: 1. Sommerville, Ian. Software Engineering, 9/E. Pearson Education India, 2011.		
Reference Books: 1. Roger Pressman software Engineering A practitioner’s Approch 9th Edition McGrowHill		
e-sources: NPTEL/SWAYAM Course (if any)		

Program:	**M. Tech. (Computer Engineering)**				**Semester: I**	
Course:	**Advance Database System Laboratory**				**Code: T25CDS502L**	
Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**		
	Lecture	**Practical**	**Tutorial**	**ISE**	**ESE**	**Total**
01	**-**	**02**	**-**	**25**	**25**	**50**

Prior knowledge of

Course Objectives:

This course aims at enabling students:

1. To design and develop relational database systems using Entity–Relationship (ER) modeling and normalization principles for real-world applications.
2. To implement and analyze advanced SQL operations — including joins, sub-queries, views, stored procedures, triggers, transactions, and query optimization — for efficient and reliable data management.
3. To introduce students to XML-based data modeling and transformation techniques for ensuring interoperability, validation, and querying in distributed and web-based database environments.

Course Outcomes:

After learning the course, the students will be able to:

CO1 : Design and implement relational database models using Entity–Relationship (ER) concept and normalization principles up to 3NF/BCNF for real-world applications.

CO2 : Develop, execute, and analyze advanced SQL programs involving joins, nested queries, views, stored procedures, triggers, transactions, and indexing for performance optimizatio and concurrency control.

CO3 : Model, validate, and transform structured data using XML technologies and demonstrate data interoperability and querying in distributed and web-based environments.

Detailed Syllabus

Experiment No	**Title**
1	Design an ER diagram for a real-world application and implement the corresponding Database Schema.
2	Normalize the given database schema up to Third Normal Form (3NF) or Boyce-Code Normal Form (BCNF) and implement the normalized structure using SQL.
3	Implementation of SQL Commands (DDL, DML, DCL) and Data Retrieval Operations Using SELECT Queries
4	Advanced SQL Queries – Performing Nested Queries, Joins, Set Operations, and Views on University Management System
5	SQL Programming – Implementation of Stored Procedures, Functions, Cursors, Triggers, Exception Handling, and Loops
6	Dependency Analysis and Decomposition – Identifying Functional Dependencies and Demonstrating Lossless Decomposition
7	Transaction Management and Concurrency Control in SQL

8	Indexing and Query Optimization – Creating Indexes, Analyzing Execution Plans, and Optimizing SQL Queries
9	Implementation of Data Fragmentation in Distributed Database Systems
10	XML-based University Management System with Validation, Transformation, and Querying

Text Books:

3. Elmasri and Navathe, Fundamentals of Database Systems
4. Ramakrishnan and Gehrke, Database Management Systems

Reference Books:

4. Abraham Silberschatz, Henry F. Korth, and S. Sudharsan, “Database System Concepts”, 7th Edition, McGraw Hill, 2019.
5. Fawcett, Joe, Danny Ayers, and Liam RE Quin. “Beginning XML” Wiley India Private Ltd., 5th Edition, 2012
6. Rigaux, Ph, Michel Scholl, and Agnes Voisard. “Spatial databases: with application to GIS” Morgan Kaufmann, 2002.

e-sources: NPTEL/SWAYAM Course (if any)

2. https://www.coursera.org/specializations/database-systems

Program:	**M. Tech. (Computer Engineering)**			**Semester: I**		
Course:	**Machine Learning and Data Visualization Lab**			**Code: T25CDV506**		
Credit	**Teaching Scheme (Hrs./Week)**			**Evaluation Scheme and Marks**		
	Lecture	**Practical**	**Tutorial**	**ISE**	**ESE**	**Total**
02	**-**	**04**	**-**	**50**	**50**	**100**

Prior knowledge of Python Programming

Course Objectives:

This course aims to enable students:

1. Introduce students to the core concepts and workflow of data preprocessing, visualization, and supervised machine learning for analytical problem-solving.
2. Develop the ability to design and implement various regression and classification algorithms using appropriate datasets and performance evaluation techniques.
3. Equip students with the knowledge to apply probabilistic learning models, Support Vector Machines (SVM), and ensemble learning methods for efficient prediction and decision-making.
4. Enable students to apply unsupervised learning, including clustering and dimensionality reduction techniques, to discover hidden data patterns and structures.
5. Enhance practical understanding of Artificial Neural Networks (ANNs), recommender systems, and model evaluation frameworks for real-world data-driven applications.

Course Outcomes:

After learning the course, the students will be able to:

CO1: Apply data preprocessing, feature engineering, and visualization techniques using Python libraries (NumPy, Pandas, Seaborn, Matplotlib, Plotly) to prepare and explore datasets.

CO2: Design and evaluate supervised learning models (Regression, Decision Tree, KNN, Naïve Bayes, and SVM) using appropriate metrics and visualizations.

CO3: Implement and compare ensemble learning algorithms such as Random Forest, AdaBoost, and XGBoost to improve predictive performance and model robustness.

CO4: Apply unsupervised learning methods such as K-Means, Hierarchical Clustering, and PCA for pattern discovery and dimensionality reduction in real-world data.

CO5: Develop and optimize Artificial Neural Network (ANN) and Recommender System models, and perform comprehensive model evaluation to identify the most suitable algorithm for a given problem.

Detailed Syllabus

Note: Perform any 10 experiment from Group A and 10 from Group B

Group A Data Visualization Lab

Expt. No.	**Practical Description**
1	Apply data cleaning and transformation techniques to prepare raw data for visualization using Pandas and NumPy (e.g., Titanic / Iris dataset).
2	Analyze datasets using statistical summaries and correlation visualization techniques with Pandas, Seaborn, and Matplotlib (e.g., COVID-19 / Student Performance).
3	Apply data balancing techniques using SMOTE (Synthetic Minority Oversampling Technique) to handle imbalanced datasets and visualize class distributions before and

	after resampling (e.g., Credit Card Fraud / HR Attrition dataset).
4	Create visualizations to interpret distributions and relationships between two variables using Matplotlib and Seaborn (e.g., Automobile / House Prices dataset).
5	Apply visual analysis techniques to multivariate datasets using Seaborn (pairplot, jointplot) and Plotly (e.g., Iris / Wine Quality dataset).
6	Develop visualizations to analyze temporal trends and patterns using Matplotlib and Plotly (e.g., Stock Market / Weather dataset).
7	Use geo-mapping tools to visualize spatially distributed data using Plotly, Folium, and GeoPandas; include Terrain Graph visualization in Folium for representing elevation or topographical features (e.g., COVID-19 / Population / Earthquake / Elevation data).
8	Design and deploy interactive dashboards for data-driven decision-making using Dash (e.g., E-commerce / Sales dataset).
9	Implement visualization methods for grouped and hierarchical data using Plotly, Seaborn, and Matplotlib (e.g., HR Attrition / Sales data).
10	Apply statistical visualization techniques to identify correlations between numerical features using Seaborn and Matplotlib (e.g., Medical Cost / Marketing dataset).
11	Generate animated and interactive plots to illustrate trends and comparisons using Plotly Express (e.g., Gapminder / World Bank data).
12	Construct visualizations for textual datasets such as word frequency and sentiment using WordCloud, NLTK, and Matplotlib (e.g., Twitter Sentiment / News Headlines).
13	Develop an end-to-end visualization mini-project demonstrating applied data visualization skills using Python (e.g., Custom / Kaggle dataset).

Group B Machine Learning Lab

Experiment No.	**Practical Description**
1	Perform Feature Engineering and Data Preprocessing for Machine Learning. Apply techniques such as handling missing values, encoding categorical variables, feature scaling, and feature selection using Pandas, NumPy, and Scikit-learn. Visualize feature distributions before and after transformation using Seaborn and Matplotlib.(e.g., Titanic / Housing Dataset)
2	Develop a program to demonstrate the working of Linear Regression and Polynomial Regression. Use the Boston Housing Dataset for Linear Regression and the Auto MPG Dataset (for vehicle fuel efficiency prediction) for Polynomial Regression.
3	Develop a program to demonstrate the working of the Decision Tree Algorithm. Use the Breast Cancer Dataset for building the decision tree and apply this knowledge to classify a new sample.
4	Develop a program to implement the K-Nearest Neighbor (KNN) Classifier. Compute the accuracy of the classifier, considering a few test datasets.
5	Develop a program to implement the Naïve Bayes Classifier.

	Use the Olivetti Faces Dataset for training and compute the accuracy on a few test datasets.
6	Write a program to implement the Kernel SVM Algorithm for a sample dataset stored as a .csv file. Compute the accuracy of the classifier, considering a few test datasets, and visualize the model using Python.
7	Implement Ensemble Learning algorithms Random Forest, AdaBoost, and XGBoost for classification. Use a suitable dataset (e.g., Breast Cancer / Iris / Wine Quality). Compare the performance of all three models using evaluation metrics and visualization plots.
8	Develop a program to implement K-Means Clustering using the Wisconsin Breast Cancer Dataset. Visualize the clustering results and analyze cluster distribution.
9	Develop a program to implement Principal Component Analysis (PCA) for Dimensionality Reduction. Visualize the variance explained by principal components and the transformed data.
10	Write a program to implement the Hierarchical Agglomerative Clustering Algorithm. Use a sample .csv dataset, visualize the dendrogram, and interpret clustering results.
11	Build a Recommender System using Content-Based and Collaborative Filtering methods. Create a simple movie/user recommender system and visualize similarity matrices.
12	Implement an Artificial Neural Network (ANN) for Regression and Classification tasks.Build a Multilayer Perceptron using TensorFlow/Keras or Scikit-learn MLP, tune hyperparameters, and visualize learning curves.
	Perform Model Evaluation and Comparison for a real-world dataset stored as a .csv file. Train multiple models (e.g., Logistic Regression, SVM, Random Forest, XGBoost) and compare their performance metrics to determine the most suitable algorithm for the given problem.

Text Books:

1. Tom M. Mitchell, Machine Learning, McGraw-Hill Education, 1st Edition, 1997.
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
3. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020.
4. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

Reference Books:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
2. Sebastian Raschka and Vahid Mirjalili, Python Machine Learning, Packt Publishing, 3rd Edition, 2019.
3. Prateek Joshi, Artificial Intelligence with Python, Packt Publishing, 2017.
4. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2nd Edition, 2018.

e-sources: NPTEL/SWAYAM Course (if any)

1. Machine Learning, Prof. Balaraman Ravindran, IIT Madras, https://nptel.ac.in/courses/106/106/106106202/
2. Machine Learning Foundations Prof. Sudeshna Sarkar IIT Kharagpur https://nptel.ac.in/courses/106/105/106105215/
3. Deep Learning Prof. Mitesh Khapra, Prof. Balaraman Ravindran IIT Madras https://nptel.ac.in/courses/106/106/106106184/