AC –
Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 5		
Name of the Programme – B.E. Computer Science and Engineering (Artificial Intelligence & Machine Learning)		
Faculty of Engineering		
Board of Studies in Computer Engineering		
U.G. Third Year Programme	Exit Degree	B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)
Semester		V & VI
From the Academic Year		2026-27

Draft Copy

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.E. Computer Science and Engineering (Artificial Intelligence & Machine Learning)**
2	**Exit Degree**	**B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)**
3	**Scheme of Examination** **R: _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R: _____**	**Attached herewith**
6	**Semesters**	**Sem. V & VI**
7	**Program Academic Level**	**5.5**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2026-27**

Draft Copy

Sd/-	Sd/-	Sd/-
Dr. Subhash K. Shinde	Dr. Deven Shah	Prof. Shivram S. Garje
BoS-Chairman-Computer Engineering	Associate Dean	Dean
Faculty of Technology	Faculty of Science & Technology	Faculty of Science & Technology

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The third-year engineering course consists of a core and elective training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, program elective course, multidisciplinary minor, IKS, PCE and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program, elective program, communication skills and skills required for the Computer Engineering Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends. Elective courses are offered to the learners to understand the concepts of emerging technology. This will help with the holistic development of learners through interdisciplinary skills development and mobility among other departments as real-time solutions. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Computer Engineering in Engineering for open electives and multidisciplinary minor courses in sem V and VI. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course and Program Elective Courses Cover Computer Engineering core and elective courses. Also, MDM is where a pool of subjects is given for selection. In addition to this learner will have IKS and PCE courses. Considering the present scenario, diverse choices need to be made available to fulfil the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. The IKS course will be program-specific, which can be achieved in Undergraduate training by giving an objective viewpoint to the learning technology, from ascent time and how it was learned and the process and transitioning a learner from a rote learner to a creative professional. For this purpose, PCE is introduced in the First Semester to orient a communication skills to become skilled professionals for placement and higher education entrance exams. Considering the NEP-2020 structure of award of Certificate after U.G. First Year, U. G. Diploma after the Second Year and B.Sc. Tech Engineering after the Third Year at multiple exit-point pools of Vocational skills, is arranged to give exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every core and elective course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught to the learners, which will enhance the learners' learning process. NEP 2020 grading system enables a much-needed shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on the holistic development of the learners through mobility and continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020; however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, self learning, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the third-year syllabus must not be heavily loaded for the learner and it is of utmost importance that the learner entering into the third year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Third Year of Engineering from the academic year 2026-27. Subsequently, this system will be carried forward for Final Year Engineering in the academic years 2027-28, respectively.

Sd/-	**Sd/-**	**Sd/-**
Dr. Subhash K. Shinde	**Dr. Deven Shah**	**Prof. Shivram S. Garje**
BoS-Chairman-Computer Engineering	**Associate Dean**	**Dean**
Faculty of Technology	**Faculty of Science & Technology**	**Faculty of Science & Technology**

Draft Copy

B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)

	R: _____C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
5.5	V	PCC501:3 PCC502:3 PCC503:3 PCL501: 1 PCL502:1 PCL503:1	PEC501:3 PEL501:1	MDC:3 MDL:1	OE:2	--	IKS: 2	--	24	**B.Sc. Tech Engineering 48**
	R:_____D									
	VI	PCC601:3 PCC602:3 PCL601:1 PCL602:1	PEC601:3 PEC602:3 PEL601:1 PEL602:1	MDC: 3 MDL:1	--	VSEC:2	AEC: 2	--	24	
	Cum Cr.	20	12	8	2	2	*2+2*	--	48	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Draft Copy

T.E.
Scheme

Draft Copy

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER V

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2165111	Statistics for Machine Learning and Data Science	3	--	--	3	--	--	3
2165112	Artificial Intelligence and Soft Computing	3	–	--	3	--	--	3
2165113	Agile Software Development and DevOps	3	--	--	3	--	--	3
216511#	Program Electives -I	3	--	--	3	--	--	3
2165211	Multi-disciplinary Minor	3	--	--	3	--	--	3
2165117	Statistics for Machine Learning and Data Science Lab	--	2	--	--	--	1	1
2165118	Artificial Intelligence and Soft Computing Lab	--	2	--	--	--	1	1
2165119	Agile Software Development and DevOps Lab	--	2	--	--	--	1	1
216512#	Program Electives -I Lab	--	2	--	--	--	1	1
2165212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1
2165311	To be taken from the bucket provided by the University from other Faculty	2#	--	---	2	--	--	2
2165511	Indian Knowledge System (Syllabus common to all Branch).	--	2*+2	--	--	--	2	2
Total		**17**	**14**	**00**	**17**	**00**	**07**	**24**

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

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

@ Institute shall offer a course for MDM from other Engineering Boards Group.

Program Electives:	
Course Code	Course Description
2175114	Web and Mobile Application Development
2175115	User Experience Design with VR
2175116	Computer Network
Lab Code	Lab Description
2175120	Web and Mobile Application Development Lab
2175121	User Experience Design with VR Lab
2175122	Computer Network Lab

Draft Copy

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER V

Course Code	Course Description	Examination Scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2165111	Statistics for Machine Learning and Data Science	20	20	40	60	2	--	--	100
2165112	Artificial Intelligence and Soft Computing	20	20	40	60	2	--	--	100
2165113	Agile Software Development and DevOps	20	20	40	60	2	--	--	100
216511#	Program Electives -I	20	20	40	60	2	--	--	100
2165211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2165117	Statistics for Machine Learning and Data Science Lab	--	--	--	--	--	25	25	50
2165118	Artificial Intelligence and Soft Computing Lab	--	--	--	--	--	25	25	50
2165119	Agile Software Development and DevOps Lab	--	--	--	--	--	25	25	50
216512#	Program Electives -I Lab	--	--	--	--	--	25	-	25
2165212	Multi-disciplinary Minor Lab	--	--	--	--	--	25	25	50
2165311	To be taken from the bucket provided by the University from other Faculty	--	--	--	--	--	50	--	50
2165511	Indian Knowledge System (Syllabus common to all Branch).	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**100**	**825**

Program Electives:	
Course Code	Course Description
2165114	Web and Mobile Application Development
2165115	User Experience Design with VR
2165116	Computer Network
Lab Code	Lab Description
2165120	Web and Mobile Application Development Lab
2165121	User Experience Design with VR Lab
2165122	Computer Network Lab

Draft Copy

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER VI

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2166111	Machine Learning and Swarm Intelligence	3	--	--	3	--	--	3
2166112	Data Analytics and Visualization	3	--	--	3	--	--	3
216611#	Program Electives -II	3	--	--	3	--	--	3
216611#	Program Electives -III	3	--	--	3	--	--	3
2166211	Multi-disciplinary Minor	3	--	--	3	--	--	3
2166411	Mini Project-II	--	4	--	--	--	2	2
2166511	Professional Skill	--	2*+2	--	--	--	2	2
2166119	Machine Learning and Swarm Intelligence Lab	--	2	--	--	--	1	1
2166120	Data Analytics and Visualization Lab	--	2	--	--	--	1	1
216612#	Program Electives -II Lab	--	2	--	--	--	1	1
216612#	Program Electives -III Lab	--	2	--	--	--	1	1
2166212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1
Total		15	18	00	15	00	09	24

* Two hours of practical class to be conducted for full class as demo/discussion.
Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours
@ Institute shall offer a course for MDM from other Engineering Boards.

	Course Code	Course Description	Lab Code	Lab Description
Program Electives -II	2166113	Internet of Things	2166121	Internet of Things Lab
	2166114	Cyber Security	2166122	Cyber Security Lab
	2166115	Nature-Inspired Optimization Algorithms	2166123	Nature-Inspired Optimization Algorithms Lab
Program Electives -III	2166116	Edge Computing	2166124	Edge Computing Lab
	2166117	Digital Imaging Techniques and Analysis	2166125	Digital Imaging Techniques and Analysis Lab
	2166118	Blockchain Technology	2166126	Blockchain Technology Lab

Draft Copy

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER VI

Course Code	Course Description	Examination Scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2166111	Machine Learning and Swarm Intelligence	20	20	40	60	2	--	--	100
2166112	Data Analytics and Visualization	20	20	40	60	2	--	--	100
216611#	Program Electives -II	20	20	40	60	2	--	--	100
216611#	Program Electives -III	20	20	40	60	2	--	--	100
2166211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2166411	Mini Project-II	--	--	--	--	--	50	--	50
2166511	Professional Skill	--	--	--	--	--	25	--	25
2166119	Machine Learning and Swarm Intelligence Lab	--	--	--	--	--	25	25	50
2166120	Data Analytics and Visualization Lab	--	--	--	--	--	25	25	50
216612#	Program Electives -II Lab	--	--	--	--	--	25	25	50
216612#	Program Electives -III Lab	--	--	--	--	--	25	25	50
2166212	Multi-disciplinary Minor Lab	--	--	--	--	--	25	25	50
Total		**100**	**100**	**200**	**300**	**10**	**200**	**125**	**825**

	Course Code	Course Description	Lab Code	Lab Description
Program Electives -II	2166113	Internet of Things	2166121	Internet of Things Lab
	2166114	Cyber Security	2166122	Cyber Security Lab
	2166115	Nature-Inspired Optimization Algorithms	2166123	Nature-Inspired Optimization Algorithms Lab
Program Electives -III	2166116	Edge Computing	2166124	Edge Computing Lab
	2166117	Digital Imaging Techniques and Analysis	2166125	Digital Imaging Techniques and Analysis Lab
	2166118	Blockchain Technology	2166126	Blockchain Technology Lab

Draft Copy

Vertical – 1
Major

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165111	Statistics for Machine Learning and Data Science	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2165111	Statistics for Machine Learning and Data Science	20	20	40	60	2	--	--	100

Rationale: In modern data-driven systems, statistical understanding forms the backbone of Machine Learning and Data Science. This course integrates foundational statistical concepts with practical statistical learning methods such as regression, classification, resampling, unsupervised learning, Bayesian reasoning, and ethical AI practices. Students will learn to perform sound data analysis, build interpretable ML models, evaluate them rigorously, and understand the implications of data-driven decisions.

Course Objectives: By the end of the course, the learner will be able to:

1. Understand fundamental concepts of probability, descriptive statistics, and sampling techniques for data analysis.
2. Apply estimation and hypothesis testing methods, including classical and resampling approaches.
3. Build and interpret regression models (linear, nonlinear, and regularized) for predictive analytics.
4. Implement supervised and unsupervised statistical learning algorithms to model real-world data.
5. Evaluate models using cross-validation, MLE, and statistical performance metrics.
6. Analyze ethical implications, interpret model decisions, and apply Bayesian reasoning for explainable ML.

Draft Copy

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

1. Summarize and interpret datasets using descriptive measures, sampling and distributional analysis.
2. Perform statistical inference using estimation, hypothesis testing, and resampling approaches.
3. Build, evaluate, and interpret regression and regularized models.
4. Apply supervised and unsupervised learning algorithms effectively.
5. Evaluate models using appropriate metrics, Bayesian reasoning, and cross-validation.
6. Analyze fairness, model interpretability, and ethical implications in ML workflows.

Prerequisite: Basic Probability, Linear Algebra, Python Programming

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basic Probability, Linear Algebra, Python Programming	00	
I	**Statistical Foundations, Covariance, and Sampling**	Types of data: categorical, numerical, ordinal. Descriptive statistics: central tendency, dispersion, skewness, kurtosis. Data visualization: histograms, boxplots, scatter plots, pair plots. Probability distributions: Normal, Binomial, Poisson, Exponential, Uniform, Beta, Gamma. Covariance, correlation, confounding, correlation matrices. Distance measures: Euclidean, Manhattan, Cosine. Sampling techniques: simple random, stratified, cluster. Central Limit Theorem (CLT) and Law of Large Numbers. Data preprocessing: handling missing values, outlier treatment, encoding, normalization. **Self-learning Topics:** Explore "An Introduction to Statistical Learning (ISLR2)" YouTube videos and Kaggle datasets for descriptive analytics and sampling distribution exercises. https://onlinecourses.nptel.ac.in/noc24_ma03/preview https://onlinecourses.nptel.ac.in/noc24_ma64/preview?	07	CO1, CO6
II	**Statistical Inference, Estimation & Resampling**	Point estimates, properties of estimators: bias, variance, MSE. Maximum Likelihood Estimation (MLE) and Method of Moments. Confidence intervals (mean, proportion, variance). Hypothesis testing: z-test, t-test, chi-square, ANOVA. Nonparametric tests: Mann–Whitney U, Kruskal–Wallis. Resampling methods: bootstrap CI, permutation tests. Likelihood Ratio Tests (LRT). Multiple testing correction: Bonferroni & FDR. Introduction to Monte Carlo simulation for sampling distribution approximation. **Self-learning Topics:** Explore bootstrapping via Python tutorials. Read "Practical Statistics for Data Scientists", Ch. 5 on inference. https://onlinecourses.nptel.ac.in/noc25_ma55/preview?	06	CO2, CO5
III	**Regression Models, Regularization & Nonlinear Models**	Simple & multiple linear regression. Model assumptions, residual analysis, multicollinearity (VIF), leverage, Cook's distance. Polynomial regression and interaction effects. Nonlinear regression models. Regularization: Ridge, Lasso, **Elastic Net.** Spline regression, LOESS (conceptual). Generalized Linear Models (GLM) – logistic (binary), Poisson (count data). Model evaluation metrics: MAE, MSE, RMSE, R^2, Adjusted R^2. Subset selection: forward, backward	08	CO3, CO6

Draft Copy

		(overview). Model calibration and interpretability basics. **Self-learning Topics:** Practice polynomial/spline regression using scikit-learn. Use open datasets to implement regularization. https://onlinecourses.nptel.ac.in/noc24_cs20/preview		
IV	**Classification & Supervised Learning**	Classification metrics: accuracy, precision, recall, F1, ROC–AUC, PR curves. Logistic regression, odds ratio, decision boundary. Naïve Bayes (Frequentist and Bayesian views). k-NN classifier. Decision Trees & pruning. Ensemble methods: Bagging, Random Forest, Gradient Boosting (overview). Support Vector Machines (SVM): linear & kernel trick– (conceptual). Handling imbalanced data: SMOTE, class weights, stratified splits. Confusion matrix analysis & error trade-offs. Model calibration: Platt scaling & isotonic regression (overview). **Self-learning Topics:** Implement logistic and k-NN in Python. Explore metric trade-offs using real datasets.	07	CO4, CO6
V	**Unsupervised Learning & Dimensionality Reduction**	PCA: covariance matrix, Eigen decomposition, explained variance. LDA (supervised dimensionality reduction). Kernel PCA & ICA (conceptual overview). Clustering: K-means, hierarchical, DBSCAN. Cluster evaluation: silhouette score, Davies–Bouldin index. Dimensionality reduction for visualization: t-SNE, UMAP (conceptual). Outlier detection methods: Z-score, IQR, Isolation Forest (intro), Local Outlier Factor (LOF) – conceptual overview. **Self-learning Topics:** Visualize PCA/t-SNE clusters. Compare clustering methods with real-world datasets.	05	CO4, CO6
VI	**Model Evaluation, Bayesian Inference & Ethics**	Model selection: AIC, BIC.Bias–variance trade-off. Cross-validation: k-fold, LOOCV, repeated CV. Calibration curves and Brier score (conceptual). Bayesian fundamentals: prior, likelihood, posterior. MAP vs MLE, conjugate priors (Beta–Bernoulli, Normal–Normal). Naïve Bayes from Bayesian perspective. Interpretability tools (conceptual): SHAP, LIME, PDP, ICE. Fairness in ML: equalized odds, demographic parity. Ethical considerations: model bias, transparency, data leakage, Data Privacy Overview: anonymization, differential privacy. Reproducible ML: dataset versioning, MLflow / Weights & Biases (W&B conceptual). **Self-learning Topics:** Read beginner-friendly blogs on Bayesian stats. Interpretability tools: SHAP, LIME (high-level); permutation importance, PDPs, ICE plots.	06	CO5, CO6

Text Books:

Draft Copy

1. Aggarwal, Charu C. *Probability and Statistics for Machine Learning*. Springer, 2024.
2. James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert. *An Introduction to Statistical Learning: with Applications in Python* (2nd ed.). Springer, 2023.
3. Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
4. *Practical Statistics for Data Scientists — Peter Bruce et al., 2020*

References:

1. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer, 2009.
2. DasGupta, Anirban. *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*. Springer, 2011.
3. Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
4. McElreath, Richard. Statistical Rethinking: *A Bayesian Course with Examples in R and Stan*. CRC Press, 2020 (2nd ed.).
5. *Before Machine Learning, Volume 3: Probability & Statistics for AI*. Packt Publishing,

Online References: Students are encouraged to use the following verified online learning platforms for additional learning and self-study.

Sr. No.	Website Name
1.	https://www.statlearning.com/
2.	https://scikit-learn.org/stable/
3.	https://developers.google.com/machine-learning/crash-course
4.	https://distill.pub/2016/misread-tsne/
5.	https://christophm.github.io/interpretable-ml-book/
6	https://onlinecourses.nptel.ac.in/noc25_ma20/preview?
7.	https://onlinecourses.nptel.ac.in/noc25_ma21/preview?

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3,4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165112	Artificial Intelligence & Soft Computing	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165112	Artificial Intelligence & Soft Computing	20	20	40	60	2	--	--	100

Course Objectives: The course aims to:

1. To introduce the fundamental concepts, evolution, and significance of Artificial Intelligence and Soft Computing, highlighting their roles in solving real-world problems.
2. To enable students to understand and apply classical AI search strategies such as uninformed, informed, and local search for goal-driven problem solving.
3. To develop the ability to represent and reason with knowledge using logical and probabilistic frameworks for handling uncertainty in intelligent systems.
4. To familiarize learners with fuzzy set theory, fuzzy rules, and inference mechanisms for approximate reasoning and intelligent decision-making.
5. To impart knowledge of neural network architectures and learning algorithms for modelling cognitive and pattern-recognition processes.
6. To integrate fuzzy logic and neural networks into hybrid intelligent systems and analyze their comparative performance across applications.

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

1. Describe the core concepts, constituents, and applications of Artificial Intelligence and Soft Computing.
2. Apply various uninformed, informed, and local search techniques to solve real-world and optimization problems.
3. Represent and reason with knowledge using propositional and predicate logic, and manage uncertainty using probabilistic reasoning.
4. Develop and analyze fuzzy inference systems for decision-making and control applications.
5. Design, train, and evaluate neural network models using supervised and unsupervised learning algorithms.
6. Integrate fuzzy and neural models into hybrid intelligent systems such as ANFIS and evaluate their performance.

DETAILED SYLLABUS:

Draft Copy

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Mathematical Foundations: Linear Algebra, Programming Skills: Python (preferred), Basic understanding of data structures, Discrete Mathematics	--	--
I	**Introduction to Artificial Intelligence & Soft Computing**	Introduction to Artificial Intelligence: Artificial Intelligence (AI), AI Perspectives, Applications of AI, The present state of AI Introduction to Soft Computing: Soft computing Constituents, Characteristics of Neuro Computing and Soft Computing, Difference between Hard Computing and Soft Computing, Concepts of Learning and Adaptation. **Self-learning Topics:** Recent advances in Generative AI, ChatGPT-like models, ethical challenges in AI.	05	CO1
II	**Solving Problems by Searching**	Definition, State space representation, Problem as a state space search, Problem formulation, Uninformed Search: Depth First Search, Breadth First Search, Informed Search: Greedy Best First Search, A* Search. Local Search: Hill Climbing Search, Simulated Annealing Search, **Self-learning Topics:** Game Playing, Adversarial Search Techniques	07	CO2
III	**Knowledge and Reasoning**	Knowledge Representation Systems, Propositional Logic (PL): Syntax, Semantics, Formal logic-connectives, Predicate Logic: FOPL, Syntax, Semantics, Quantification, Inference rules in FOPL, Forward Chaining, Backward Chaining and Resolution in FOPL. Handling Uncertain Knowledge, Random Variables, Prior and Posterior Probability **Self-learning Topics:** Ontologies and Semantic Web reasoning	07	CO3
IV	**Fuzzy Set Theory & Fuzzy Rules, Reasoning and Inference System**	Fuzzy Set Theory: Fuzzy Sets, Fuzzy relations, Fuzzification and Defuzzification. Features of the membership Functions Fuzzy Rules, Reasoning and Inference System: Fuzzy Rules: Fuzzy If-Then Rules, Fuzzy Reasoning Fuzzy Inference System (FIS): Mamdani FIS, Sugeno FIS, Comparison between, Mamdani and Sugeno FIS **Self-learning Topics:** Adaptive fuzzy systems and membership optimization	07	CO4
V	**Neural Networks**	Basics of Neural Networks: Introduction to Neural Networks, Biological Neural Networks, McCulloch Pitt model Supervised Learning algorithms: Perceptron (Single Layer, Multi-layer), Linear separability, Delta learning rule, Back Propagation algorithm Un-Supervised Learning algorithms: Hebbian Learning, Winner take all, Self Organizing Maps, Learning Vector Quantization. **Self-learning Topics:** CNNs and RNNs — introductory overview and comparison	08	CO5
VI	**Hybrid System**	Introduction to Hybrid Systems, Key applications of hybrid systems (industrial control, robotics, forecasting), Adaptive Neuro-Fuzzy Inference System	05	CO6

Draft Copy

		(ANFIS) , Comparative evaluation of Fuzzy, Neural, and Neuro-Fuzzy models. **Self-learning Topics:** Sustainability and governance in hybrid AI systems.		

Text Books:

1. Stuart Russell and Peter Norvig**,** *Artificial Intelligence – A Modern Approach*, 4th Edition, Pearson Education, 2021.
2. S. Rajaraman and G. Vijayalakshmi Pai**,** *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications,* PHI Learning, 2017.
3. S. N. Sivanandam and S. N. Deepa**,** *Principles of Soft Computing, 3rd Edition*, Wiley India, 2018.

References:

1. Elaine Rich, Kevin Knight, and Shivashankar B. Nair**,** *Artificial Intelligence,* 3rd Edition, McGraw-Hill Education, 2017
2. Simon Haykin**,** *Neural Networks and Learning Machines,* 3rd Edition, Pearson, 2017.
3. Bart Kosko**,** *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence,* Prentice Hall, 1992.

Online References:

Sr. No.	Website Name
3.	https://onlinecourses.nptel.ac.in/noc25_cs48/preview?
4.	https://nptel.ac.in/courses/106105077
3.	https://stanford-cs221.github.io/spring2024/

Assessment:
Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165113	Agile Software Development & DevOps	3	-	-	3	-	-	3

Course Code	*Course Name*	*Theory*					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165113	Agile Software Development & DevOps	20	20	40	60	2	--	--	100

Course Objectives:

1. Understand software engineering frameworks, SDLC models, and the need for Agile software development.
2. Recognize different Agile development models, their methods, values, roles, artifacts, and business benefits.
3. Understand Agile requirements elicitation, modeling, prioritization, and Agile design principles.
4. Learn Agile project planning, estimation techniques, forecasting, and risk management.
5. Understand Agile approaches to software quality, testing principles, automation, and continuous testing.
6. Understand DevOps principles, tools, lifecycle, and integration with Agile for business agility.

Course Outcomes: After learning this course, students will be able to

1. Describe SDLC process models, differentiate traditional vs. Agile models, and summarize Agile principles and stakeholder roles.
2. Compare Agile development models (Scrum, XP, Kanban, FDD, Lean, Crystal, ASD), and apply Agile values and tools to practical scenarios.
3. Develop and prioritize Agile requirements using product and sprint backlogs, user stories, story cards, and apply Agile design principles.
4. Estimate, plan, and track Agile projects using techniques such as Planning Poker, story points, velocity, burndown charts, and forecasting methods.
5. Implement Agile testing strategies including TDD, BDD, acceptance and exploratory testing, and analyze quality metrics using automation tools.
6. Describe DevOps principles and lifecycle, implement CI/CD pipelines using DevOps tools, and evaluate the impact on software delivery and team collaboration.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basics of Programming and concepts in Software development.		
I	**Introduction to Agile Software Development**	Software Engineering-process framework, Software Development Life Cycle (SDLC) Process Models: The Linear Sequential Model, The Prototyping Model, The RAD Model,	6	CO 1

Draft Copy

		Evolutionary Process Models, Agile Process Model, Component-Based Development. Agile Process model, Need of Agile software development, Agile Manifesto and Principles, Stakeholders and Challenges, History of Agile Software Development, Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Teams, Team Meetings and Ceremonies, Agile Drivers, Capabilities and Values. **Self-learning Topics:** explore advanced Agile concepts such as Agile scaling frameworks (SAFe, LeSS), Agile estimation techniques, and metrics like velocity and burndown charts.		
II	**Agile Processes**	Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, Adaptive Software Development, Test Driven Development, and Lean Software Development, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility. **Self-learning Topics:** To Study Agile metrics like cycle time, lead time, and cumulative flow diagrams along with tools such as Jira, Azure DevOps, and Trello for practical implementation.	8	CO 2
III	**Agile Requirements Engineering & Design**	Impact of Agile Processes in RE, Product Backlog, Sprint Backlogs, Current Agile Practices, Variance, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Prioritization, Agile Requirements Modeling and Generation, Concurrency in Agile Requirements Generation, User Stories – Specification and Formats, Story-Cards. Agile Design: Symptoms of poor design, Agile design principles **Self-learning Topics:** Explore advanced backlog management techniques such as user story mapping, impact mapping, and MoSCoW prioritization to deepen their understanding of Agile requirements.	7	CO 3
IV	**Agile Planning & Estimation**	Planning levels; User stories and estimation; Planning Poker; Velocity and Burndown Charts; Agile risk management. Agile forecasting and project Management, velocity, progress tracking, Track Done pattern, Project forecasting, Estimation: Level of Effort, Ideal Time, Hours, Story Points, Team Participation, Sprint Planning and Estimation. **Self-learning Topics:** Advanced Agile forecasting techniques such as Monte Carlo simulations, probabilistic planning, and	6	CO 4

Draft Copy

		throughput-based forecasting for more accurate predictions.		
V	**Agile Quality Assurance & Testing**	Agile Metrics, Agile Approach to Quality Assurance. Testing: Functionality Testing, UI Testing, Performance Testing, Security Testing, Agile Testing: Principles of Agile Testers; The Agile Testing Quadrants, Test Driven Development, Acceptance Tests, Refactoring, Agile Automation, Test Automation Pyramid. Case Study – Selenium/ Jira **Self-learning Topics:** Continuous Testing, BDD, exploratory testing, and modern automation tools such as Selenium, Cypress, Postman, and JMeter.	6	CO 5
VI	**DevOps**	DevOps: Introduction, DevOps - Importance and Benefits, DevOps Principles and Practices, 7-C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, Choosing the right DevOps tools and challenges in DevOps Implementation. Mapping App to DevOps-Assessment. Study of various Open Source Tools in DevOps. **Self-learning Topics:** CI/CD pipelines, cloud-based DevOps tools, and real-world case studies on DevOps adoption and automation practices.	6	CO 6

Text Books:

1. Roger Pressman, Software Engineering: A Practitioner's Approach, 6th edition, McGraw Hill International Publication.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Introduction to Agile Methods, Sondra Ashmore
4. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.
5. Mike Cohn – Agile Estimating and Planning (Pearson)

Reference Books:

1. David J. Anderson and Eli Schragenheim, ―Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results‖, Prentice Hall, 2003.
2. Craig Larman, Agile and Iterative Development: A Manager's Guide, Addison-Wesley, 2004.
3. Sricharan Vadapalli , DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive, , Packt, 2018.

Online References:

Sr. No.	Website Name
1.	https://www.edx.org/course/agile-software-development
2.	https://www.coursera.org/learn/agile-software-development
3.	https://devops.com/most-popular-open-source-devops-tools/
4.	https://github.com/jidibinlin/Free-DevOps-Books-1/blob/master/book/Practical%20DevOps.pdf

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Draft Copy

- **Question paper format:**
- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1)Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Draft Copy

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

.....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical & Oral	Tutorial	Total
2165114	Web and Mobile Application Development	03	–	--	03	–	--	03

Course Code	Course Name	Examination Scheme							
		Theory Marks							
		Internal assessment			End Sem. Exam	Time Duration (in Hrs.)	Term Work	Practical/ Oral	Total
		IA-1	IA-2	IA-1 + IA-2					
2165114	Web and Mobile Application Development	20	20	40	60	02	–	–	100

Rational

This course provides a practical foundation in modern web and mobile development using React, Kotlin, and Flutter. It equips students to design, build, and deploy scalable, cross-platform applications while encouraging exploration of emerging technologies like PWAs, AI/ML APIs, and cloud-based apps.

Course Objectives

1. Understand the core concepts of modern web and mobile application architectures, including REST APIs and development stacks.
2. Develop interactive and efficient front-end interfaces using React and state management tools.
3. Explore Kotlin programming fundamentals and apply object-oriented and functional programming concepts for robust app logic.
4. Build scalable and secure backend services using Kotlin with frameworks
5. Learn the foundations of Flutter and Dart to create cross-platform mobile applications with effective state management.
6. Integrate advanced UI designs, APIs, and Firebase services into Flutter applications for real-world functionality.

Course Outcomes

1. Explain the architectural patterns, design principles, and technology stacks used in modern web and mobile development.
2. Implement front-end applications using React components, routing, hooks, and data management techniques.
3. Apply Kotlin programming constructs, collections, and OOP principles to develop efficient application logic.
4. Design and deploy backend systems in Kotlin using micro services, authentication, and database integration.
5. Create cross-platform mobile apps using Flutter and Dart, incorporating layouts, gestures, and state management.
6. Develop advanced mobile solutions by integrating APIs, Firebase authentication, and real-time database.

Draft Copy

Detailed Syllabus:

Sr. No..	Module Title	Contents of the module	Hours	CO
0	**Pre-Requisite Topics**	Basic Concepts: HTML5, CSS3; JavaScript, Client-Server paradigm, Front-end and Back-end development.		–
I	**Core Web and Mobile Development Concepts**	API architectures such as REST APIs. Architectural principles and styles, Design and Architectural patterns (MVC, MVVM). Modern development stacks for Web and mobile (MERN, MEAN, PERN). **Self-Learning:** Overview of Progressive Web Apps (PWAs),	3	CO1
II	**Front-End Development with React**	Foundations of React, JSX, all about Components, React Devtools and Data Flow, Events, Form, Refs, Styling with React, Hooks, rouring, error boundaries, deploying React, initialing projects, Fetching and Caching Data, react portals and accessibility, Redux. **Self-learning Toics:** Performance profiling using React DevTools, Custom Hooks creation	7	CO2
III	**Kotlin Programming Core Concepts**	Basics of kotlin, variables, data types, conditional statements, Functions: structure, calling single-expression functions, Unit, Anonymous functions, standard functions. Null Safety and Exception handling. Collections: List, Set, Map, etc., Collection operations: map, filter, forEach, reduce, fold. Classes: Initialization, Inheritance, Object keyword, nested classes, data classes, enumerated classes, operator overloading, interfaces and abstract classes. Generics and Extensions, Java Interoperability. **Self-learning Topics:** Jamstack architecture, Webassembly, Low-code/No-code development	8	CO3
IV	**Kotlin Backend Development and Advanced Concepts**	Building scalable services with Spring boot, Database integration and ORM, Authentication and Authorization: OAuth2, JWT, session-based authentication, role-based access. Building microservices with Kotlin, Deploying Kotlin backend applications, Testing Monitoring and Scaling. Annotations and Reflection, Domain Specific Languages, Concurrency: Coroutines, coroutine control-flow, concurrency communication. Testing with Kotlin: Assertions, Fixtures and configuration., REST APIs with Ktor, Introduction to Jetpack Composer and Kotlin Multiplatform Mobile. **Self-learning Topics:** API Gateway concept, Dockerizing Kotlin backend applications	8	CO4
V	**Flutter Foundations and Dart Programming**	Introduction to Flutter: Features, Advantages and Disadvantages, Installation (Windows & Mac OS), Creating Simple Application in Android Studio, Architecture of Flutter Applications Flutter Basics: Widgets, Gestures, Introduction to Dart Programming, Variables and Data Types,	7	CO5

Draft Copy

		Decision Making, Loops, and Functions, OOP Concepts, Widget Build Visualization Dart Programming Essentials: Basic Programming in Dart, Exception Handling, Debugging, Asynchronous Programming (Futures, Async, Await, Streams) Layouts and State Management: Types of Layout Widgets, Single Child and Multiple Child Widgets, Advanced Layout Applications, Introduction to Gestures, State Management in Flutter, Ephemeral and Application State, Scoped Model, Navigation, and Routing. **Self-learning Topics:** Exploring the internals of **Flutter** framework, Mixins and advanced OOP.		
VI	**Advanced UI, API Integration, and Firebase: Flutter**	1. UI Development: Styles, Assets, and Fonts, Model, API, and Media Query, Working with Lists and Grids, Implementing Animations, Customizing UI and Drawing/Painting, Using Flutter Packages to Enhance UI 2. API and Package Development: Introduction to API and Packages, Types of Packages and Using Dart Packages, developing a Flutter Plugin Package, Accessing REST API, Accessing Product Service API 3. Flutter and Firebase: Introduction to Firebase and Its Features, Integrating Firebase with Flutter Apps, Implementing User Authentication with Firebase, Using Firestore for Real-time Database **Self-learning Topics:** AI/ML API integration, Cloud-based mobile apps, Mobile E-commerce	6	CO6

Text Books

1. Learning React: Modern Patterns for Developing React Apps – Alex Banks & Eve Porcello, O'Reilly (2023).
2. Beginning ReactJS Foundations: Building user interfaces with ReactJS - Chris Minnick, Wrox publication.
3. Kotlin Programming: The Big Nerd Ranch Guide by Josh Skeen and David Greenhalgh.
4. Kotlin for Backend development: Build scalable REST APIs, Microservices and Web Applications with Ktor, Spring Boot and Coroutines, Stokes J. Harrett.
5. Kotlin In-Depth: A Guide to a Multipurpose Programming Language for Server-Side, Front-End, Android, and Multiplatform Mobile - Aleksei Sedunov, BPB Publications.
6. Simplifying Application Development with Kotlin Multiplatform Mobile - Robert Nagy Packt publication
7. Beginning App Development with Flutter Create Cross-Platform Mobile Apps — Rap Payne, Apress
8. Flutter Cookbook, by Simone Alessandria, Brian Kayfitz, O'Reilly

Reference Books

1. Full-Stack Web Development with Node, Express, and React – Adam Freeman, Apress (2022)
2. Head First Kotlin - Dawn Griffiths & David Griffiths, O'Reilly.
3. Kotlin Development Complete Guide create 45 android applications including Jetpack Composer – Henry Codwell.
4. Adam Bretz and Colin J. Ihrig, "Full Stack JavaScript Development with MEAN", SitePoint Pty. Ltd., 2015 Edition.
5. Android UI development with Jetpack Compose - Thomas Kunneth, Packt Publishing.
6. Scalable Android applications in Kotlin.

Draft Copy

7. E-Commerce, Pondicherry university

Online Resources:

1. https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012
2. https://www.coursera.org/learn/advanced-programming-in-kotlin
3. https://www.coursera.org/learn/advance-kotlin-for-multi-platform-development
4. https://www.pluralsight.com/paths/android-development-with-kotlin-fundamentals
5. https://github.com/androiddevnotes/awesome-android-learning-resources
6. https://www.udemy.com/course/android-app-development-with-kotlin-beginner-to-advanced
7. https://www.codecademy.com/learn/learn-kotlin
8. https://www.codingblocks.com/android-development-using-kotlin.html
9. https://www.geeksforgeeks.org/flutter/flutter-tutorial/
10. https://www.youtube.com/playlist?list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165115	User Experience Design with VR	3	-		3	-		3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165115	User Experience Design with VR	20	20	40	60	2	-	--	100

Rationale:
The User Experience Design with VR, AI, and DS syllabus provides a forward-looking framework for developing creative, analytical, and ethical designers who can shape the future of human-technology interaction. By merging immersive technologies with data-driven intelligence, the course empowers learners to design experiences that are adaptive, meaningful, and human-centered.

Course Objectives:

1. To understand the fundamentals of interface design and core elements of user experience.
2. To learn the stages of the UX design life cycle and their importance in building user-centered systems.
3. To explore the UX design process, including information architecture, prototyping, and testing for web and mobile applications.
4. To study methods for evaluating and improving user experience using usability and heuristic techniques.
5. To understand the basic principles, components, and working of Virtual Reality systems.
6. To explore various applications of Virtual Reality and analyze the human factors involved in immersive environments.

Course Outcomes On successful completion, of course, learner/student will be able to:

1. Explain the basic concepts of interface design and elements of user experience.
2. Describe the complete UX design life cycle and its relevance in real-world projects.
3. Apply the UX design process to create and test interactive prototypes for web and mobile systems.
4. Evaluate user experience using different usability and heuristic evaluation techniques.
5. Illustrate the core principles and components of Virtual Reality systems.
6. Analyze various Virtual Reality applications and interpret human performance aspects in VR environments.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Computer Graphics	-	-
I	Introduction	Introduction to interface design, Understanding and conceptualizing Interface, understanding user's conceptual cognition, Core Elements of User Experience, Working of UX elements. **Self-learning Topics:** Norman's principles of design by Don Norman, Human perception and	04	CO1

Draft Copy

		visual processing		
II	**The UX Design - Life Cycle**	What is UX, Ubiquitous interaction, Emerging desire for usability, From usability to user experience, Emotional impact as part of the user experience, User experience needs a business case, Roots of usability. Introduction, A UX process lifecycle template, Choosing a process instance for your project, The system complexity space, Meet the user interface team, Scope of UX presence within the team, More about UX life cycle. **Self-learning Topics**: Historical evolution of UX, UX in IoT and smart device.	07	**CO2**
III	**The UX Design Process**	Introduction, The system concept statement, User work activity gathering, Look for emotional aspects of work practice, Abridged contextual inquiry process, Data-driven vs. model driven inquiry, Contextual Analysis, Extracting Interaction Design Requirements, Constructing Design Information Models. Information ,Architecture and Interaction Design and Prototyping Introduction, Design paradigms, Design thinking, Design perspectives, User personas, Ideation, Sketching, More about phenomenology, Mental Models and Conceptual Design, Wireframe, Prototyping Web UX Design Principles, Mobile UX Design Principles, Prototyping and Testing for Web/Mobile. **Self-learning Topics:** Translating user work activity into design requirements, Use case modeling for interaction design.	10	**CO3**
IV	**The UX Evolution and Improvement**	UX Evaluation and Improve UX Goals, Metrics and Targets, UX Evaluation Techniques.- Formative vs summative ,types of formative and informal summative evaluation methods, types of evaluation data, some data collection techniques Rapid Evaluation Methods: Design walkthroughs and reviews UX Inspection, Heuristic evaluation, a UX inspection method, practical approach to UX Inspection, Do UX Evaluation rite, Quasi-empirical UX evaluation Questionnaires, Specialized rapid UX evaluation methods. **Self-learning Topics:** Iterative UX improvement lifecycle, Role of evaluation in Agile & design thinking, UX maturity models.	06	**CO4**
V	**Introduction to Virtual Reality**	Defining Virtual Reality, The three I's of Virtual Reality, History, The five classic components of a VR system, Input Devices: Trackers, Navigation and Gesture Interfaces, Output Devices: Graphics, Three dimensional sound and Haptic displays. **Self-learning Topics:** Interaction design in immersive environments, Imagination and cognitive engagement in VR, Gesture recognition	06	**CO5**

Draft Copy

		concepts.		
VI	**Applications Virtual Reality**	Modeling: Geometric modeling, Kinematics modeling, Physical Modeling, Behavior modeling, Model management. Human factor in VR: Methodology and terminology, User performance studies. Traditional VR Applications, Emerging Applications of VR. **Self-learning Topics:** Collision detection algorithms, Real-time rendering optimization, Forward and inverse kinematics concepts.	06	**CO6**

Textbooks:

1. The UX Book by Rex Hartson and PardhaPyla, MK Publication Smashing UX Design by Jesmond Allen and James Chudley, John Wiley & Sons
2. LaValle, S. M. (2017). Virtual Reality. Cambridge University Press.
3. Alan Dix, Janet Finlay, Gregory Abowd, Russel Beale, "Human–Computer Interaction", Pearson, 2009.
4. Rogers Sharp Preece, "Interaction Design: Beyond Human Computer Interaction", 5th Edition, Wiley, 2019.
5. Kalbande Dhananjay R, Kanade P., Iyer S., Galitz's, "Human Machine Interaction", Wiley Publications, 2015.

Reference Books:

1. Jeff Johnson, "Designing with the mind in mind", 2nd Edition, Morgan Kaufmann Publication, 2014.
2. Brian Fling, "Mobile Design and Development", 1 st Edition, O'Reilly Media Inc., 2009.
3. Wilbert O. Galitz, "The Essential Guide to User Interface Design", Wiley publication, 2002.

Online References:

Sr. No.	Website Name
1.	https://archive.nptel.ac.in/courses/124/107/124107008/
2.	https://nptel.ac.in/courses/106106138
3.	https://www.coursera.org/specializations/virtual-reality
4.	https://msl.cs.uiuc.edu/vr/vrbook.pdf

Draft Copy

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165116	Computer Network	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165116	Computer Network	20	20	40	60	2	-	--	100

Rationale:
This course provides foundational knowledge of computer networks, covering architectures, topologies, transmission media, and devices. It develops skills in error control, IP addressing, routing, and socket programming while introducing advanced concepts like VPN, SDN, NFV, and data center networks to prepare students for modern networking applications.

Course Objectives:
1. Understand the fundamentals of computer networks, their types, architectures, topologies, and communication models.
2. Explain the design issues, error control, and media access mechanisms used in the Data Link Layer.
3. Analyze addressing, routing techniques, and protocols implemented in the Network Layer for data delivery.
4. Describe transport layer mechanisms including connection establishment, flow control, and congestion management.
5. Illustrate the functionality of common application layer protocols and their role in network communication.
6. Explore emerging networking technologies.

Course Outcomes: On successful completion of course, learner will be able to
1. Explain the basic concepts of computer networks, their architectures, topologies, transmission media, and network devices.
2. Demonstrate understanding of Data Link Layer design issues and implement error detection, correction, and flow control mechanisms.
3. Apply IPv4 and IPv6 addressing, subnetting, and routing algorithms for efficient packet delivery across networks.
4. Describe the working of transport layer protocols and implement basic socket programming for process-to-process communication.
5. Analyze the structure and operation of major application layer protocols such as HTTP, DNS, FTP, and SMTP.
6. Evaluate and discuss the concepts of VPN, SDN, NFV, and Data Center Networks as part of advanced networking technologies.

Prerequisite: Operating System Fundamentals

Draft Copy

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction To Computer Networks**	Definition, **Types of Networks**: Local area networks (LAN), Metropolitan area networks (MAN), Wide area networks (WAN), Wireless networks, Networks Software, Protocol, Design issues for the Network layers. **Network Models**: The OSI Reference Model, TCP/IP Model, Network Topologies, Types of Transmission Medium. **Network Architectures**: Client-Server, Peer to Peer, Hybrid. **Network Devices**: Bridge, Switch, Router, Gateway, Access Point. **Self-learning Topics:** Study of Campus wide networking	**6**	CO1
II	**Data Link Layer**	Introduction, Design Issues (Services to Network Layer, Framing. Error control, Flow control), Error Detection and correction, Parity Bits, Checksum, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocols. Elementary Data Link protocols MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, WDMA, Introduction to IEEE standards (IEEE 802.3, IEEE 802.11). Self-learning Topics: Demonstration of DLL protocols on Simulator	**8**	CO2
III	**Network Layer**	Network Layer design issues, **Switching Techniques**: Circuit switching, Message Switching, Packet Switching. **IP Protocol**: Classes of IP (Network addressing), IPv4 , IPv6, Network Address Translation, Sub-netting, Supernetting, CIDR. **Network layer Protocols**: ARP, RARP, ICMP, IGMP. **Network Routing and Algorithms**: Static Routing, Dynamic Routing, Distance Vector Routing, Link State Routing. **Routing Protocols**: RIP, OSPF, BGP **Self-learning Topics:** Count-to-Infinity Problem, AODV AND DSR routing protocol	**9**	CO 3
IV	**Transport Layer**	Process to Process Delivery, Services, Socket Programming. **Elements of Transport Layer Protocols**: Addressing, Connection establishment, Connection release, CP timers and TCP state transition diagram, Flow control and buffering, Multiplexing, Congestion Control: slow start protocol. **Transport Layer Protocols**: TCP and UDP, Quality of Service (QoS). **Self-learning Topics:** Demonstration of Transport layer protocols on Simulator.	**07**	CO4

Draft Copy

V	**Application Layer**	Introduction, Web and HTTP, Web Caching, DNS, Email: SMTP, POP3, Webmail, FTP, TELNET, DHCP, SNMP. **Self-learning Topics:** GOOGLE DNS	**04**	CO5
VI	**Emerging & Advanced Topics**	VPN (Virtual Private Networks) - Types and basic principle. **Software-Defined Networking (SDN):** Concepts, Control and Data Planes, OpenFlow protocols and controllers. **Network Function Virtualization (NFV):** Concepts and Benefits. **Data Center Networks (DCNs):** Topologies (Fat Tree - Concept) **Self-learning Topics:** SD-WAN	**05**	CO6

Text Books:

1. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, Publications, ISBN:0–07 – 058408 – 7
2. Andrew S. Tanenbaum, Computer Networks, 5th Edition, Pearson India, 2012.
3. A. S. Godbole and A. Kahate, Data Communications & Networks, 2nd ed. New Delhi: McGraw Hill Education, 2011.

References:

1. Kurose, Ross, "Computer Networking a Top Down Approach Featuring the Internet", Pearson, ISBN-10: 0132856204
2. L. Peterson and B. Davie, "Computer Networks: A Systems Approach", 5th Edition, Morgan-Kaufmann, 2012.
3. Douglas E. Comer & M.S Narayanan, "Computer Network & Internet", Pearson Education

Online References:

Sr. No.	Website Name
1.	NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/
2.	CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165117	Statistics for Machine Learning and Data Science Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165117	Statistics for Machine Learning and Data Science Lab	--	--	--	--	25	25	50

Rationale- The laboratory component reinforces theoretical understanding through hands-on data experiments using real-world datasets and tools such as Python (NumPy, Pandas, Scikit-learn). Students gain skills in exploratory analysis, sampling, hypothesis testing, regression, clustering, and model evaluation. It encourages experiential learning, data storytelling, and critical interpretation, building confidence to apply statistical learning methods for decision-making and AI development.

Lab Objectives: After completing the lab, students will be able to:

1. Apply descriptive statistics and sampling to analyze and visualize data.
2. Perform estimation, hypothesis testing, and resampling-based inference.
3. Implement regression and classification models and interpret diagnostic results
4. Apply unsupervised learning methods including PCA and clustering.
5. Evaluate ML models using appropriate metrics and cross-validation.
6. Interpret models with transparency, fairness, and ethical considerations.

Lab Outcomes: Upon successful completion, the student will be able to:

1. Perform EDA and sampling-based data exploration.
2. Conduct hypothesis testing and resampling analysis.
3. Implement regression and classification models.
4. Apply clustering and PCA for unsupervised analysis.
5. Evaluate statistical learning models using performance metrics.
6. Ensure interpretability and ethical awareness in statistical modeling.

Prerequisite: Python

Draft Copy

Suggested List of Experiments (10-12 Experiments must be conducted)

Sr No	List of Experiments	Hrs
01	Exploratory Data Analysis: descriptive stats, histograms, correlation matrix	02
02	Data preprocessing: missing values, encoding, scaling, outlier detection.	02
03	Sampling methods: SRS, stratified, cluster sampling + dataset demonstration	02
04	Demonstration of Central Limit Theorem using simulations.	02
05	Hypothesis testing (t-test, chi-square, ANOVA) on real dataset.	02
06	Bootstrap confidence intervals & permutation tests.	02
07	MLE estimation for Normal & Bernoulli distributions.	02
08	Linear regression & diagnostics (VIF, residual plots, Cook's distance)	02
09	Ridge, Lasso, Elastic Net with hyperparameter tuning.	02
10	Classification models: Logistic regression, KNN, Naïve Bayes, ROC analysis.	02
11	Decision Trees & Random Forest; feature importance.	02
12	PCA + clustering (K-means, hierarchical)	02
13	Interpretability: SHAP/LIME/PDP/ICE (conceptual + coding)	02
14	Fairness & bias detection metrics (equalized odds, parity)	02

Sr No	List of Assignments / Tutorials	Hrs
01	Mini project on complete ML pipeline (EDA → model → evaluation)	02
02	Comparison of resampling methods (bootstrap vs. permutation)	02
03	Regularization comparison (Ridge / Lasso / ElasticNet) on multiple datasets	02
04	Bias–variance tradeoff study	02
05	Ethical ML case study (fairness, interpretability, transparency)	02

Students must use at least 3 real-world datasets from Kaggle/UCI/OpenML
Common recommended datasets:

- Iris
- Titanic
- Breast Cancer Wisconsin
- California Housing
- Mall Customers
- Bank Marketing

Tools Requirement- Python libraries: NumPy, Pandas, Matplotlib/Seaborn, Scikit-Learn, SciPy, SHAP, LIME
All implementations must follow responsible and ethical data practices.
Text Books (For Lab Use):

1. Wes McKinney – *Python for Data Analysis*, O'Reilly.
2. Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly.
3. Peter Bruce & Andrew Bruce – *Practical Statistics for Data Scientists*.
4. James et al. – Introduction to Statistical Learning (ISLR2).

References:

1. Charu Aggarwal – *Probability & Statistics for ML*.
2. Scikit-learn documentation (https://scikit-learn.org/stable/).

Draft Copy

Assessment :
Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus**.**

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165118	Artificial Intelligence & Soft Computing Lab		2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165118	Artificial Intelligence & Soft Computing Lab	--	--	--	--	25	25	50

Lab Objectives: Six Lab Objectives

1. To introduce practical applications of AI fundamentals including intelligent agents, PEAS-based problem formulation, and basic PROLOG programming for knowledge representation and reasoning.
2. To implement classical search algorithms such as BFS, DFS, and heuristic-based A* for solving real-world planning and navigation problems.
3. To simulate foundational Neural Network models such as perceptron classification and explore their role in advanced neural architectures including Transformer-based systems.
4. To explore Generative AI applications including LLM-based text generation along with RAG and CAG enhancements.
5. To design and analyze Soft Computing models including fuzzy sets, membership functions, and fuzzy logic controllers for approximate reasoning and adaptive decision-making.
6. To develop and evaluate Hybrid Neuro-Fuzzy systems integrating neural learning with fuzzy inference mechanisms for intelligent automated solutions.

Lab Outcomes: Six Lab Outcomes (Based on Blooms Taxonomy)

1. Explain AI fundamentals and formulate problems using the PEAS model and PROLOG.
2. Apply search algorithms to achieve goal-oriented problem solving.
3. Implement basic neural network learning using perceptron models and interpret Transformer evolution conceptually.
4. Construct LLM-based text generation and enhance it using RAG & CAG techniques.
5. Design and simulate Fuzzy Logic and Neural Network models for adaptive decision making.
6. Create and evaluate a Hybrid Neuro-Fuzzy system for intelligent learning.

DETAILED SYLLABUS: Syllabus related Lab experiment must be considered and mapped with Blooms Taxonomy.

Total six modules for each subject lab (13 weeks) to be distributed among six modules.

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	**Engineering Mathematics:** Matrix Algebra, **Discrete Structures:** Set Theory and Logic	--

Draft Copy

I	**Introduction to Artificial Intelligence and Problem Formulation**	Introduction to AI — applications in engineering; Intelligent agents and PEAS model; Problem formulation in AI systems. Introduction to Data Centers and Cloud Services for AI deployment. Basics of PROLOG — facts, rules, queries; Ethical and Explainable AI. **Self-Learning Topics:** Applications of AI in autonomous systems, PEAS modeling.	**LO1**
II	**Search Strategies and Classical AI Problem Solving**	Search-based problem-solving techniques: uninformed (BFS, DFS) and informed (Best-First, A*) algorithms. Problem representation and use of heuristics in goal-driven tasks. Adversarial search: Mini-max Search. **Self-Learning Topics:** Bidirectional & iterative deepening search.	**LO2**
III	**Neural Networks Fundamentals**	Biological vs. Artificial neuron; Activation functions; Single-layer Perceptron and forward propagation; Introduction to advanced neural architectures with the basic concept of Transformer models as an evolution of neural network design. **Self-Learning Topics:** Gradient Descent intuition.	**LO3**
IV	**Generative AI and Advanced Architectures (LLM + RAG & CAG)**	Introduction to Generative AI workflow. Transformer architecture — Attention mechanism. Language Model Classification: Foundation Models (FM), Large Language Models (LLM), Small Language Models (SLM), Micro Language Models (µLM). Retrieval-Augmented Generation (RAG) using vector databases (FAISS/ChromaDB). Cache-Augmented Generation (CAG) for performance improvement. Text generation/summarization using GPT-2/Mistral/LLaMA-2via HuggingFace/Ollama. **Self-Learning Topics:** Prompt engineering and memory-optimized GenAI pipelines.	**LO4**
V	**Soft Computing Techniques — Fuzzy Logic**	Soft computing concepts, uncertainty and approximate reasoning. Fuzzy sets and membership functions; Design of Fuzzy Logic Controller (Mamdani/Sugeno). Implementation of a basic McCulloch–Pitts Neural Network for logic operations (AND/OR). **Self-Learning Topics:** Adaptive fuzzy systems and rule optimization.	**LO5**
VI	**Learning Rules and Hybrid AI Systems**	Basic learning rules: Perceptron, Delta, Hebbian. Concept of Neuro-Fuzzy System- architecture and benefits (Hybrid Neuro-Fuzzy System integrating fuzzy logic with neural learning). Comparative evaluation with standalone models. **Self-Learning Topics:** Applications of hybrid systems in decision support.	**LO6**

Draft Copy

Text Books:

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach ―Second Edition" Pearson Education.
2. Elaine Rich and Kevin Knight ―Artificial Intelligence‖ Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
3. S. Rajasekaran & G. A. Vijayalakshmi Pai — *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, PHI, Second Edition
4. J. S. R. Jang, C. T. Sun & E. Mizutani — *Neuro-Fuzzy and Soft Computing,* Pearson Education.

References:

1. Ivan Bratko ―PROLOG Programming for Artificial Intelligence‖, Pearson Education, Third Edition.
2. S. N. Sivanandam & S. N. Deepa — *Principles of Soft Computing*, Wiley India.
3. Sebastian Raschka — *Build a Large Language Model (From Scratch) (*MIT Press Draft).
4. OpenAI / Hugging Face — *Introduction to Transformers and LLMs (Free Online).*
5. Jay Alammar — *The Illustrated Transformer* (jalammar.github.io).

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/neural-networks-deep-learning
2.	https://www.coursera.org/courses?query=generative+ai
3.	https://www.coursera.org/learn/introduction-to-ai

List of Experiments.

Sr No	List of Experiments	Hrs
01	Select a problem statement relevant to Artificial Intelligence: Identify the problem, describe it using PEAS (Performance, Environment, Actuators, Sensors), and perform problem formulation	02
02	Introduction to AI Programming Language: Write and execute basic programs in PROLOG.	02
03	Implement uninformed search algorithms (BFS, DFS) to reach a goal state.	02
04	Implement any one of the Informed search techniques E.g. A* algorithm for 8 puzzle problem	02
05	Implement adversarial search using min-max algorithm.	02
06	Creating/building a neural network using basic language (C Programming/Java Programming)	02
07	Train a perceptron to learn AND, OR, XOR gates.	02
08	To implement Basic Supervised / Unsupervised Neural Network learning rules for a problem	02
09	Exploration and Implementation of Generative AI Techniques using Open-Source Tools and Frameworks	02
10	Developing Language Models (Small Language Models (SLM), Micro Language Models (µLM).)	02
11	Design and Implementation of a Retrieval-Augmented Generation (RAG) Framework for Contextual Question Answering	02
12	Development of a Cache-Augmented Generation (CAG) System for Efficient AI Responses	02
13	Implement a simple Fuzzy Logic Controller for real-world control problems.	02
14	Implement Mamdani/Sugeno Fuzzy Inference Systems (FIS) using MATLAB / Python.	02
15	Developing a Hybrid Neuro-Fuzzy System	02

Draft Copy

Sr No	List of Assignments / Tutorials	Hrs
01	Applications and Impact of Artificial Intelligence in Modern Engineering	02
02	Role of Heuristics in Enhancing Search Efficiency in AI Systems	02
03	Open-Source Frameworks and Tools for Generative AI Development	02
04	Retrieval-Augmented Generation (RAG): Architecture and Applications	02
05	Cache-Augmented Generation (CAG): Concept and Performance Optimization in AI Systems	02
06	Fuzzy Logic Controllers and Neural Network Approaches for Adaptive Decision-Making	02
07	Design and Applications of Hybrid Neuro-Fuzzy Systems in Intelligent Automation	02
08	Comparative Study of Learning Approaches in Neural Networks and Fuzzy Logic Systems	02

Assessment:
Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165119	Agile Software Development & DevOps Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165119	Agile Software Development & DevOps Lab	--	--	--	--	25	25	50

Lab Objectives: The student will be able:

1. To understand and implement the Agile Software Development Lifecycle using Jira and integrate it with DevOps tools for continuous project management.
2. To perform version control operations using Git and apply collaboration workflows for software development.
3. To design and configure Continuous Integration / Continuous Deployment pipelines using Jenkins, GitHub Actions, and containerization tools.
4. To automate provisioning, configuration, container orchestration, and cloud resource management using Ansible, Docker, Kubernetes, and Terraform.
5. To study and implement MLOps workflows including model tracking, deployment, and data pipeline automation using MLflow and Airflow.
6. To integrate and implement a complete end-to-end DevOps pipeline through a mini project covering planning, building, testing, deployment, and monitoring.

Lab Outcomes: Upon completion, the student will be able to:

1. Apply Agile methodologies and use Jira for sprint planning, task tracking, and integrating with DevOps workflows.
2. Use Git effectively for version control, source code management, and collaborative team development.
3. Build and manage automated CI/CD pipelines for application development, testing, packaging, and deployment.
4. Containerize, orchestrate, and deploy applications using Docker, Docker Compose, Kubernetes, and cloud infrastructure tools.
5. Implement ML lifecycle management and automated workflows using MLflow, Airflow, and monitoring tools like Prometheus/Grafana/Nagios.
6. Develop and demonstrate an end-to-end DevOps pipeline mini-project integrating all stages of DevOps from code commit to deployment and monitoring.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Knowledge of Programming, Computer Networks, Web Programming.	

Draft Copy

I	**Agile Project Management & DevOps Integration**	Fundamentals of Agile and Scrum, Jira for Agile project lifecycle management, Creating epics, user stories, tasks, sprints, Planning, assigning, tracking sprint progress, Integrating Jira with CI/CD pipelines Agile reporting: Burndown chart, velocity chart. **Self-learning Topics:** Kanban vs Scrum, Agile ceremonies: Sprint review, retrospective	LO 1
II	Version Control & Collaboration using Git	Introduction to Git, GitHub, Basic operations: clone, commit, push, pull, Branching strategies and merging, Git workflows: Feature branching, Git Flow, Handling merge conflicts, GitHub Actions basics **Self-learning Topics:** GitHub Issues & Project Boards Webhooks in GitHub	LO 2
III	CI/CD Automation using Jenkins & Containerization using Docker	Understanding CI/CD and DevOps pipelines, Jenkins installation and job creation, GitHub webhook integration, CI pipeline: build → test → package, Docker fundamentals: images, containers, Creating Dockerfiles, Multi-service deployment using Docker Compose **Self-learning Topics:** Jenkinsfile (Pipeline as Code), Publishing images to Docker Hub/ECR	LO 3
IV	Configuration Automation, Orchestration & Cloud Provisioning	Ansible automation: inventory, playbooks, roles, Provisioning and configuring applications, Kubernetes concepts: Pods, Deployment, Service, Ingress, Managing containerized apps in Kubernetes, Terraform basics: providers, resources, state files, Automating cloud infrastructure creation. **Self-learning Topics:** Helm Charts, Terraform Cloud / Remote state management	LO 4
V	Monitoring, Serverless Computing & MLOps Fundamentals	Monitoring tools: Prometheus, Grafana, Nagios, Exporters, dashboards, alerting, AWS Lambda fundamentals and deployment, MLOps lifecycle overview, MLflow for tracking experiments, versioning & deployment, Airflow basics: DAGs, tasks, operators, scheduling **Self-learning Topics:** CloudWatch metrics for serverless apps, Airflow for ML/ETL automation	LO 5
VI	End-to-End DevOps Pipeline	Designing a full DevOps automation pipeline, Integrating Agile → Git → CI → CD → Containers → Kubernetes → Monitoring, Cloud deployment using AWS/GCP/Azure, Pipeline security basics (DevSecOps), Documentation, demonstration & evaluation. **Self-learning Topics:** Industry-grade DevOps architecture, Cost-efficient cloud deployment strategies	LO 6

Text Books:

1. The DevOps handbook : how to create world-class agility, reliability, and security in technology organizations by Kim, Gene.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.

References

1. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive . By Sricharan Vadapalli, Packt, 2018

Draft Copy

Online Resources:

Sr. No.	Website Name
1.	https://devops.com/most-popular-open-source-devops-tools/
2.	https://github.com/bregman-arie/devops-resources?
3.	https://seblum.github.io/mlops-engineering-book/?

List of Experiments.

Sr No	List of Experiments	Hrs
01	To study and Implement Agile Software Project lifecycle using Jira with DevOps Pipeline Integration.	2
02	To perform version control operations using Git.	2
03	To set up a Continuous Integration Pipeline using Jenkins/GitHub Actions.	2
04	To automate software provisioning and configuration using Ansible	2
05	To containerize applications using Docker.	2
06	To deploy multi-service applications using Docker Compose.	2
07	To deploy and manage containers using Kubernetes.	2
08	To automate application deployment using Github/ Jenkins.	2
09	To provision cloud infrastructure using Terraform.	2
10	To deploy a serverless function using AWS Lambda.	2
11	To implement monitoring using Prometheus / Grafana / Nagios.	2
12	To study and Implement machine learning lifecycle with model development & deployment using MLFlow.	2
13	To study and Implement designing, scheduling, and monitoring workflows and data pipelines using AirFlow.	2
14	To design and implement an end-to-end DevOps pipeline through a mini project.	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1	
01	Choose any one application domain (E-commerce / Healthcare / College ERP / Banking / Food Delivery) and Create Agile Requirements using User Stories & Acceptance Criteria.	2
02	For the user stories created in Assignment 1 do Sprint Planning, Story Point Estimation & Release Planning	
03	Test Case Design using Agile QA Techniques	
	Assignment 2	
01	DevOps Pipeline – CI/CD Setup using GitHub Actions / Jenkins.	2
02	Simulate a Full Agile Process Implementation Using Jira.	
03	Integrate Agile Planning with DevOps Automation.	

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Draft Copy

Subject Code	Subject Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical & Oral	Tutorial	Total
2165120	Web and Mobile Application Development Lab	--	2	--	--	1	--	1

Subject Code	Subject Name	Examination Scheme						
		Theory Marks				Term Work	Practical / Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165120	Web and Mobile Application Development Lab	--	--	--	--	25	--	25

Lab Objectives

1. Understand and implement basic programming principles using different languages such as Kotlin, React, and Flutter.
2. Develop web and mobile applications using modern frameworks (React for web and Flutter for mobile).
3. Implement backend services and RESTful APIs using Kotlin with database connectivity.
4. Integrate frontend, backend, and mobile interfaces into a unified full-stack system.
5. Apply basic machine learning concepts to enhance application functionality using self-developed logic or models.
6. Deploy and evaluate the complete project for performance, usability, and scalability.

Lab Outcomes

1. Demonstrate understanding of modern web and mobile development environments and tools.
2. Develop responsive front-end applications using React components and state management.
3. Implement backend services in Kotlin with database integration and secure authentication.
4. Apply microservices, REST APIs, and deployment techniques for scalable applications.
5. Design cross-platform mobile apps using Flutter and integrate them with backend services.
6. Integrate advanced functionalities such as Firebase, API connectivity, and animations to enhance user experience

Guidelines:

- Mentors / Lab instructors should get the labs performed as an internal project-based lab assigned to a group of 2 - 3 students.
- The project development should focus on covering all Web and Mobile development concepts consisting of React Native, Kotlin and flutter topics as mentioned in the syllabus.
- All the following experiments/tasks, which together make up the mini-project.
- Students are expected to implement the projects integrally and sequentially, demonstrating both frontend and backend development, API integration, authentication, and deployment practices.
- Appropriate project management practices shall be utilized during the entire project life-cycle Requirement gathering and designing then following experiments to be implemented

Draft Copy

Suggested List of experiments:

Sr. No.	Module Title	Experiment	LO
1	**Core Web & Mobile Development Concepts**	Study and demonstrate REST API architecture with sample CRUD operations using Postman.	LO1
2	**Front-End Development with React**	Create a simple React app with components and JSX; display static product/user data.	LO2
3	**Front-End Development with React**	Implement state management using Hooks and Context API; add form handling and validation.	LO 2
4	**Kotlin Programming Core Concepts**	Write Kotlin programs to demonstrate data types, collections, and OOP concepts (classes, inheritance).	LO 3
5	**Kotlin Programming Core Concepts**	Implement exception handling and use collection operations (map, filter, reduce).	LO 3
6	**Kotlin Backend Development**	Create RESTful APIs using Spring Boot for basic CRUD operations (users/products).	LO 4
7	**Kotlin Backend Development**	Integrate database using ORM (Hibernate/JPA) and perform CRUD operations.	LO 4
8	**Kotlin Backend Development**	Implement user authentication using JWT/OAuth2 and test with Postman.	LO 4
9	**Kotlin Backend Development**	Build microservice architecture (two services communicating via REST).	LO 4
10	**Kotlin Backend Development**	Deploy Spring Boot app to a cloud platform (Render/Heroku) or local server.	LO 4
11	**Flutter Foundations & Dart Programming**	Create a Flutter "Hello App" and explore basic Widgets and Layouts.	LO 5
12	**Flutter Foundations & Dart Programming**	Implement State Management (Provider/ScopedModel) for dynamic data display.	LO 5
13	**Advanced UI & Firebase Integration**	Integrate REST API from Kotlin backend into Flutter app using HTTP package.	LO 6
14	**Advanced UI & Firebase Integration**	Implement Firebase Authentication (Signup/Login) and Firestore database connectivity.	LO 6
15	**Advanced UI & Firebase Integration**	Develop and deploy a complete Flutter app with backend API and Firebase storage (Final Project Demo).	LO 6

One sample project with experiment mapping

Phase	Technology
Front-End (React)	Admin dashboard for managing students and courses (Expt 2–3)
Backend (Kotlin + Spring Boot)	APIs for CRUD operations on students, attendance, and grades (Expt 6–9)
Database Integration	Persistent storage of academic records (Expt 7)
Authentication	Role-based access (admin, faculty, student) (Expt 8)
Mobile (Flutter)	Student app for viewing attendance, grades, and notices (Expt 11–15)
Final Demo	Fully functional academic management system across platforms (Expt 15)

Note: Implement any 10-12 experiments from the given list

Draft Copy

Text Books

1. Learning React: Modern Patterns for Developing React Apps – Alex Banks & Eve Porcello, O'Reilly (2023).
2. Node.js Design Patterns – Mario Casciaro, Packt (2023).
3. Kotlin Programming: The Big Nerd Ranch Guide by Josh Skeen and David Greenhalgh.
4. Kotlin for Backend development: Build scalable REST APIs, Microservices and Web Applications with Ktor, Spring Boot and Coroutines, Stokes J. Harrett.
5. Kotlin In-Depth: A Guide to a Multipurpose Programming Language for Server-Side, Front-End, Android, and Multiplatform Mobile - Aleksei Sedunov, BPB Publications.
6. Simplifying Application Development with Kotlin Multiplatform Mobile - Robert Nagy Packt publication
7. Beginning App Development with Flutter Create Cross-Platform Mobile Apps — Rap Payne, Apress
8. Flutter Cookbook, by Simone Alessandria, Brian Kayfitz, O'Reilly

Reference Books

1. Full-Stack Web Development with Node, Express, and React – Adam Freeman, Apress (2022)
2. Head First Kotlin - Dawn Griffiths & David Griffiths, O'Reilly.
3. Kotlin Development Complete Guide create 45 android applications including Jetpack Composer – Henry Codwell.
4. Adam Bretz and Colin J. Ihrig, "Full Stack JavaScript Development with MEAN", SitePoint Pty. Ltd., 2015 Edition.
5. Android UI development with Jetpack Compose - Thomas Kunneth, Packt Publishing.
6. Scalable Android applications in Kotlin.
7. E-Commerce, Pondicherry university

Online Resources:

1. https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012
2. https://www.coursera.org/learn/advanced-programming-in-kotlin
3. https://www.coursera.org/learn/advance-kotlin-for-multi-platform-development
4. https://www.pluralsight.com/paths/android-development-with-kotlin-fundamentals
5. https://github.com/androiddevnotes/awesome-android-learning-resources
6. https://www.udemy.com/course/android-app-development-with-kotlin-beginner-to-advanced
7. https://www.codecademy.com/learn/learn-kotlin
8. https://www.codingblocks.com/android-development-using-kotlin.html

Assignment 1: MAD (Any one)

1. To Study basics of ReactJS and React Native language and design/develop basic Web App
2. To include Files and JSON data in App
3. To build interactive App by including Flutter Gestures and Animations

Assignment 2: Kotlin Advanced Concepts (Any one)

1. To study the requirement for progressive web application for Ecommerce using the concept of service worker, Webapp Manifest and framework tools
2. To Design a wireframe for simple PWA for E-commerce website
3. Case study for successful real life implementation of PWA.

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165121	User Experience Design with VR Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165121	User Experience Design with VR Lab	--	--	--	--	25	--	25

Lab Objectives

1. Understand how to design user interfaces that are effective, easy to use, and meet user needs.
2. Learn different methods for creating prototypes and testing user experience.
3. Apply the principles of good UI and UX design in real projects.
4. Install and set up Unity software and learn how VR devices work.
5. Create and explore simple VR scenes using Unity.
6. Perform usability and experience evaluation of UI and VR prototypes

Lab Outcomes

Learners will be able to :

1. Use suitable tools and methods to design and present user interface models.
2. Build a high-fidelity prototype that represents a complete design solution.
3. Evaluate user experience using standard testing techniques.
4. Set up a VR development environment and use VR devices like HTC Vive, Google Cardboard, or Samsung Gear VR.
5. Develop a basic VR scene and add interactive objects.
6. Conduct structured usability evaluations of UI and VR applications using various methods.

Suggested List of Experiments

Sr No	Suggested List of Experiments	Hrs
01	Project Proposal and Requirement Gathering (Choose the project). Briefly state the problem(s) that the project will seek to solve. Take the user's point of view. Consider what the user's goals are, and what obstacles lie in the way.	02
02	Create 2 Personas on the project topic selected.	02
03	Creation of Scenario: Write a scenario that involves all of the tasks identified for the chosen project.	02
04	Creating a Wireframe on selected problem statements.	02
05	Low Fidelity prototype using Figma tool.	02

Draft Copy

06	High Fidelity prototype using Figma tool.	02
07	Usability Evaluation of the Design Testing of User Interface from Third Party(Test scripts)	02
08	Perform Rapid Evaluation on prototype design.	02
09	Perform Heuristic evaluation technique on the prototype design.	02
10	Emotional Design Evaluation of a user interface design by analyzing user reactions and emotional responses at various interaction points, in order to identify design elements that evoke positive, negative, or neutral feelings.	02
11	Installation of Unity and Visual Studio, setting up Unity for VR Development, understanding documentation of the same.	02
12	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	02
13	Understand basic VR environment design using Unity's *Play Mode* (desktop simulation).	02
14	Prototype VR Interaction using Mouse and Keyboard	02
15	Evaluate VR Usability using Video/Screen-based Heuristic Analysis	02

Useful Links	
1	https://www.coursera.org/professional-certificates/google-ux-design
2	https://nptel.ac.in/courses/124107008
3	https://www.coursera.org/learn/develop-augmented-virtual-mixed-extended-reality-applicatio ns-webxr-unity-unreal
4	https://tih.iitr.ac.in/AR-VR.html

Sr No	Suggested List of Assignments / Tutorials	Hrs
	Assignment 1	
01	While creating your project proposal, how did you identify the **main user problem** and define measurable **user goals**? Give an example from your project.	2 hrs
02	How did the **personas** you created influence your design decisions and the features included in your prototype?	
03	When designing your **wireframe**, what layout or navigation issues did you face, and how did you resolve them?	
	Assignment 2	
01	Describe the process you followed to create your **low-fidelity prototype in Figma**. How did it help before moving to the high-fidelity version?	2 hrs
02	During **usability testing**, what kind of feedback did you receive from users, and what changes did you make based on that feedback?	
03	In your **Unity VR project**, explain how you applied transformations (position, rotation, scale) to objects and added multimedia components like audio or video.	

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165122	Computer Network Lab		2	-		1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165122	Computer Network	--	--	--	--	25	--	25

Lab Objectives:

1. To introduce students to networking components, commands, cables, and connectors essential for building and troubleshooting networks.
2. To enable students to design and simulate various network topologies and test their connectivity using modern simulation tools.
3. To develop understanding of data communication protocols and techniques such as sliding window and error detection/correction methods.
4. To provide hands-on experience in configuring routing protocols and core network services like FTP and DNS.
5. To enhance students' skills in network analysis, traffic monitoring, and performance evaluation using tools like Wireshark.
6. To acquaint students with advanced networking concepts such as VLANs, NAT, and Software Defined Networking (SDN).

Lab Outcomes:

1. Identify and use different network cables, connectors, and commands for establishing and troubleshooting network connections.
2. Design and simulate wired LANs and complex topologies, performing connectivity tests and file sharing between devices.
3. Implement and analyze data transmission techniques including sliding window protocols and error detection/correction algorithms.
4. Configure and verify static and dynamic routing protocols (RIP, OSPF) and network services like FTP and DNS.
5. Capture, analyze, and interpret network traffic using Wireshark to evaluate protocol behavior and network performance.
6. Design and simulate VLANs, NAT, and SDN topologies to understand modern network architectures and their applications.

List of Experiments.

Sr No	List of Experiments	LO
01	To study different networking commands and network cables & connectors.	LO1
02	Simulate a wired LAN using networking devices and test connectivity using PING utility.	LO1
03	Create different network topologies using any network simulator (e.g. Cisco Packet Tracer, GNS3). For each topology perform connectivity tests such as ping, file sharing between devices.	LO2
04	Simulate sliding window protocols using any packet tracer tool (any one from Stop and Wait, Go-Back-N, Selective Repeat)	LO3

Draft Copy

05	Implementation of error detection and correction techniques (any one from CRC, Hamming Code, Checksum) using any programming language.	LO3
06	To analyze network traffic using the Wireshark tool. Perform following activities: • Packet capture and filtering • Analyze ethernet frames • IP packet analysis • ICMP analysis (PING test) • HTTP/HTTPs request analysis • Analyze header of TCP /UDP dump..	LO2, LO4, LO5
07	To configure and verify static routing and dynamic routing protocols (RIP and OSPF) in a simulated network environment using Cisco Packet Tracer.	LO4
08	To implement Socket programming using any programming language.	LO4, LO5
09	To configure FTP server and transfer files to demonstrate the working of the same.	LO5
10	To configure the DNS server and to demonstrate the working of the same.	LO5
11	To design and simulate VLANs on switch/router using any network simulator (e.g. Cisco Packet Tracer, GNS3)	LO6
12	To design and simulate NAT on router using any network simulator (e.g. Cisco Packet Tracer, GNS3)	LO6
13	Implementation of small SDN based topology using Mininet.	LO6

Online Resources:

Sr. No.	Website Name
1.	NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/
2.	CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US

Sr No	List of Assignments / Tutorials	Hrs
01	Assignment covers the topics from the first three units.	CO1,CO2,CO3
02	Assignment covers the topics from the last three units.	CO4,CO5,CO6

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practicals based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Draft Copy

IKS

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
IKS501	Indian Knowledge System	--	2*+2	--	--	2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
IKS501	Indian Knowledge System	--	--	--	--	--	50	--	50

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

Rationale:

The proposed IKS modules contextualize core engineering principles across Computer, IT, EXTC, Electronics, and Electrical engineering—within scientifically analyzable Indian knowledge traditions. By mapping indigenous practices to modern technical frameworks, students critically examine foundational engineering concepts through interdisciplinary and systems-based perspectives. This approach strengthens conceptual understanding, promotes analytical reinterpretation of traditional knowledge using contemporary tools, and aligns engineering education with culturally rooted yet technically rigorous inquiry.

Course Objectives:

1. Provide foundational knowledge of Indian acoustic practices.
2. To Understand Ancient Indian use of renewable energy.
3. Introduce Sanskrit grammar in simple language, with special attention to its rule-based structure.
4. Learners should be able to explain key contributions of ancient Indian mathematics in simple language.
5. Understand how Indian traditions of logic, language, knowledge organization, design thinking, governance, sustainability, and wellbeing can support modern IT solutions.
6. Understand the basic structure of Sanskrit grammar and relate selected grammatical ideas to computational modules used in information technology.

Course Outcomes:

On successful completion of the course learner will be able to:

1. Analyze traditional Indian acoustic practices and interpret their relevance to frequency analysis.
2. Apply the traditional knowledge to modern energy solutions and sustainable engineering.
3. Apply Indian Knowledge Systems in linguistics, Sanskrit-based NLP, traditional number systems, measurement practices, and ancient rainfall prediction techniques using calendars, nakshatras, and natural indicators.
4. Apply basic modular arithmetic, pattern recognition, and place-value ideas to simple IT examples such as coding, data representation, and checksums.
5. Apply basic concepts of Indian Knowledge Systems such as logic, language, knowledge organization, ethics, sustainability, and responsible decision-making to design simple technology-based solutions like apps, chatbots, databases, or awareness tools.
6. Apply basic Sanskrit grammar concepts and Pāṇinian rule-based principles to understand formal grammar, algorithms, and simple computational language processing applications in IT.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping

Draft Copy

I	**Acoustic Science in Vedic Chanting and Indian Music**	I.1 Concept of Nāda (Sound) and Shruti • Physical interpretation: frequency, pitch, harmonic content • Shruti system (22 microtones) and frequency discrimination I.2 Resonance and Architectural Acoustics • Temple architecture and echo control • Natural acoustic amplification • Relation to reverberation time and standing waves I.3 Structured Sound Encoding in Vedic Recitation • Pada, Krama, Ghana patha recitation methods • Built-in redundancy → error detection and correction • Parallel to digital communication redundancy schemes **Self-Study:** Study of sounds and analysis in time and frequency domains.	**10**	**CO1**
II	**Tradition**	1. Introduction to IKS in renewable energy 2. Ancient Indian use of renewable energy a) Solar Energy: Solar Architecture, Solar Heating Systems b) Wind Energy: Ancient Windmills, Wind-Powered Water Lifting c) Water (Hydro) Energy: Ancient Watermills, Irrigation Systems d) Bioenergy: Biomass Fuel, Bioenergy Production 3. Integration of ancient Traditional knowledge of renewable energy with Modern Technologies. **Students shall study the following to understand the use of ancient technologies in Renewable Energy:** • Gujarat Windmills • Watermills in the Indus Valley • Konark Sun Temple	**10**	**CO2**
III	**Linguistics, Number Systems, and rainfall Prediction**	Linguistics: • Components of language • Panini's work on Sanskrit grammar • Role of Sanskrit in Natural Language Processing Number System and Units of Measurement • Number system in India- Historical Evidence • Decimal System • Unique Approaches to represent Numbers-Katapayadi System • Pingala and the Binary System Ancient Indian techniques for predicting Rainfall • Introduction • Elements of the Indian Calendar • Panchang and Nakshtras • Non-bio indicators and Bio indicators **Self-Learning:** • An Ecosystem for Sanskrit Language Processing • Salient Features of the Indian Numeral System • Salient Features of the Indian Numeral System	**10**	**CO3**

Draft Copy

		• Measurement for Time Distance and Weight • Measurement of Time- An Illustration from Purana		
IV	**Mathematics Foundations in Ancient India and Its Relevance to Information Technology**	4.1 Historical context and number ideas: Indian knowledge systems, the development of numeration, zero as placeholder and number, decimal place value. 4.2 Major contributors and procedural mathematics: Aryabhata, Brahmagupta, Bhaskara II, and examples of rule-based or sutra-like calculation methods in plain language. 4.3 Algorithmic thinking and computation: Historical mathematical procedures connect to algorithms, flowcharts, decomposition, and computational logic. 4.4 Modular arithmetic, binary-style ideas, and data representation: Clock arithmetic, remainders, cyclic reasoning, two-state representation, bits, bytes, and simple binary conversion. 4.5 Relevance to IT and synthesis: Coding, checksums/parity, simple cryptography idea, data representation, and how ancient mathematical thinking supports modern IT problem-solving. **Self-Learning Topics and Activities:** • Zero and place value in daily life • Algorithmic thinking • Modular arithmetic • Binary representation	**10**	**CO4**
V	**Indian Logic and its applications in Information Technology**	5.1 Introduction to IKS and its relevance to contemporary technology. 5.2 Indian logic, language, and computation with simple examples. 5.3 Knowledge organization, data, and decision-making. 5.4 Ethics, society, sustainability, and responsible digital systems. 5.5 Mini activity where students design a small app, chatbot, database, or awareness tool using one IKS principle. **Self-Learning Topics and Activities:** • Basic ideas of Nyaya logic: observation, inference, debate, and evidence. • Introduction to Paninian grammar and why structured language matters in NLP. • Traditional knowledge classification and its connection with databases and knowledge graphs. • Technology for Indian languages, heritage preservation, digital museums, and public services. • Design thinking activity: identify a local problem and propose an IKS-inspired IT solution.	**10**	**CO5**
VI	**Sanskrit Grammar and Computational Modules**	6.1 Introduction to Indian Knowledge Systems, Sanskrit as structured language, and relevance to IT. 6.2 Basic Sanskrit grammar concepts: sounds, words, roots, suffixes, sandhi, and sentence structure. 6.3 Pāṇinian grammar as a rule-based system; relation with algorithms and formal grammar.	**10**	**CO6**

Draft Copy

		6.4 Computational modules: tokenization, morphology, parsing, language rules, and NLP examples. 6.5 Demonstration of Sanskrit digital tools, reflection activity, and discussion on interdisciplinary applications. **Self-Learning Topics and Activities:** • Basic Sanskrit alphabet and sound classification. • Simple sandhi examples and word joining rules. • Introduction to Pāṇini and the Aṣṭādhyāyī. • Dhātu, pratyaya, pada, and kāraka in simple terms. • Basics of natural language processing: tokenization, morphology, and parsing. • Exploration of Sanskrit digital dictionaries and Sanskrit reader tools. • Comparison of grammar rules and computer program rules.		

Text and Reference Books or Articles:

1. Acoustics of Chants, Conch-Shells, Bells & Gongs in Worship Spaces. Available: https://www.academia.edu/38208102/Acoustics_of_Chants_pdf?utm_source=chatgpt.com
2. V. R. S. Nalluri, V. J. K. S. Sonti, and G. Sundari, "Analysis of frequency dependent Vedic chanting and its influence on neural activity of humans," International Journal of Reconfigurable and Embedded Systems, vol. 12, no. 2, pp. 230–239, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp230-239.
3. The Garland Encyclopedia of World Music: South Asia. Available: https://api.pageplace.de/preview/DT0400.9781351544399_A30892028/preview-9781351544399_A30892028.pdf?utm_source=chatgpt.com
4. J. Pattanayak, Y. Belur, and J. Ilavarasu, "Acoustic and psychological effects of Omkar chanting in novice Indian adults: A quasi-experimental study," Journal of Clinical and Diagnostic Research, 2025, doi: 10.7860/JCDR/2025/77365.21111. Available: https://www.jcdr.net/articles/PDF/21111/77365_CE%5BRa1%5D_F%28IS%29_QC%28PS_SS%29_P F1%28AB_IS%29_redo_PFA%28IS%29_PB%28AB_IS%29_PN%28IS%29.pdf?utm_source=chatgpt. com
5. "Indian Knowledge System: Integrating Heritage with Engineering" by Gagan Bansa, Deep Science Publishing; ISBN: 978-93-49307-77-3 E-ISBN: 978-93-49307-29-2;
6. The textbook of Indian Knowledge Systems, University of Mumbai, ISBN: 978-93-5915-553-1
7. Introduction to Indian Knowledge System: Concepts and Applications, Mahadevan, B., Bhat, Vinayak Rajat, Nagendra Pavana R.N. by PHI Learning Pvt Ltd. Delhi, ISBN: 978-93-91818-20-3
8. Echoes of the Monsoon: Ancient Indian techniques for predicting Rainfall, Janak Joshi, Kandarp
9. Parmar, Lakhan Jain, Sharmistha Bhowmik and Bindu Bhatt, DOI: https://www.doi.org/10.22271/27067483.2025.v7.i7a.386.
10. Kim Plofker, Mathematics in India: A clear historical overview of Indian mathematical developments, suitable for faculty use and advanced self-learners.
11. George Gheverghese Joseph, The Crest of the Peacock: An accessible introduction to non-European mathematical traditions, including India.
12. Charles Petzold, Code: The Hidden Language of Computer Hardware and Software: A beginner-friendly bridge from numbers and symbols to data representation and computing.
13. Simon Singh, The Code Book: Useful for simple explanations of ciphers, encryption, and the mathematics behind communication security.
14. T. A. Sarasvati Amma, Geometry in Ancient and Medieval India: A more specialized reference for deeper exploration of traditional mathematical methods.
15. Mahadevan, B., Bhat, V. R., and Nagendra Pavana R. N. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning.

Draft Copy

16. Kapur, K., and Singh, A. K. (Eds.). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla.
17. Bora, M. C. Bharatiya Knowledge Systems. Khanna Publishing House.
18. Rao, S. S. Engineering and Technology in Ancient India. Indian National Science Academy / related academic editions.
19. Selected open articles or chapters from IKS Division / AICTE / Ministry of Education resources may be used as supplementary reading.
20. S. D. Joshi and J. A. F. Roodbergen, The Aṣṭādhyāyī of Pāṇini.
21. V. N. Jha, The Philosophy of Sanskrit Grammar.
22. Peter M. Scharf, Sanskrit Syntax and Discourse Structures.
23. Daniel Jurafsky and James H. Martin, Speech and Language Processing — for introductory NLP concepts.
24. Rajeev Sangal, Akshar Bharati, and Vineet Chaitanya, Natural Language Processing: A Paninian Perspective.

Other Online Resources:

Sr. No.	Website Name
1.	NPTEL course page: For a structured academic overview of mathematics in India. https://nptel.ac.in/courses/111101080
2.	NPTEL YouTube playlist: For lecture-based learning that can support flipped or self-paced study. https://www.youtube.com/playlist?list=PLbMVogVj5nJThf31TNSQzuN7zqxe7HdRN
3.	IKS India: For broader Indian Knowledge Systems context and additional learning resources. https://www.nist.gov/cryptography
4.	Responsible AI and India-context resources: https://indiaai.gov.in/
5.	IKS Division, Ministry of Education / AICTE: https://iksindia.org/
6.	NPTEL: Introduction to Pāṇinian Grammar URL: https://onlinecourses.nptel.ac.in/noc24_hs44/preview
7.	Sanskrit Heritage Reader URL: https://sanskrit.inria.fr/

List of Practical and List of Assignments:

Sr No	List of Experiments	Hrs
01	Record a temple bell sound, conch sound, or classical "Sa" note and analyze it by plotting its time-domain waveform and frequency spectrum, identifying the fundamental frequency and harmonics, and comparing the observed harmonic structure with theoretical frequency ratios of Just Intonation and Equal Temperament.	04
02	Observe the movement of sunlight and shadow around a building, monument, or open courtyard at different times of the day, record the shadow directions and lengths, and analyze how solar position can guide orientation, time estimation, and passive architectural planning in the context of traditional Indian solar design.	04
03	Examine images, plans, or a simple model of the Konark Sun Temple or any sun-oriented structure, identify how geometry, orientation, and sunlight interaction are used in the design, and prepare a visual explanation of how ancient solar architecture supports modern sustainable building concepts.	04
04	Construct a small paper or cardboard windmill, test its rotation under natural wind or fan airflow at different speeds, and document how blade movement demonstrates the basic principle of converting wind energy into mechanical motion, linking it to traditional wind-powered systems and modern turbines.	04
05	Build a simple working model using cups, straws, thread, or cardboard to simulate water lifting or channel-based irrigation, observe how water can be moved with minimal energy input, and relate the model to ancient Indian irrigation systems and watermill-based practices.	04

Draft Copy

06	Survey homes or nearby community spaces to identify traditional biomass fuels such as wood, agricultural residue, or cow-dung cakes, record their current use, benefits, and limitations, and compare the observed practices with modern ideas of bioenergy and sustainable fuel management..	04
07	Pronounce and classify selected Sanskrit sounds into vowels and consonants using a simple sound chart, group them according to articulation patterns, and analyze how this systematic sound organization supports pronunciation training, language learning, and speech-processing applications.	04
08	Create all possible combinations of short and long syllabic patterns of a fixed length, tabulate them systematically, and compare the resulting patterns with binary 0/1 combinations to understand how ancient combinatorial thinking connects with digital representation.	04
09	Solve real-life problems based on clocks, calendars, weekdays, or repeated events using modular arithmetic, record the remainder-based reasoning used in each case, and explain how cyclic patterns support time management, scheduling, and introductory cryptographic logic.	04
10	Convert selected decimal numbers into binary form, arrange them in bit patterns, append parity bits for even or odd parity, and interpret how binary representation and simple error-detection methods are used in digital systems for reliable data storage and transmission.	04
11	Use a basic Sanskrit dictionary, transliteration tool, or reader tool to input selected words, observe the output such as root, meaning, or grammatical analysis, and describe how such tools support language technology and digital preservation.	04
12	Select a real-life problem such as fake-news verification, waste segregation, or campus awareness, identify observation, evidence, inference, and conclusion using the basic framework of Indian logic, and represent the reasoning process as a decision tree, chatbot flow, or awareness tool concept.	04
13	Take a simple historical or rule-based calculation procedure associated with Indian mathematics, rewrite it as a sequence of input-process-output steps, and convert it into a flowchart to demonstrate the connection between procedural mathematics and modern algorithmic thinking.	04
14	Represent selected large and small numbers using a place-value table, deliberately insert or remove zero in different positions, and analyze how zero functions both as a number and as a placeholder, showing its importance in arithmetic, accounting, and digital data representation.	04
Sr No	**List of Assignments**	**Hrs**
01	Study the concepts of Nāda and Shruti, temple acoustics, and Vedic recitation methods, and briefly explain their relation to modern concepts of frequency, resonance, reverberation, and redundancy/error correction in digital communication systems. Also write a short note on how Indian Knowledge Systems demonstrate scientific understanding of sound and communication.	01
02	Study the ideas of solar architecture in traditional Indian structures such as the Konark Sun Temple or courtyard-based houses, and explain how sunlight, orientation, shading, and ventilation were used for thermal comfort and energy efficiency. Also write a short note on how these traditional design ideas are relevant to modern sustainable architecture and climate-responsive planning..	01
03	Study one traditional Indian renewable energy practice such as wind-powered water lifting, watermills, solar heating, or biomass use, and compare it with one modern renewable energy technology. Explain how traditional knowledge systems supported sustainable living and discuss how these ideas can inspire present-day energy conservation and green technology.	01
04	Study the basic ideas of Pāṇinian grammar as a structured rule-based system, and explain	01

Draft Copy

	how concepts such as sounds, roots, suffixes, and word formation relate to modern ideas in natural language processing, machine translation, and language analysis. Also write a short note on why Sanskrit grammar is often considered relevant to computational linguistics.	
05	Study the historical development of zero as both a number and a placeholder, and explain how the decimal place-value system simplified arithmetic and record keeping. Also write a short note on how zero and place value continue to be essential in digital computation, data storage, and computer-based numerical processing.	01
06	Study the basic ideas of Indian logic, especially observation, inference, debate, and evidence in the Nyāya tradition, and explain how these ideas support careful reasoning and decision-making. Also write a short note on how such logical methods can be applied in present-day contexts such as data analysis, fake-news verification, ethical decision-making, and responsible digital systems.	01
07	Identify one idea from Indian Knowledge Systems in the syllabus, such as renewable energy, Indian numeration, Indian logic, or Sanskrit grammar, and propose a simple real-life solution such as an app, chatbot, awareness campaign, digital archive, poster, database, or educational tool based on that idea. Explain the problem being addressed, the IKS principle used, and how the proposed solution can be useful in present-day society.	01
08	Study the basic Sanskrit grammar concepts of sounds, words, roots, suffixes, sandhi, and sentence structure, and explain how grammar rules work like structured instructions. Also compare grammar rules with computer program rules, and briefly discuss how tokenization, morphology, and parsing in computational systems resemble traditional rule-based language analysis.	01

Assessment:

Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term Work Journal must include at least 4 assignments
Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 20 Marks (Assignments) + 5 Marks (Attendance)

Draft Copy

Sem VI

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166111	Machine Learning and Swarm Intelligence	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-I	IAT-I + IAT-II					
2166111	Machine Learning and Swarm Intelligence	20	20	40	60	2	--	--	100

Rationale :
This course provides a strong foundation in Machine Learning concepts and techniques, covering both traditional algorithms and modern bio-inspired optimization methods. It enables students to design, implement, and analyze intelligent models for solving real-world problems efficiently. The inclusion of swarm intelligence introduces learners to emerging approaches in adaptive and scalable computation.

Course Objectives:

1. To understand the fundamental concepts, types, and workflow of Machine Learning.
2. To learn and apply key algorithms for classification, clustering, and association rule mining.
3. To study and implement linear models for regression and classification problems.
4. To explore feature selection and dimensionality reduction techniques for handling high-dimensional data.
5. To understand the concepts, characteristics, and mechanisms of swarm intelligence and bio-inspired algorithms.
6. To analyze the use of swarm intelligence algorithms in optimization and Machine Learning applications.

Course Outcomes: On successful completion, of course, learner/student will be able to:

1. Explain the fundamental concepts of Machine Learning, its applications, and performance evaluation methods.
2. Apply classification, clustering, and association rule algorithms to analyze data patterns.
3. Implement linear models such as regression, logistic regression, and SVM for prediction and classification tasks.
4. Use dimensionality reduction and feature selection techniques to simplify high-dimensional data.
5. Describe the principles and working of swarm intelligence and bio-inspired optimization algorithms.
6. Analyze the performance and applications of swarm intelligence-based algorithms in solving real-world optimization and machine learning problems.
7.

DETAILED SYLLABUS:

Draft Copy

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0		Linear Algebra, Vectors, Probability and statistics, Programming Concepts		
I	**Introduction to Machine Learning**	1.1 Introduction to Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps of developing a Machine Learning Application. 1.2 Supervised and Unsupervised Learning: Concepts of Classification, Clustering and prediction 1.3 Training, Testing and validation dataset, cross validation, over fitting and under fitting of model Performance Measures: Measuring Quality of model- Confusion Matrix, Accuracy, Recall, Precision, Specificity, F1 Score, RMSE. 1.4 Symmetric Positive Definite Matrices, Determinant, Trace, Eigenvalues and vectors, Orthogonal Projections, Diagonalization, SVD and its applications **Self-Learning :** Transfer Learning and Self-Supervised Learning	**8**	**CO1**
II	**Classification Clustering and Association Rules**	2.1 Classification Algorithm: Random Forest 2.2 Clustering Algorithms - K means, Hierarchical 2.3 Rule based Algorithm- Apriori **Self-Learning :** Semi-Supervised and Active Learning Techniques	**6**	**CO2**
III	**Linear Models**	3.1 The least-squares method, Multivariate Linear Regression, Regularized Regression, Using Least-Squares Regression for classification 3.2 Support Vector Machines Linear SVM Maximum margin classifier Support vectors Hard margin vs. soft margin Non-Linear SVM Kernel trick Common kernels: polynomial, RBF, sigmoid 3.3 Logistic regression **Self-Learning :** Support Vector Regression and Multi-class SVMs	**6**	**CO3**
IV	**Dimensionality Reduction**	4.1 Curse of Dimensionality. 4.2 Feature Selection (Filter Method, Embedded Method, Wrapper Method) and Feature Extraction 4.3 Dimensionality Reduction Techniques, Principal Component Analysis. **Self-Learning :** t-SNE and UMAP for Data Visualization , - Feature Importance and Selection Metrics,- Relationship between Dimensionality and Model Accuracy	**5**	**CO4**
V	**Swarm Intelligence and Bio-Inspired Computation**	5.1 Introduction to Swarm Intelligence and Bio-inspired computation, 5.2 Current Issues in Bio-Inspired Computing, 5.3 Classifications and Terminology 5.3 Tuning of Algorithm-Dependent Parameters Necessity for Large-Scale and Real-World Applications Choice of Algorithms , Search for	**8**	**CO5**

Draft Copy

		the Magic Formulas for Optimization, sense of an Algorithm, Algorithms and Self-Organization , Links Between Algorithms and Self-Organization,The Magic Formulas, Characteristics of Metaheuristics, Intensification and Diversification, Randomization Techniques , 5.4 Swarm-Intelligence-Based Algorithms , Ant Algorithms , Bee Algorithms , Bat Algorithm , Particle Swarm Optimization , Firefly Algorithm , Cuckoo Search , Flower Pollination Algorithm **Self-Learning :** Hybrid Optimization (PSO + GA) and Reinforcement Learning-based Optimization		
VI	**Analysis of Swarm Intelligence Based Algorithms**	6.1 Introduction , 6.2 Optimization Problems , 6.3 Applications of Swarm Intelligence in Machine Learning, swarm Intelligence Based Optimization Algorithms: Ant Colony Optimization, Particle Swarm Optimizer , ABC Algorithm , Firefly Algorithm , Cuckoo Search Algorithm Bat Algorithm **Self-Learning :** Quantum-Inspired and Multi-Objective Swarm Optimization	**6**	**CO6**

Text Books:

1. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher Education
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖,
3. James Kennedy & Russell Eberhart, Swarm Intelligence, Morgan Kaufmann
4. Handbook of Swarm Intelligence: Concepts, Principles and Applications (Panigrahi, Shi & Lim, Springer, 2011)

References:

a. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
b. Particle Swarm Optimization by Maurice Clerc (2006)
c. Swarm Intelligence Optimization: Algorithms and Applications- Abhishek Kumar, Pramod Singh Rathore, Vicente Garcia Diaz & Rashmi Agrawal, Published by Wiley

5. Shai Shalev-Shwartz, Shai Ben-David, ―Understanding Machine Learning‖, Cambridge University Press

Online References:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/machine-learning
2.	https://onlinecourses.nptel.ac.in/noc25_me22/preview
3.	https://www.coursera.org/learn/evolutionary-computation-and-its-engineering-applications

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Draft Copy

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part(a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Theory	Pract.	Total
2166112	Data Analytics and Visualization	3	-	-	-	3	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-I					
2166112	Data Analytics and Visualization	20	20	40	60	2	--	--	100

Rationale:

Data visualization, prediction, and analytics play a crucial role in enabling engineering students to interpret complex technical data with clarity and precision. Visual analytics tools such as graphs, dashboards, and statistical charts help students quickly identify patterns, anomalies, performance issues, and design optimization opportunities. By applying analytical and predictive techniques—such as regression, trend forecasting, and data modelling—students can make informed, data-driven decisions in experiments. These skills enhance problem-solving abilities, improve the quality of engineering solutions, and prepare students for industry roles where data-centric thinking, predictive analytics, and evidence-based decision-making are essential.

Course Objectives:

1. To introduce the concept of fundamentals of Data Analytics.
2. To preprocess data and build regression models.
3. To understand the fundamental concepts of Data exploration and anomaly detection.
4. To understand, apply time series techniques in real world scenarios.
5. To understand and apply the knowledge and skills to perform exploratory data analysis visualizations using Python.
6. To enable students to develop interactive visualizations and dashboards using R and modern Tools.

Course Outcomes: At the end of the course, the students should be able to

1. Understand the basics of data analytics.
2. Apply data pre-processing techniques and evaluate regression models using analytical tools to derive accurate and data-driven predictions.
3. Apply data exploration and data anomaly detection techniques to solve real world data analysis problems.
4. Apply time series modelling techniques for trend analysis, forecasting, and decision-making in real-world scenarios.
5. Apply exploratory data analysis techniques, and develop visualizations in R to support data-driven decision making.
6. Create, customize, integrate visualizations using Python and BI tools to design interactive dashboards.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Data warehouse and mining and Python Programming		
I	Introduction to Data Analytics	**1. Background and Overview of Data Analytics:** Key Roles for a Successful Analytics, Data Analytics life cycle, Analytics value chain. **2.** Types of analytics, Roles in Industry,	3	CO1

Draft Copy

		Business Analytics in Practice, Ethics, and Privacy. Data Discovery, Data Preparation, Common tools for data analytics. **Self-learning Topics:** Data sources, relational DB design, NoSQL.		
II	**Data Pre-processing and Regression Model Evaluation**	**1. Data Pre-processing:** Data Cleaning, Normalization and Standardization, Outlier Detection and Handling, PCA, Feature Creation and Selection, Encoding Categorical Variables, Correlation and Multicollinearity Analysis. **2.Regression Models:** The Regression Equation Fitted Values and Residuals, Least Squares Method, Simple vs. Multiple Linear Regression, Model Assessment, Cross-Validation Techniques, Model Selection, Prediction Using Regression Models. **3.Logistic Regression and Generalized Linear Models:** Logistic Response Function and Logit Transformation, Logistic Regression as a Generalised Linear Model (GLM), Interpreting Coefficients and Odds Ratios, Model Evaluation, Linear vs. Logistic Regression, Step wise regression, Advanced and Nonlinear Regression, Performance matrix. **Self-learning Topics:** Regularization techniques.	**8**	**CO2**
III	**Data Exploration and Anomaly Detection**	**1. Data Exploration:** Descriptive Statistics: Univariate and multivariate Exploration, Measure of Central Tendency, Measure of Spread, Symmetry, Skewness: Karl Pearson Coefficient of skewness. **2. Anomaly Detection**: Outliers, Causes of Outliers, Anomaly detection techniques, Outlier Detection using Statistics, Outlier Detection using Distance based method, Outlier detection using density-based methods. **Self-learning Topics:** Anomaly Detection in Real-World Datasets	**6**	**CO3**
IV	**Time Series Analysis**	**1. Overview of Time Series Analysis:** Time Series Analysis Steps, Time series Forecasting, Components of Time series, Box Jenkins Methodology and its applications. **2.Time Series Models :** Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Autoregressive Models, Moving Average Model (ARMA), Autoregressive Integrated Moving Average Model (ARIMA),Exponential smoothing, Seasonal decomposition, Forecasting accuracy (MAE, RMSE), Vector Auto Regression (VAR) and Multivariate time series, Time series feature engineering, Case studies of time series on real datasets. **Self-learning Topics:** Time series clustering	**8**	**CO4**

Draft Copy

V	**Data Visualization with Python**	1. Introduction to Data Visualization: Importance and need for visualization in analytics, Types of data visualization: Univariate, Bivariate, and Multivariate.Visualization before analysis, Role of Exploratory Data Analysis (EDA) its applications. 2. Introduction to Python for Visualization: Overview of Python for data analytics and visualization, Essential Data Libraries: NumPy, Pandas, Matplotlib, Seaborn. 3. Advanced Visualization Libraries: Plotly, Bokeh, Altair, Plotting and customizing graphs using Python. **Self-learning Topics:** Text analytics.	**6**	**CO5**
VI	**Data Visualization with R, Tools and recent Applications**	1. Introduction to R for Visualization : Data Import and Export techniques in R, Attributes and Data Types in R, Descriptive Statistics using R, Matrix and Array Operations, Exploratory Data Analysis in R, Visualizing single variable and multiple variables, Building Barplots, Histograms, Box plots, Whisker plots, and Scatterplots. 2. Visualization Tools and Dashboards: Overview of modern visualization tools like Tableau, Power BI-Building blocks, components, applications. Designing real time dashboards and integrating multiple visualizations 3. Recent Applications: Marketing analytics, finance, operations, IoT / sensor data, web analytics, manufacturing analytics and any other real trend application in industry. **Self-learning Topics:** Social media analytics	**8**	**CO6**

Draft Copy

Text Books:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data,EMC Education services Wiley Publication
2. Peter Bruce, Andrew Bruce & Peter Gedeck, Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python, O'Reilly Publications 2nd Edition.
3. Mastering Data Visualization with Tableau by Dr. Arpana Chaturvedi, Prof. Praveen Malik.
4. Gagan Gupta Raj, Naresh Kumar Nagwani, Data Analytics and Visualization,, PHI Learning.
5. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data by Hadley Wickham and Garrett Grolemund
6. Interactive Data Visualization with Python: Present Your Data As an Effective and Compelling Story, 2nd Edition by Abha Belorkar, Sharath Chandra Guntuku, and Shubhangi Hora.

References:

1. David G. Kleinbaum & Mitchel Klein, Logistic Regression: A Self-Learning Text , Springer
2. *Dr. Anil Maheshwari* , *"Data Analytics Made Accessible"* (2nd Edition, 2020) , Mcgraw Hill Education.
3. Dr. Bharti Motwani, Data Analytics using Python, Wiley India Pvt. Ltd

Online References:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/introduction-to-data-analytics?utm_source=chatgpt.com
2.	https://www.udemy.com/course/data-storytelling-mastering-data-visualization-techniques/?couponCode=PMNVD2025
3.	https://www.datacamp.com/category/data-visualization?page=1
4.	https://onlinecourses.swayam2.ac.in/imb25_mg218/preview
5.	https://www.udemy.com/course/data-analytics-masters-basics-to-advanced/?couponCode=PMNVD2025

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166113	Internet of Things	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract. / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166113	Internet of Things	20	20	40	60	2	--	--	100

Prerequisite: Basics of Microcontrollers, computer networks

Course Objectives:

1. Understand the fundamental concepts, architecture, and components of the Internet of Things (IoT).
2. Explore various communication technologies, protocols, and design principles for connected IoT devices.
3. Analyze IoT network architectures and data management frameworks involving edge, fog, and cloud computing.
4. Apply data analytics concepts to IoT applications using cloud computing paradigms.
5. Evaluate IoT privacy issues, security threats, and implement suitable protection mechanisms.
6. Develop IoT-based solutions and platforms for real-world use cases through systematic design methodologies.

Course Outcomes:

1. Explain the fundamental concepts, architecture, and components of the Internet of Things, including sensors, actuators, and IoT development boards.Apply properties of Relation sets in real-life problem-solving domains.
2. Apply suitable communication technologies and web connectivity protocols for designing efficient IoT-enabled systems
3. Analyze IoT network architectures and data management frameworks involving edge, fog, and cloud computing.
4. Demonstrate the process of data acquisition, organization, and analytics using cloud-based IoT platforms.
5. Evaluate IoT security requirements, identify vulnerabilities, and apply appropriate layered security models.
6. Design and develop IoT-based applications and platforms for real-world domains such as smart homes, cities, and agriculture.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to IoT**	**Introduction:** Definition and Characteristics of IoT **Physical Design of IoT:** Things in IoT, Overview of IoT protocols **Logical design of IoT:** Functional Blocks, Communication Models, Communication API's. **Smart Objects and IoT boards:** Sensors, Actuators and Smart Objects. Boards: Comparison of Arduino, Node MCU, ESP and Ri Pi.	6	CO1

Draft Copy

		Self-learning Topics: Pin diagrams of various sensors and boards		
II	**Design Principles for Connected Devices and Web Connectivity**	**Connected Devices:** Communication Criteria: Range, Frequency, Power, Topology. **Communication Technologies:** Wireless: NFC, RFID, BT, Zigbee, WiFi. Wired:- UART, SPI,I2C . A Comparison. **Web Connectivity Protocols:** HTTP: Limitations in constrained environment, Need for protocols in constrained environment: COAP, MQTT, XMPP, AMQP, SOAP. **Long Range Connectivity:** LoRA, NBIoT **Communication API's:** REST and Websockets **Self-learning Topics:** Application Layer protocols: HTTP, FTP, Telnet	8	CO2
III	**IoT Network Architecture and Design**	**Drivers Behind New Network Architectures:** Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support. **Architecture:** The IoT World Forum (IoTWF) Standardized Architecture: Layer 1-7, IT and OT Responsibilities in the IoT Reference Model, Additional IoT Reference Models. **A Simplified IoT Architecture:** The Core IoT Functional Stack:Layer 1, Layer 2 (only overview) Layer-3: Analytics Versus Control Applications, Data Versus Network Analytics Data Analytics Versus Business Benefits, Smart Services. **IoT Data Management and Compute Stack:** Fog Computing, Edge Computing, The Hierarchy of Edge, Fog, and Cloud **Self-learning Topics:** Brief review of applications of IoT: Connected Roadways, Connected Factory, Smart Connected Buildings , Smart Creatures etc.	8	C03
IV	**IoT Analytics for cloud**	**IoT Data Acquiring, Organizing:** Introduction**,** Data sources in IoT, data generation, Data acquisition, data validation, data categorization for storage, Data store management. **IoT Data Organization:** Database, Query Processing, SQL, NOSQL, ETL. **IoT Data Analytics Phases:** Descriptive Analytics, Predictive Analytics, Prescriptive Analytics. **Cloud computing paradigm for data collection Storage and Computing:** Introduction, Cloud computing paradigm, cloud deployment models. **Overview of cloud computing models:** IaaS**,** PaaS, SaaS. *Self-learning Topics: Cloud Services for IoT Analytics: IoT Integration and Automation using IFTTT, Blynk App*	7	CO4

Draft Copy

V	**IoT Privacy, Security and Vulnerabilities**	Introduction: Vulnerabilities, Security Requirements and Threat Analysis. IoT Security: Tomography and Layered Attacker Model, Security Models, Profiles and Protocols. **Self-learning Topics:** Study of Security issues in IoT referring reputed papers.	4	CO5
VI	**IoT platforms design Methodology and Case Studies**	**General Design Methodology (10 Steps):** Based on case study of Home automation, detail study of steps 1-4 and Overview from 5-10 **IoT Case Study:** Transportation: Connected Car and its applications, Smart City: Smart Parking, Waste Management, Street Light Control, Connected ATM premises monitoring system, Agriculture: Smart Irrigation System, Environmental Monitoring System. **Self-learning Topics:** Design of software for IoT applications	6	CO6

Text Books:

1. Arsheep Bahga (Author), Vijay Madisetti, Internet Of Things: A Hands-On Approach Paperback, Universities Press, Reprint 2020.
2. Raj Kamal, Internet of Things, Architecture and Design Principles, McGraw Hill Education, Reprint 2018.
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, IoT
4. Fundamentals Networking Technologies, Protocols, and Use Cases for the Internet of Things CISCO.
5. Andrew Minteer: Analytics for the Internet of Things (IoT) Intelligent Analytics for Your Intelligent Devices.

Reference Books:

1. Pethuru Raj, Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases by , CRC press,
2. Giacomo Veneri , Antonio Capasso,‖ Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0‖, Packt.
3. Perry Lea, Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security, Packt Publications, Reprint 2018.
4. Amita Kapoor, ―Hands on Artificial intelligence for IoT‖, 1st Edition, Packt Publishing, 2019.
5. Sheng-Lung Peng, Souvik Pal, Lianfen Huang Editors: Principles of Internet of Things (IoT)Ecosystem:Insight Paradigm, Springer

Online References:

1. NPTEL: Sudip Misra, IIT Khargpur, Introduction to IoT: https://nptel.ac.in/courses/106/1 05/106105166/
2. NPTEL: Prof. Prabhakar, IISc Bangalore, Design for Internet of Things.

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Draft Copy

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part(a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166114	Cyber Security	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166114	Cyber Security	20	20	40	60	2	--	--	100

Rationale :

This course provides foundational knowledge of computer security, network security, cybercrimes, and cyberspace regulations essential for understanding today's digital threat landscape. It equips learners with core cryptographic concepts—including symmetric keys, public key systems, and hashing techniques—critical for securing information systems. The course also builds awareness of common attack techniques. Finally, the course introduces the role of AI and ML in modern cybersecurity to enhance threat detection and cyber defense capabilities.

Course Objectives:

1. To understand the fundamentals of computer, network, and cyber security along with cybercrimes and cyberspace
2. To understand symmetric and public key cryptography along with hashing techniques and their applications
3. To understand how cyberattacks are planned and executed
4. To understand various cyber-attacks, malware techniques, and security threats
5. To understand the fundamentals of e-commerce, digital payments, security mechanisms, and regulatory guidelines.
6. To understand the role of AI and ML in modern security, threat detection, and cyber defense applications.

Course Outcomes:

1. Analyze security principles, cybercrime classifications, and the structure and regulation of cyberspace
2. Apply symmetric keys, public key algorithms, and hashing techniques to secure data and communication processes.
3. Analyze attack methods, mobile security threats, and organizational measures for securing environments.
4. Analyze different attack techniques and vulnerabilities
5. Evaluate e-commerce security practices, digital payment modes, related frauds, and RBI guidelines for secure transactions.
6. Analyze how AI-based techniques enhance threat detection, vulnerability assessment, and real-world cyber defense.

Prerequisite: Computer Networks

DETAILED SYLLABUS:

Draft Copy

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Computer Networks concepts		
I	**Introduction to security**	Defining Computer security and Network Security, CIA, Services, Mechanisms and attacks. Information Security (InfoSec) **Cyber Security:** Concept of cyber security, Issues and challenges of cyber security**,** Cybercrime definition, Cybercrime and information security, Classifications of cybercrime, a global Perspective on cybercrimes. **Cyber Space:** Cyberspace, Architecture of cyberspace, Regulation of cyberspace **Self-learning Topics:** Security Policies, Standards, and Frameworks	6	CO1
II	**Cryptography**	**Symmetric key ciphers:** Block ciphers, stream ciphers, Data Encryption standard, Advanced Encryption Standard **Public key cryptography:** Principles Public key crypto Systems, Diffie Hellman Key Exchange, the RSA algorithm, Key Management **Hashing Techniques:** SHA256, SHA-512, HMAC and CMAC. **Self-learning Topics:** Modes of Operation for Block Ciphers, Public Key Infrastructure (PKI) and Digital Certificates, ECC	**7**	CO2
III	**Cybercrime Techniques and Attack Vectors**	Types of attacks, Social Engg, Cyber stalking, Cyber café and Cybercrimes, Bot nets, Attack vector, Cloud computing, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Devices-Related Security Issues **Self-learning Topics:** Lifecycle of a Cyberattack, Real-World Case Studies	**7**	CO3
IV	**Tools and Methods Used in Cyber**	Phishing, Password Cracking, Key loggers and Spywares, Virus and Worms, Steganography, DoS and DoS Attacks, SQL Injection, Buffer Over Flow, Attacks on Wireless Networks, Phishing, Identity Theft (ID Theft). Penetration Testing Methodologies (OWASP, PTES, NIST) **Self-learning Topics:** Tools Used for Ethical Hacking, Secure Coding Practices	**5**	CO4

Draft Copy

V	**E-Commerce and Digital Payments**	Definition of E- Commerce, Main components of E-Commerce, Elements of E-Commerce security, E-Commerce threats, E-Commerce security best practices, Introduction to digital payments, Components of digital payment and stake holders, Modes of digital payments Banking Cards, Unified Payment Interface (UPI), e-Wallets, Unstructured Supplementary Service Data (USSD), Aadhar enabled payments, Digital payments related common frauds and preventive measures. RBI guidelines on digital payments and customer protection in unauthorized banking transactions. **Self-learning Topics:** Case Studies of E-Commerce Security Breaches, Fraud Detection Techniques	**7**	CO5
VI	**AI in Cyber Security**	Need and importance of AI for modern security, Data Preparation for Security:Cybersecurity data (logs, network traffic, malware samples), Feature Engineering (extracting meaningful features like packet size, API calls), and Data Labeling (creating ground truth for attacks/benign activities) The Dual Role of AI: Offensive and Defensive: Adversarial AI Attacks (Attacks at Training Time - Poisoning Attack & Attacks at Deployment Time - Evasion Attack AI for Phishing and Social Engineering Detection, NLP models to detect phishing emails/messages,AI-powered URL reputation scoring, Deepfake detection Role of ML and DL in threat detection, AI for threat intelligence and pattern mining, AI in vulnerability scanning and patch prioritization, Real-world AI applications in cyber defense. **Self-learning Topics:** Case Studies of AI Implementation in Cyber Defense, AI-Based Malware Detection, Generative AI for creating sophisticated decoys or cyberattack simulations.	**7**	CO6

Draft Copy

Text Books:

1. William Stallings, Cryptography and Network Security, Principles and Practice, Pearson Education
2. Behrouz A. Ferouzan, "Cryptography & Network Security", Tata Mc Graw Hill
3. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India
4. Advocate Prashant Mali, "Cyber Law & Cyber Crimes", Snow White Publications
5. Elias M. Awad, "Electronic Commerce", Prentice Hall of India Pvt Ltd
6. Leslie F. Sikos, "Artificial Intelligence for Cybersecurity", Springer
7. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies, Packt Publishing.

References:

1. Atul Kahate, "Cryptography and Network Security", Tata Mc Graw Hill

2. Kennetch J. Knapp, "Cyber Security &Global Information Assurance", Information Science Publishing
3. Yadav & De, AI in Cybersecurity, CRC Press

Online References:

Sr. No.	Website Name
1.	https://www.nist.gov/cybersecurity
2.	https://www.cisco.com/c/en_in/products/security/what-is-cybersecurity.html
3.	https://www.npci.org.in

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166115	Nature-Inspired Optimization Algorithms	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166115	Nature-Inspired Optimization Algorithms	20	20	40	60	2	-	--	100

Rationale:

This course introduces algorithms inspired by natural processes for solving complex optimization problems. Students will learn theoretical foundations, algorithm design, implementation, parameter tuning, and application to engineering problems. Emphasis is placed on hands-on implementation, experimental comparison, and critical evaluation.

Course Objectives:

1. To introduce the foundational principles and motivation behind nature-inspired optimization techniques.
2. To explore evolutionary algorithms and their application in multi-objective optimization problems.
3. To study swarm intelligence algorithms and their role in solving complex optimization tasks.
4. To understand novel and hybrid swarm-based optimization models and their underlying mechanisms.
5. To examine modern metaheuristic algorithms and their application in edge computing and IoT optimization.
6. To apply advanced optimization algorithms in real-world engineering and machine learning problems.

Course Outcomes: Upon successful completion, of course, learner/student will be able to

1. Explain the fundamental principles and categories of nature-inspired optimization algorithms.
2. Apply evolutionary algorithms to handle multi-objective optimization problems.
3. Analyse the working and performance of advanced swarm intelligence algorithms.
4. Implement novel swarm intelligence and hybrid optimization techniques.
5. Evaluate modern metaheuristic algorithms for optimization in Edge and IoT environments.
6. Design efficient optimization solutions for engineering and AI-based applications.

Draft Copy

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Students should be familiar with: Linear Algebra, Probability and Statistics, Fundamentals of Optimization, Computational & Programming Skills, Knowledge of search and optimization in AI	**0**	
I	**Introduction to Nature Inspired Optimization**	1.1Basic principles of optimization: motivation and need for nature-inspired algorithms, Overview of classical vs. nature-inspired optimization. 1.2 Examples of nature-based algorithms: biological, physical process inspired, Types of nature-inspired algorithms: evolutionary, swarm intelligence, bio-inspired, **Self-learning Topics:** Problem solving and Optimization: A Super-Simple optimization problem. Performance Metrics: Measuring performance, Solution Quality (Error Rate, F1 Score, Accuracy), Convergence Speed, and Robustness.	**3**	**CO1**
II	**Advanced Evolutionary Algorithms**	2.1 Introduction to Multi-Objective Optimization: Concepts of Pareto optimality, dominance, Pareto front, and trade-off analysis. methods of MOO: classical and evolutionary Evolutionary algorithm framework for MOO; population-based search for Pareto front. solution 2.2 MOEA Algorithms: NSGA-II, SPEA-II, MOEA/D, and NSGA-III mechanisms. 2.3 Case Studies & Applications: MOEAs in feature selection, neural network training, scheduling, and energy optimization. **Self-learning Topics:** Parameter Tuning: Genetic Algorithms (GA): principles, operators, representation, selection, crossover, mutation. Evolution Strategies and Evolutionary Programming. Applications and variants.	**6**	**CO2**
III	**Advanced Swarm Algorithms**	3.1Advance Algorithms: Introduction to Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO): α, β, δ, ω hierarchy guides search around prey, Whale Optimization Algorithm (WOA): Encircling prey and spiral updating mechanism. 3.2Dragonfly Algorithm: Separation, alignment, cohesion, attraction, and distraction rules. 3.3Multi-Objective Swarm Intelligence: Pareto-based PSO, Hybrid Swarm Models: PSO–GA hybrid, Firefly–ANN hybrid, ACO with local search; **Self-learning Topics:** Swarm Intelligence for AI & Edge Systems: Application in feature selection, hyperparameter tuning, federated learning, IoT & Edge resource optimization.	**8**	**CO3**

Draft Copy

IV	**Novel Swarm Intelligence Optimization Algorithm (SIOA)**	4.1 Water Wave Optimization: Objective Function, Power Balance Constraints, Generator Capacity Constraints, Water Wave Optimization Algorithm, Mathematical Model of WWO Algorithm. 4.2 Brain Storm Optimization: Multi-Objective Brain Storm Optimization Algorithm, Clustering Strategy, Generation Process, Mutation Operator, Selection Operator. 4.3 Swarm Cyborg: Swarm Intelligence Cyborg, Swarm Cyborg Taxis Algorithms, Cyborg Alpha Algorithm. **Self-learning Topics:** Application of Swarm Intelligence: Swarm Intelligence Robotics, Swarm Robotics, System-Level Properties, Coordination Mechanisms.	**6**	**CO4**
V	**Modern Algorithms**	5.1 Ant Lion Optimizer (ALO): Random walks of ants influenced by antlion traps, Moth Flame Optimization (MFO): Logarithmic spiral motion towards flames. 5.2 Harris Hawks Optimization (HHO): Dynamic switching between soft and hard besiege. 5.3 NIOA for Edge Resource Optimization - Optimization of task scheduling, load balancing, and energy consumption in Edge networks. PSO for IoT data routing, Firefly for energy-efficient edge nodes. Apply NIOA techniques for optimizing performance in distributed Edge systems. **Self-learning Topics:** Emerging Research Directions: Quantum-behaved PSO, Deep Swarm Learning, Chaotic maps, Self-adaptive swarm models.	**8**	**CO5**
VI	**Advanced Topics & Engineering Applications**	6.1 Machine Learning & Deep Learning Optimization Applications: Cost-based & Correlation-based feature selection with swarm intelligence. Predictions with reduced features, SVM and random forest, Crime forecasting with PSO-SVM. CNN and LSTM optimization. 6.2 Case Studies in Engineering: Application of metaheuristics in: Scheduling and Logistics (e.g., vehicle routing). Machine Learning (e.g., hyperparameter tuning, feature selection). **Energy Consumption Prediction** using Multi-objective GA, **Scheduling Tasks in Cloud or Social Networks** using ACO, **Stock Market Forecasting** using Swarm Intelligence. **Self-learning Topics:** Feature Subset Selection: In-depth case studies using GAs, PSO, and Binary versions of swarm algorithms to find the optimal subset of features for classification tasks.	**8**	**CO6**

Draft Copy

Books:

1. Nature-Inspired Optimization Algorithms, Xin-She Yang,
2. Introduction to Evolutionary Computing, A.E. Eiben and J.E. Smith.
3. Swarm Intelligence, James Kennedy et al.
4. Swarm Intelligence and Evolutionary Computation, Georgios N. Kouziokas.
5. Advanced Optimization by Nature-Inspired Algorithms, O. Bozorg-Haddad et al.
6. Swarm Intelligence by Kuldeep Singh Kaswan.
7. K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, 2001.
8. Carlos Coello Coello, Gary B. Lamont, David A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007
9. Seyedali Mirjalili, Evolutionary Algorithms and Metaheuristics in Engineering Applications, Springer, 2020.
10. Coello, C.A. et al., Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2019.

Online References:

Sr. No.	Website Name
1.	https://onlinecourses.nptel.ac.in/noc25_me22
2.	https://www.youtube.com/watch?v=E2dJOYDz7b0

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Draft Copy

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

.....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166116	Edge Computing	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166116	Edge Computing	20	20	40	60	2	--	--	100

Rationale :

This course provides students with essential knowledge of modern IoT and Edge Computing systems, addressing the need for real-time and low-latency processing. It introduces practical skills using Raspberry Pi, sensor interfacing, and lightweight protocols like MQTT. By studying real-world edge applications, learners become capable of designing efficient and scalable IoT-Edge solutions for industrial and smart environments.

Course Objectives:

1. To understand the basics of Cloud, Fog, and Edge Computing for low-latency IoT systems.
2. To study IoT and Edge architectures and device-to-cloud integration.
3. To learn Raspberry Pi hardware, OS setup, and remote-access tools.
4. To develop skills in sensor interfacing and lightweight protocols like MQTT.
5. To explore Raspberry Pi as an edge device for real-time processing.
6. To analyze edge computing applications and AI-enabled edge solutions.

Course Outcomes:

1. Explain Cloud, Fog, and Edge Computing concepts and the need for edge services.
2. Describe IoT–Edge architecture, communication models, and data flow.
3. Configure Raspberry Pi, install OS, use GPIO, and enable remote access.
4. Interface sensors with Raspberry Pi and implement MQTT communication.
5. Apply Raspberry Pi for edge processing in Edge–Fog–Cloud systems.
6. Evaluate edge computing applications and use AI tools at the edge.

Prerequisite:

Basic knowledge of Computer Networks, Foundational understanding of IoT concepts, Hands-on experience with programming, preferably using Python or C. Basic Linux/command-line knowledge

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Concepts of IoT, Computer Network	01	
I	**Introduction to Cloud Computing, Fog Computing And Edge Computing**	Overview of Cloud Computing, Cloud computing & its limitation to support low latency and RTT, , Introduction to Fog Computing, Introduction to Edge Computing, Why move cloud services to Edge? **Self-learning Topics:** Edge Data Centres	04	CO1
II	**IoT and Edge Architecture**	Fundamentals of IoT and Edge Architecture: Overview of IoT ecosystem, device-to-cloud communication models, role of sensors, gateways, and cloud.	06	CO2

Draft Copy

		IoT Edge Architecture Components & Data Flow: Architecture blocks: IoT devices, edge nodes, gateways, communication interfaces, cloud integration. **Self-learning Topics:** Running Docker containers on edge nodes like Raspberry Pi. Security in IoT and Edge Systems		
III	**Introduction to Raspberry Pi**	Introduction to Raspberry Pi, evolution of Raspberry Pi boards, features and capabilities of Raspberry Pi. Raspberry Pi board layout, GPIO pinouts, hardware components, interfaces, and peripheral connectivity.Installing and configuring operating systems on Raspberry Pi Connecting Raspberry Pi via SSH, using remote access tools such as VNC, PuTTY, and VS Code Remote, remote file transfer methods. **Self-learning Topics:** Overview of different Raspberry Pi models (Pi Zero, Pi 3, Pi 4, Pi 5) Basic Linux commands for Raspberry Pi users Virtual Raspberry Pi simulators (Wokwi, Pi-emulators)	06	CO3
IV	**Interfacing Raspberry Pi & Edge - to - Cloud Protocols**	IInterfacing DHT temperature –humidity sensor with Raspberry Pi, reading sensor values, data logging, and real-time monitoring. Need for lightweight protocols for IoT communication; Comparison of protocols: MQTT, CoAP, HTTP for IoT environments. Introduction to MQTT protocol and its role in IoT; Detailed MQTT architecture and workflow. MQTT Packet Structure & Data Formats. **Self-learning Topics:** Introduction to Mosquitto MQTT Broker installation on Raspberry Pi, Using Node-RED for MQTT-based IoT workflows, Cloud MQTT services (AWS IoT Core, Azure IoT Hub, Google IoT Core)	08	CO4
V	**Edge Computing With Raspberry Pi**	Fundamentals of edge computing, need for low-latency processing, comparison with cloud computing. Raspberry Pi as an edge device for data acquisition and local processing. Edge–Fog–Cloud hierarchy and deployment models. **Self-learning Topics:** Setting up Docker containers on Raspberry Pi for edge deployments, Edge security techniques (device authentication, secure boot, encrypted communication)	08	CO5
VI	**Applications of Edge Computing**	Study of edge computing use cases in smart cities, industrial automation, healthcare, transportation, agriculture, and retail. Implementation of edge nodes for video analytics, environmental monitoring, energy management, and autonomous systems. **Self-learning Topics:** Hands-On Exploration of Edge AI Tools and Platforms, Experimenting with TensorFlow Lite, NVIDIA Jetson, Google Coral, or EdgeX Foundry to develop AI-enabled edge	06	

Draft Copy

		applications.		

Text Books:

1. **Handbook of Internet of Things and Big Data** – Pethuru Raj, Anupama C. Raman
2. **Raspberry Pi User Guide** – Eben Upton & Gareth Halfacree
3. **Architecting the Internet of Things** – Dieter Uckelmann, Mark Harrison

References:

1. **Internet of Things: A Hands-On Approach** – Arshdeep Bahga & Vijay Madisetti
2. **MQTT Essentials: A Lightweight IoT Protocol** – Gastón C. Hillar
3. **Edge Computing: A Primer** – Jie Cao, Qun Li, and Philip Yu

Online References:

Sr. No.	Website Name
1.	**NPTEL — Introduction to Internet of Things:** https://nptel.ac.in/courses/106/105/106105166/
2.	**Coursera — Internet of Things and Embedded Systems** https://www.coursera.org/specializations/iot
3.	**Udemy — Raspberry Pi for Beginners & IoT Developers** https://www.udemy.com/course/raspberry-pi-full-stack/

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166117	Digital Imaging Techniques and Analysis	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2166117	Digital Imaging Techniques and Analysis	20	20	40	60	2	--	--	100

Rationale :

The course Digital Image Processing Techniques and Analysis enables students to understand and apply fundamental and advanced methods for processing, enhancing, and analyzing digital images. It bridges theory and practicality through tools like OpenCV and Python, preparing students to solve real-world problems in domains such as medical imaging, satellite image analysis, and machine vision. The course builds a foundation for higher studies and research in computer vision, artificial intelligence, and pattern recognition.

Course Objectives:

1. To introduce the fundamental concepts of digital image formation, acquisition, and representation.
2. To develop understanding of spatial and frequency domain methods for image enhancement.
3. To apply image restoration techniques in image processing.
4. To impart knowledge of segmentation and feature extraction techniques for image analysis.
5. To apply geometric and statistical approaches for image representation and classification.
6. To acquaint principles of image compression and applications of Image processing.

Course Outcomes:

1. Understand the fundamental principles of digital image formation and representation.
2. Apply image enhancement techniques for quality improvement.
3. Apply image restoration techniques to improve degraded image quality.
4. Implement Image segmentation and feature extraction techniques for analysis.
5. Represent and classify image data using geometric and statistical methods.
6. Evaluate image compression techniques and their applications in real-world systems.

Prerequisite: Mathematics for Computer Science, Computational Theory, Python Programming

DETAILED SYLLABUS:

Draft Copy

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Mathematics for Computer Science, Computational Theory, Python Programming		
I	**Introduction to Digital Imaging**	Fundamentals of digital image processing system, Components of digital imaging system, Image formation and acquisition, Sampling and quantization. Sampling and quantization, Pixel relationships, Connectivity Color models (RGB, CMY, HSI), Human visual perception and image display techniques. **Self-learning Topics:** Image file Formats	05	CO1
II	**Image Enhancement Techniques**	Spatial domain techniques: point and neighborhood processing, Histogram processing – equalization, specification, Smoothing and sharpening filters. Frequency domain enhancement – Fourier Transform, 2D-DFT and its properties,, filtering concepts, Homomorphic filtering. **Self-learning Topics:** Comparison of Enhancement techniques	07	CO2
III	**Image Restoration**	Image degradation model Restoration filters – inverse, Wiener, and adaptive filters. Geometric transformations and image registration. Motion blur and deblurring. **Self-learning Topics: I**mage reconstruction from projections	06	CO3
IV	**Image Segmentation and Feature Extraction**	Point, Line and Edge detection: Detection of isolated points, Line detection, Edge models, Sobel, Prewitt, Canny operators. Region-based segmentation: growing, splitting, merging. Morphological operations – dilation, erosion, opening, closing. Feature extraction: shape, texture, and color descriptors. **Self-learning Topics:** Thresholding techniques, Morphological watersheds	08	CO4
V	**Image Representation, Description, and Classification**	Boundary representation, chain codes, and polygonal approximation. Region descriptors: area, perimeter, moments, compactness. Principal Component Analysis (PCA). Image classification and clustering (K-Means, hierarchical). **Self-learning Topics:** Pattern recognition through case study on handwritten digits (0 - 9) from scanned images	06	CO5
VI	**Image Compression and**	Fundamentals of image compression – redundancy, fidelity criteria.	07	CO6

Draft Copy

	Applications	Lossless compression: Run-length, Huffman, LZW. Lossy compression: Transform coding (DCT), JPEG, MPEG. Emerging Image Processing applications: Case study on medical. imaging, satellite imaging, machine vision **Self-learning Topics:** Real life applications of image processing techniques		

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, _Digital Image Processing‘, Pearson Education Asia, Third Edition, 2009
2. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006
3. Milan Sonka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision, Cengage Learning, 2008
4. S. Jayaraman, E. Esakkirajan and T. Veerkumar, ―Digital Image Processing, TataMcGraw Hill Education Private Ltd, 2009

References:

1. W Pratt, “Digital Image Processing”, Wiley Publication, 3rd Edition, 2002
2. Frank Y Shish ,Image Processing and Pattern Recognition: Fundamentals and Techniques, Wiley Wiley-IEEE Press, 2010
3. Fundamentals of digital image processing, by Anil K. Jain.

Online References:

Sr. No.	Website Name
1.	**NOC: Digital Image Processing, IIT Kharagpur** Prof. Prabir Kumar Biswas https://nptel.ac.in/courses/117105135
2.	Computer Vision And Image Processing - Fundamentals And Applications By Prof. M. K. Bhuyan | IIT Guwahati https://nptel.ac.in/courses/108103174
3.	**NOC: Computer Vision, IIT Kharagpur** Prof. Jayanta Mukhopadhyay https://nptel.ac.in/courses/106105216

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

Draft Copy

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)						
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166118	Blockchain Technology	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2166118	Blockchain Technology	20	20	40	60	2	--	--	100

Course Objectives:

1. Understand the fundamentals of blockchain technology, its history, evolution, components, and structure of blocks, enabling students to differentiate blockchain from traditional databases.
2. Analyze consensus mechanisms and blockchain platforms, comparing their features, strengths, and limitations, and understand mining processes in Bitcoin and other platforms.
3. Develop and deploy smart contracts on Ethereum using Solidity, understanding the Ethereum ecosystem, accounts, transactions, and related tools.
4. Examine private and consortium blockchains, including Hyperledger Fabric, and understand permissioned networks and consensus protocols like Paxos and Raft.
5. Evaluate cryptocurrencies, tokens, ICOs, security threats, and mitigation strategies, along with practical usage scenarios and wallet management.
6. Explore emerging trends in AI integration with blockchain, including AI-enabled smart contracts, decentralized AI models, and case studies for real-world applications.

Course Outcomes:

1. Describe the Blockchain fundamentals, block structures, Merkle trees, nodes, and miners, and differentiate Blockchain from traditional databases.
2. Compare consensus algorithms (PoW, PoS, PoET, etc.), analyze Blockchain platforms (Bitcoin, Ethereum, Hyperledger, Corda), and explain mining and transaction processes.
3. Design and implement smart contracts using Solidity, demonstrate Ethereum transactions, and understand Gas, EVM, and associated tools.
4. Analyze private and consortium Blockchains, evaluate Hyperledger Fabric frameworks, and explain consensus mechanisms like Paxos and Raft.
5. Evaluate various cryptocurrencies, tokens, ICOs, wallet types, and security threats, and propose mitigation strategies for Blockchain vulnerabilities.
6. Apply AI models to Blockchain use-cases, develop AI-enabled smart contracts, and evaluate real-world case studies integrating AI with Blockchain applications.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Cryptography and system Security and Computer Network.		
I	**Introduction to Blockchain & Block Design**	**Introduction to Blockchain**: History, evolution, fundamentals concepts, components, types Key components in a Blockchain: Structure of a Block,	4	CO 1

Draft Copy

		Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Blockchain, Merkle Tree, nodes, miners **Self-learning Topics** Difference Between Blockchain and Traditional Databases, Blockchain Architecture Layers		
II	**Consensus Mechanisms and Blockchain Platforms**	**Consensus:** Byzantine Generals Problem, consensus algorithms: PoW, PoS, PoET, PoA, LPoS, pBFT, Proof-of-Burn (PoB), Life of a miner, Mining difficulty, Mining pool and its methods. **Bitcoin:** History, Common terminologies, Bitcoin Mining, propagation Script and transactions **Overview of Blockchain platforms**: Bitcoin, Ethereum, Hyperledger, Corda, Ripple Comparative analysis of platform features, Blockchain as a Service (BaaS), Case study of platform applications. **Self-learning Topics:** Hybrid models combining PoW + PoS, How platforms address transaction throughput, latency, and cross-chain communication	7	CO 2
III	**Ethereum and Smart Contracts**	**Ethereum:** History, Components, Architecture of Ethereum, Consensus, Miner and mining node, Ethereum virtual machine, Ether, Gas, Transactions,Accounts, Patricia Merkle Tree, Swarm, Whisper and IPFS, complete transaction working and steps in Ethereum. **Smart Contracts:** Introduction to smart contracts, Solidity programming: set -up tools and installation, Basics, functions, Visibility and Activity Qualifiers, Ethereum networks, solidity compiler, solidity files and structure of contracts, data types, storages, array, functions. **Self-learning Topics:**Overview of Ethereum's shift from PoW to PoS consensus, Tools for auditing and securing smart contracts	8	CO 3
IV	**Private and Consortium Blockchains**	**Introduction to Private Blockchain**: Key characteristics, need, Private vs Consortium Blockchains, Smart contracts in private Blockchain. **Hyperledger:** Introduction to Hyperledger, Tools and Frameworks, Hyperledger Fabric, Fabric Architecture, Comparison between Hyperledger Fabric & Other Technologies, Paxos and Raft consensus. **Self-learning Topics:** Real -world implementations in supply chain, banking, healthcare etc., Role of MSPs in authenticating participants in permissioned networks	7	CO 4
V	**Cryptocurrencies and digital tokens**	Cryptocurrency basics, types, usage, ERC20 and ERC721 Tokens, **ICO**: basics and related terms, launching an ICO, pros and cons, evolution and platforms, **STO**, Different Crypto currencies, Defi, Metaverse, **Cryptocurrency wallets**: Hot and cold wallets, Cryptocurrency usage, Transactions in Blockchain, UTXO and double spending problem **Common security threats in Blockchain**.	7	CO 5

Draft Copy

		Mitigating attacks: 51% attack, Sybil attack, replay attack. **Self-learning Topics:**Stablecoins and Central Bank Digital Currencies, Tokenomics and Cryptocurrency Valuation Models		
VI	**Emerging trends of AI in Blockchain and case studies**	**AI–Blockchain Integration:** Overview, need, and synergy between AI and Blockchain, benefit, AI Models on Blockchain: decentralized AI. **Case study:** AI for transaction validation, fraud detection, and performance optimization, AI-based consensus mechanisms for various applications like Supply chain, Healthcare, Energy, Education. **AI-enabled Smart Contracts:** Intelligent decision-making and automation using AI agents in smart contracts, AI in Crypto Trading, Anti-Money Laundering & Risk Scoring LLMs in Smart Contract Lifecycle (Dev, Audit, Test, Docs). **Self-learning Topics:** Case Studies on AI in Blockchain	6	CO 6

Text Books:

1. Chandramouli Subramanian, Asha A. Georgeet al., "Blockchain Technology", Universities Press, 2020.
2. Dr. Asharaf S. & Sivadas Neelima, "Blockchain Technology: Algorithms and Applications", Wiley India, 2023.
3. Andreas M. Antonopoulos & Dr. Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", O'Reilly Media..
4. Ritesh Modi, Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, Packt Publishing
5. Imran Bashir, "Mastering Blockchain: Inner Workings of Blockchain, Cryptocurrencies & Smart Contracts" ,Packt Publishing.
6. Ashwani Kumar,"Hyperledger Fabric In-Depth: Learn, Build and Deploy Blockchain Applications Using Hyperledger Fabric", BPB Publications.
7. Dr. Vaibhav Agrawal, *Practical Artificial Intelligence and Blockchain*, BPB Publications, 2021.

References:

1. Andreas M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, 2nd Edition, O'Reilly Media, Inc., 2017. ISBN: 9781491954386.
2. Dr. Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 2018.
3. Kumar Saurabh & Ashutosh Saxena, Blockchain Technology: Concepts and Applications, Wiley Publication, 2020.
4. Antony Lewis, The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them, Mango Publishing, 2018.

Online References:

Sr. No.	Website Name
1.	Solidity (smart contract language) - https://docs.soliditylang.org/
2.	web3.js (Ethereum JS library) - https://web3js.readthedocs.io/
3.	ethers.js (alt JS library for Ethereum) - https://docs.ethers.org/v5/
4.	Hyperledger Fabric (permissioned blockchain framework) - https://hyperledger-fabric.readthedocs.io/

Assessment:

Draft Copy

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _____and Course Name: .

Date of Exam: Duration: 2 Hours Max. Marks: 60

Draft Copy

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166411	**Mini Project-II**	-	4	-	-	2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I+IAT-II				
2166411	**Min Project-II**	-	-	-	-	50	--	50

Objectives:	
1	To enable students to identify real-world problems through need-based surveys, literature review, and structured requirement analysis.
2	To develop skills in formulating clear problem statements and project objectives based on feasibility, innovation, and societal relevance.
3	To train students in designing and planning project execution using tools such as Gantt charts, PERT/CPM, and structured methodologies.
4	To provide hands-on experience in developing and implementing functional prototypes or models using appropriate engineering tools and domain-specific technologies.
5	To enhance teamwork, leadership, and project management skills through collaborative group work, progress tracking, and review presentations.
6	To cultivate professional competencies in documentation, technical communication, validation, and evaluation of project outcomes with sustainability and continuous learning focus.
Outcome: Learner will be able to…	
1	Formulate a real-world problem by conducting need-based surveys and demonstrating professional responsibility in problem selection.
2	Implement appropriate engineering methods to develop a functional solution while collaboratively exhibiting ethical teamwork behaviour.
3	Demonstrate effective leadership or collaboration by actively coordinating project tasks and responding constructively to group dynamics.
4	Validate experimental results through analytical techniques and justify findings with disciplined technical documentation.

Draft Copy

5	Evaluate the societal and environmental impact of the developed solution while applying sustainable engineering practices.
6	Execute project activities using standard engineering tools while exhibiting commitment towards continuous self-directed learning.

Guidelines for Min Project-II

1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Interdisciplinary project is also allowed.
3	Students should do survey and identify needs, which shall be converted into problem statement for project in consultation with faculty supervisor/head of department/internal committee of faculties.
4	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of project.
5	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
6	Faculty supervisor may give inputs to students during project activity; however, focus shall be on self-learning.
7	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
8	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
9	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
10	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the project.

Term Work

The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of project to be evaluated on continuous basis, minimum two reviews in the semester.

In continuous assessment focus shall also be on each individual student, assessment based on individual's contribution in group activity, their understanding and response to questions.

	Distribution of Term work marks shall be as below:	**Marks**
1	Marks awarded by guide/supervisor based on logbook	5
2	Marks awarded by review committee (Average of Review 1 & Review 2)	40
	Project Review-1	

Draft Copy

	Identification of Problem		4
	Requirement analysis and Feasibility of the proposed work		4
	Literature Review		4
	Objectives of the proposed work		4
	Methodology of the proposed work		4
	Total Marks		20
	Project Review-2		
	a	Planning of project work and team structure	4
	b	Design Methodology	4
	c	Conceptual and Technical Demonstration	4
	d	Presentation: Oral delivery, contact with audience, slides, timing	4
	e	Quality of answers	4
	Total Marks		20
3	**Quality of Project report**		**5**

References to get Project ideas:

- https://www.geeksforgeeks.org/project-idea-college-network/?ref=ml_lbp
- https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-project-ideas
- https://roadmap.sh/backend/project-ideas
- https://webflow.com/blog/website-ideas
- https://gist.github.com/MWins/41c6fec2122dd47fdfaca31924647499
- https://www.projectpro.io/article/artificial-intelligence-project-ideas/461
- https://github.com/The-Cool-Coders/Project-Ideas-And-Resources
- https://nevonprojects.com/project-ideas/software-project-ideas/
- https://roadmap.sh/projects

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166119	Machine Learning & Swarm Intelligence Lab		2	-		1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166119	Machine Learning & Swarm Intelligence Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To introduce the foundational concepts of Machine Learning and its mathematical underpinnings.
2. To implement and evaluate supervised and unsupervised learning models on real-world datasets.
3. To understand neural network learning mechanisms and dimensionality reduction techniques.
4. To study the fundamental principles of Swarm Intelligence and nature-inspired computation.
5. To design and simulate key Swarm Intelligence algorithms such as PSO, ACO, Firefly, and Cuckoo Search.
6. To integrate Swarm Intelligence methods with Machine Learning models for optimization and feature selection.

Lab Outcomes:

After successful completion, students will be able to:

1. Explain the Machine Learning workflow and evaluate model performance using standard metrics.
2. Apply supervised and unsupervised algorithms for prediction, clustering, and pattern recognition.
3. Implement neural network architectures and perform dimensionality reduction using PCA.
4. Demonstrate understanding of Swarm Intelligence principles and apply them to optimization problems.
5. Design and evaluate metaheuristic algorithms such as PSO, ACO, Firefly, and Cuckoo Search.
6. Create hybrid Swarm–ML systems for real-world optimization and decision-support applications.

Draft Copy

Suggested List of Experiments:

Sr No	List of Experiments	Hrs
1.	Introduction to platforms such as Anaconda, COLAB, Study of Machine Learning Libraries and tools (Python library, tensorflow, keras,)	02
2.	Implement data preprocessing and normalization on a given dataset; split into training / testing sets.	02
3.	Apply Linear Regression and evaluate model performance using RMSE and R^2.	02
4.	Implement Logistic Regression/ Random Forest for binary classification (e.g., diabetes / spam dataset).	02
5.	Implement Support Vector Machines	02
6.	Perform clustering using K-Means and visualize cluster separation.	02
7.	Design a simple Perceptron or MLP using Python / TensorFlow; observe learning curve.	02
8.	Perform Principal Component Analysis (PCA) for dimensionality reduction and visualize results.	02
9.	Implement Particle Swarm Optimization (PSO) to minimize benchmark function (Sphere / Rastrigin).	02
10.	Implement Ant Colony Optimization (ACO) for shortest-path (TSP / graph) problem.	02
11.	Implement Firefly or Cuckoo Search algorithm for optimization task; record convergence.	02
12.	Apply PSO for feature selection in a classification model and compare results with baseline ML model.	02
13.	Comparative study of two Swarm algorithms based on convergence and accuracy.	02
14.	*Implement a Machine Learning model using fundamental principles in any programming language (C/C++/Java) without using pre-built libraries.	02

* is Compulsory Experiment

Sr No	List of Assignments (Any Two)	Hrs
01	Case Study: Applications of Machine Learning in engineering and data-driven decision systems.	02
02	Comparison of Supervised vs Unsupervised Learning Algorithms.	02
03	Comparative study of Linear Regression vs. Logistic Regression vs. SVM with suitable use-cases.	02
04	Short report on neural architectures: Perceptron vs. MLP and effect of activation functions on learning.	02
05	Critical note on the role of PCA in dimensionality reduction and impact on ML model performance.	02
06	Swarm Intelligence: Concept of self-organization, intensification vs diversification, and metaheuristic motivation.	02
07	Performance comparison of PSO, ACO, Firefly, and Cuckoo Search for optimization — factors affecting convergence.	02
08	Study on hybrid Swarm-ML feature selection / hyperparameter tuning — real-world applications.	02

Text Books:

1. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher Education
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖,
3. James Kennedy & Russell Eberhart, *Swarm Intelligence*, Morgan Kaufmann
4. Handbook of Swarm Intelligence: Concepts, Principles and Applications (Panigrahi, Shi & Lim, Springer, 2011)

References:

1. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
2. Particle Swarm Optimization by Maurice Clerc (2006)
3. Swarm Intelligence Optimization: Algorithms and Applications- Abhishek Kumar, Pramod Singh Rathore, Vicente Garcia Diaz & Rashmi Agrawal, Published by Wiley

Draft Copy

4. Shai Shalev-Shwartz, Shai Ben-David, ―Understanding Machine Learning‖, Cambridge University Press

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/machine-learning
2.	https://onlinecourses.nptel.ac.in/noc25_me22/preview
3.	https://www.coursera.org/learn/evolutionary-computation-and-its-engineering-applications

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166120	**Data Analytics and Visualization Lab**	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166120	**Data Analytics and Visualization Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. To learn data cleaning, preparation, and feature engineering using Python and R, including normalization and dimensionality reduction.
2. To develop skills in building, evaluating, and interpreting regression and logistic models with data preparation and performance assessment using Python or R.
3. To develop skills in data exploration and anomaly detection, including identifying outliers using statistical, distance-based, and density-based methods for real-time applications.
4. To develop skills in analyzing, modelling, and forecasting time series data using Python or R.
5. To develop skills in creating and interpreting static and interactive visualizations using Python libraries like Matplotlib, Seaborn, Plotly, and Dash.
6. To develop skills in creating and interpreting static and interactive visualizations in R using ggplot2, plotly, shiny, and base R functions.

Lab Outcomes:

1. Understand the fundamentals of data analytics, normalization and feature engineering and PCA to improve predictive model performance.
2. Apply and analyze different **regression and evaluate model performance using Python or R**
3. Comprehend and apply exploratory data analysis and anomaly detection techniques using Python or R.
4. Evaluate time-series data by exploring and decomposing it, fitting ARIMA models, and assessing forecast accuracy using RMSE, MAE, and MAPE.
5. Create charts, plots, and interactive dashboards to visualize data patterns and identify outliers using Python
6. Apply data visualization techniques to create and interpret charts, plots, heatmaps, and dashboards to analyze trends, distributions, and relationships in data using R and other visualization tools like power BI/Tabulea/ open source tools

DETAILED SYLLABUS:

Draft Copy

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Data mining, python Programming	
I	**Introduction to data analytics**	**1.** Overview of data analytics, Types of data: structured, unstructured, semi-structured, Introduction to Python and R for data analytics. **2.**Python Libraries: NumPy, Pandas, Scikit-learn R Libraries: dplyr, tidyr **3.** Importance of data cleaning and preparation Handling missing values and duplicates Data normalization and standardization techniques. **4.** Concept of features and importance in predictive modelling Creating new features from existing data. Feature importance and dimensionality reduction overview (PCA). **Self- learning Topics:** Different feature engineering techniques to improve model performance	LO1
II	**Regression Model Evaluation.**	**1.** Concept of dependent and independent variables. The regression line and equation. Assumptions of linear regression. Estimation of coefficients using Least Squares Method, Residual analysis, Extension of simple linear regression with multiple predictors. **2.**Model evaluation metrics: RMSE, MAE, MAPE,nonlinear regression ,Model fitting and interpretation **3.**Introduction to logistic regression for classification Logit function and odds ratio Model fitting and interpretation **Self- learning Topics**: Explore Advance Regression Model	LO2
III	**Data Exploration and Anamoly Detection**	**1. Concepts of Data Exploration:** Different techniques: EDA and feature exploration. 2.**Anomaly Detection:** Outliers, Causes of Outliers, Anomaly detection techniques, Outlier Detection using Statistics, Outlier Detection using Distance based method, Outlier detection using density-based methods for real time applications **Self- learning Topics**: Anomaly detection in high-volume data.	LO3
IV	**Time Series Analysis**	**1. Types of time series:** univariate and multivariate, Importing and exploring time series data in Python (pandas, numpy) / R (ts, xts) Extracting time series components, Visualizing decomposed components, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). **2.**Interpretation for model identification AR, MA, ARMA, and ARIMA concepts ,Fitting ARIMA models in Python (statsmodels) / R (forecast package) ,	LO4

Draft Copy

		accuracy metrics for forecasting: MAPE, RMSE, MAE for real time applications. **Self-learning Topics:** Deep learning models for time series forecasting	
V	**Data Visualization using Python**	**1.Python Libraries for Visualization :** Matplotlib, Seaborn, **2.Plotly and Dash:** interactive visualizations and dashboards, stacked bar charts , Scatter plots for relationships, Boxplots and violin plots for distribution and outliers, Interactive plots using Plotly, Histograms and density plots. **Self-learning Topics**: making Seaborn plots interactive using plotting libraries such as Bokeh	LO5
VI	**Data visualization using R and Tools like Power BI/Tableau/open source tools**	*1. Base R plotting functions: plot(), hist(), barplot(), boxplot(), ggplot2, plotly for interactive plots, shiny for interactive dashboards.* *2.Other packages: lattice, corrplot, dplyr (for data manipulation), Line plots for trend analysis, Bar charts and histograms, Pie charts and stacked bar charts, Scatter plots for analyzing relationships, Boxplots and violin plots for distributions and outliers, Heatmaps for correlation matrices, Density plots and area plots, Histograms , Interactive plots with plotly and ggplotly(): Adding tooltips, hover info, and clickable elements, Combining static (ggplot2) and interactive (plotly) visualizations.* **3.**Overview of Power BI or Tableau capabilities, types of dashboards (operational, analytical, strategic), sources and structure of business data, importing and preprocessing datasets, creating relationships between tables, and applying visualization techniques using Power BI or Tableau, or open-source tools. **Self- learning Topics:** Storytelling with dashboards	L06

Draft Copy

Text Books:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data,EMC Education services Wiley Publication
2. Peter Bruce, Andrew Bruce & Peter Gedeck, Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python, O'Reilly Publications 2nd Edition.
3. Mastering Data Visualization with Tableau by Dr. Arpana Chaturvedi, Prof. Praveen Malik.
4. Gagan Gupta Raj, Naresh Kumar Nagwani, Data Analytics and Visualization,, PHI Learning.
5. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data by Hadley Wickham and Garrett Grolemund
6. Interactive Data Visualization with Python: Present Your Data As an Effective and Compelling Story, 2nd Edition by Abha Belorkar, Sharath Chandra Guntuku, and Shubhangi Hora.

References:

1. David G. Kleinbaum & Mitchel Klein, Logistic Regression: A Self-Learning Text , Springer
2. *Dr. Anil Maheshwari* , *"Data Analytics Made Accessible"* (2nd Edition, 2020) , Mcgraw Hill Education.
3. Dr. Bharti Motwani, Data Analytics using Python, Wiley India Pvt. Ltd

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/introduction-to-data-analytics?utm_source=chatgpt.com
2.	https://www.udemy.com/course/data-storytelling-mastering-data-visualization-techniques/?couponCode=PMNVD2025
3.	https://www.datacamp.com/category/data-visualization?page=1
4.	https://onlinecourses.swayam2.ac.in/imb25_mg218/preview
5.	https://www.udemy.com/course/data-analytics-masters-basics-to-advanced/?couponCode=PMNVD2025

List of Experiments:

Note: * marked experiments are compulsory

Sr No	List of Experiments	Hrs
1	To demonstrate the use of data analytics libraries in Python and R for dataset pre-processing and analysis.	2
2	To perform data normalization, outlier detection, and categorical data encoding to improve dataset quality.	2
3	* To apply feature engineering techniques for creating, transforming, and selecting relevant features to enhance model performance.	2
4	Simple Linear Regression in Python/R.: To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
5	*Multiple Linear Regression in Python/R:To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
6	Logist Regression in Python/R:To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
7	Outlier detection using distance based/density based method.	2
8	Perform EDA on the given data set	2
9	*To Build and Evaluating an ARIMA Model for Applications of ARIMA (Examples: Economics and Finance, Healthcare, weather Forecasting)	2
10	*To perform Time Series Analysis in Python/R	2
11	To analyze and interpret data using data visualization techniques in Python	2
12	To analyze and interpret data using data visualization techniques in R	2
13	*To create an interactive dashboard using real-world business data for decision-making insights by visualization tool (Power BI/Tableau)	2

Draft Copy

Note: It is suggested to assign one assignment covering Units 1 to 3, and another assignment covering Units 4 to 6 or refer the below suggested assignments.

Sr No	List of Assignments
01	To understand and apply various pre-processing techniques to improve dataset quality for regression modelling. Select any real-world dataset (e.g., from Kaggle, UCI Repository, or Government open data).
02	To build and evaluate regression models for predictive accuracy and interpretability. (Use the pre-processed dataset from Assignment 1.)
03	A company wants to predict customer churn using attributes such as Age, Income, Tenure, Satisfaction_Score, and Usage_Frequency. Tasks: 1. Illustrate with an example how PCA reduces redundancy in features. 2. Explain how PCA can improve the performance of a machine learning model.
04	Suppose you have the monthly revenue of a telecom company for 6 years. Describe the complete process you would follow to forecast revenue for the next year
05	Compare strengths and weaknesses of different python visualization tools
06	Write a short note on :Analyzing Recent Trends in Data Analytics and Visualization Across Industry applications Applications:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166121	Internet of Things Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166121	Internet of Things Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To understand the interfacing of sensors and actuators with IoT development boards for data acquisition and control.
2. To learn how to transmit, visualize, and store real-time sensor data on cloud platforms.
3. To implement web-based and mobile-based IoT control mechanisms for automation applications.
4. To explore IoT communication protocols such as MQTT and HTTP for device-to-cloud connectivity.
5. To design and develop IoT-based applications integrating cloud platforms like ThingSpeak, Firebase, and Blynk.
6. To analyze and evaluate the performance of IoT systems for real-world use cases such as home automation and environmental monitoring.

Lab Outcomes:

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

1. Interface various sensors and actuators with IoT boards such as NodeMCU, ESP32, and Raspberry Pi.
2. Acquire and upload real-time sensor data to cloud services using appropriate IoT protocols.
3. Develop simple IoT applications to monitor and control devices via web and mobile interfaces.
4. Implement MQTT and REST-based communication for device data exchange and automation.
5. Integrate IoT systems with platforms like Thing Speak, Firebase, and Blynk for data visualization and control.
6. Design and demonstrate end-to-end IoT solutions for smart home, environment, and automation applications.

Draft Copy

Prerequisite: Knowledge of Basic sensors and Arduino Boards

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Basic introduction to Microprocessor and microcontroller programming, hands-on with C and python programming	

I	**Arduino**	Role of Arduino in IoT, Introduction to Arduino Boards (Uno, Nano, Mega), Arduino IDE Setup and Tools, Structure of Arduino Sketch (setup and loop functions) Variables, Data Types, Operators, Digital I/O: Reading from and Writing to Pins, Serial Communication and Serial Monitor Self-learning Topics: Interrupts and Power Management, Integrating Arduino with Other Platforms	**LO1**
II	**ESP32**	Overview of ESP32 microcontroller, ESP32 architecture and features, Comparison with Arduino and Raspberry Pi, Power modes and energy efficiency, Typical IoT applications using ESP32 Self-learning Topics: Interfacing Sensors and Actuators, Web and Cloud Integration	**LO2**
III	**Raspberry Pi**	Overview of Raspberry Pi models and features, Comparison with Arduino, Components of a Raspberry Pi board, Installing and configuring Raspberry Pi OS, SSH, VNC, and remote access setup, Basic Linux terminal commands, Python Programming on Raspberry Pi Self-learning Topics: Operating Systems & Customization with Raspberry Pi	**LO3**
IV	**Sensors and Actuators in IoT Applications**	Temperature sensors (LM35, DHT11, TMP36), Pressure sensors (BMP180, MPX5010), Motion and position sensors (PIR, Ultrasonic HC-SR04, IR), Light sensors (LDR, TSL2561). Sound sensors (microphone, sound detector module), DC motors, Relays, Interfacing with microcontrollers (Raspberry Pi / Arduino / ESP32) Self-learning Topics: Practical Applications and Case Studies	**LO4**
V	**Web & IoT Applications**	Setting up a **local web server** (Apache, Flask, Node.js), Interfacing IoT devices with web servers, Data storage, retrieval, and visualization techniques, Building simple dashboards for sensor monitoring, Self-learning Topics: practice with Popular IoT cloud platforms like ThingSpeak, Adafruit IO, Blynk, Google Firebase, AWS IoT Core, Azure IoT Hub etc.	**LO5**
VI	**Project Case Studies**	Development of IoT based Applications Self-learning Topics: Software development for complete IoT applications	**LO6**

Draft Copy

Text Books:

1. 21 IoT experiments, Yashawant Kanetkar, Srirang Korde, BPB Publications.
2. Developing IoT Projects with ESP 32, Vedat Ozan Oner, Packt Publishing, 2023
3. Raspberry Pi IoT Projects, J Shovic, Apress.

References:

1. "Exploring Arduino" by Jeremy Blum, John Wiley & Sons, 22 Jul 2013
2. "Arduino Cookbook" by Michael Margolis, *Michael Margolis* with Nick Weldin (O'Reilly).

Online Resources:

Sr. No.	Website Name
3.	Official Arduino documentation: https://www.arduino.cc
4.	https://thingsboard.io
3.	https://mqttlab.iotsim.io
4	https://openremote.io/
5	https://www.instructables.com/projects/
6	https://iotdesignpro.com/esp8266-projects

Sr No	List of Assignments / Tutorials
Module 1: Introduction to IoT	
01	Explain the physical and logical design of IoT with suitable examples. Illustrate the role of sensors, actuators, and IoT boards such as Arduino, NodeMCU, and Raspberry Pi in IoT systems.
02	Compare various IoT communication models and APIs. Discuss their significance in connecting smart devices.
Module 2: Design Principles for Connected Devices and Web Connectivity	
03	Differentiate between wired (UART, SPI, I^2C) and wireless (NFC, RFID, Bluetooth, ZigBee, Wi-Fi) communication technologies based on range, power, and topology.
04	Develop a short report on MQTT or CoAP protocol implementation for IoT data transfer. Include message structure and use cases.
Module 3: IoT Network Architecture and Design	
05	Draw and explain the IoT World Forum (IoTWF) Reference Architecture highlighting key layers and their responsibilities.
06	Discuss the role of edge, fog, and cloud computing in IoT data management. Provide an example illustrating their interrelation.
Module 4: IoT Analytics for Cloud	
07	Describe the phases of IoT data analytics — descriptive, predictive, and prescriptive — with a suitable example for each.
08	Compare cloud service models (IaaS, PaaS, SaaS) in the context of IoT analytics and application deployment.
Module 5: IoT Privacy, Security, and Vulnerabilities	
09	Identify and analyze common vulnerabilities in IoT systems. Suggest techniques for securing IoT devices and data using layered security models.
Module 6: IoT Platform Design Methodology and Case Studies	
10	Design an IoT solution for any one of the following case studies: Smart City (Parking or Waste Management) * Connected Car System * Smart Agriculture (Irrigation or Environmental Monitoring) Include problem definition, architecture, components used, and expected outcomes.

Draft Copy

Suggested List of Experiments
Students can **Use Any Board** such as Arduino/ Node MCU/ ESP32/ R-pi as per requirement.
Assessment:

Sr No	List of Experiments	Hrs
Prerequisite	To study and explore Tinkercad Simulator.	02
01	Interfacing a Temperature and Humidity Sensor (DHT11) with any board.	02
02	Interfacing LED/Buzzer/ Motion Sensor (PIR)/ ultrasonic sensor using a Web-Based Interface via any board.	02
03	IoT-Based Light Intensity Monitoring System using LDR and any board.	02
04	Python program for communication between clients based on MQTT.(using paho library).	02
05	Displaying Sensor Data on a Local Web Server using Flask and RPi.	02
06	Any Application of IFTTT	02
07	Cloud Data Logging using Raspberry Pi and ThingSpeak.	02
08	Interfacing Soil Moisture Sensor with any board and Uploading Data to ThingSpeak Cloud. OR on AWS cloud	02
09	Any application of Blynk App	02
10	To study and implement IoT Data processing using Pandas	02
11	To control home devices (LED) using self-hosted page on Amazon AWS.	02
12	To send email alert using ESP32/nodeMCU via SMTP server.	02

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166122	Cyber Security Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166122	Cyber Security Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To provide hands-on skills in identifying and analyzing cyberattack techniques.
2. To enable students to use modern tools for network, web, and mobile security testing.
3. To build practical knowledge of malware analysis, phishing simulation, and forensic tools.
4. To introduce AI/ML techniques for anomaly detection and cybersecurity automation.
5. To train students to secure digital payment systems, cloud systems, and social media accounts.
6. To promote ethical practices, safe online behavior, and understanding of cyber laws through practical scenarios.

Lab Outcomes:

1. Demonstrate understanding of cyberattacks through hands-on experiments.
2. Apply security tools (Wireshark, Nmap, Burp Suite, Autopsy) to analyze threats.
3. Conduct vulnerability tests on websites, networks, and applications.
4. Analyze malware, logs, and forensic artifacts using automated and AI-assisted tools.
5. Implement AI/ML models for detecting anomalies and phishing attacks.
6. Evaluate best practices for securing social media, digital payments, mobile devices, and cloud resources.

Prerequisite: Computer Network, Cryptography, Various Operating Systems, Linux, Windows, Android etc.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Introduction to Kali Linux, Python basics for AI labs, setting up a cybersecurity lab environment (VMware/VirtualBox).	-
I	**Cybercrime Techniques & Fundamental Security Tools**	Understanding cybercrime techniques using practical demos. Footprinting, scanning, and reconnaissance. Packet capture basics using Wireshark. Hands-on Social Engineering simulation. Creating a simple phishing simulation safely **Self-learning Topics:** OSINT & Recon-ng, Wireshark Filters	LO1,LO2
II	**Attack Vectors & Mobile Security**	Network scanning using Nmap. Password cracking using John the Ripper. Mobile device vulnerability assessment. Man-in-the-middle (MITM) attack demonstration. Securing mobile devices & authentication mechanisms	LO2, LO3

Draft Copy

		Self-learning Topics: Hydra basics, Android security architecture	
III	**Web, Network & Application Security**	SQL Injection on DVWA / Juice Shop, XSS, CSRF, Cookie stealing. Burp Suite hands-on - DoS simulation in a controlled environment & Steganography detection tools **Self-learning Topics:** Understanding OWASP Top 10	LO2, LO3
IV	**Digital Forensics & AI-Assisted Incident Response**	Windows & Linux log analysis, File system forensics using Autopsy, Memory forensics using Volatility, Network forensics (PCAP investigations), AI-based log anomaly detection, Malware behavior analysis sandbox **Self-learning Topics:** MITRE ATT&CK techniques, Introduction to ELK / Wazuh	LO4, LO5
V	**Social Media & Digital Payment Security**	Privacy configuration on social media platforms, Identifying fake profiles, scams & misinformation, Practical case studies on digital payment fraud, Securing UPI / Cards / Wallets, RBI guidelines practical interpretation **Self-learning Topics:** How WhatsApp / Instagram handle reports, Digital fraud case logs	LO5, LO6
VI	**AI/ML for Cybersecurity**	Building a simple ML model for anomaly detection, Email phishing classifier using Python, AI for vulnerability prediction (demo), Prompt injection attack testing on AI models, Hands-on with AI-driven cyber tools **Self-learning Topics:** Basics of data preprocessing for cybersecurity datasets, Kaggle cybersecurity datasets	LO5, LO6

Text Books:

1. William Stallings, Network Security Essentials: Applications and Standards, Pearson, 2020.
2. Nina Godbole, Cyber Security, Wiley India, 2011.
3. Soma Halder, Machine Learning for Cybersecurity, CRC Press, 2020.
4. Olivier Beg, *Hands-On Ethical Hacking and Network Defense*, Packt Publishing, 2019.
5. Chris Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*, No Starch Press, 2017.

References:

1. OWASP Foundation, OWASP Testing Guide, Open Web Application Security Project (OWASP), 2014.
2. Michael Sikorski & Andrew Honig, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press, 2012.
3. Chris Hart, Digital Payment Security: Threats and Controls, Wiley, 2019.
4. Chet Hosmer, Digital Forensics with Open Source Tools, Syngress, 2011.

Draft Copy

Online Resources:

Sr. No.	Website Name
1.	NPTEL: Practical Cyber Security for Cyber Security Practitioners: https://nptel.ac.in/courses/106104467
2.	https://tryhackme.com/
3.	https://attack.mitre.org/

List of Experiments.

Sr No	List of Experiments	Hrs
01	Wireshark-based packet analysis	2
02	Footprinting & reconnaissance using OSINT tools	2
03	Password cracking using John the Ripper	2
04	SQL Injection & XSS attacks on DVWA/Juice Shop	2
05	Burp Suite intercept & modify traffic	2
06	Mobile security & MITM demonstration	2
07	Steganography encoding & decoding	2
08	Digital forensics using Autopsy	2
09	Memory forensics using Volatility	2
10	AI-based anomaly detection model	2
11	Phishing detection ML classifier	2
12	Log analysis using ELK / open-source SIEM	2

List of Assignments (Any Two)

Sr No	List of Assignments / Tutorials	Hrs
01	Case study on cybercrime investigation	2
02	Report on social engineering attacks & prevention	2
03	Research on digital payment frauds	2
04	Documentation on securing social media accounts	2
05	Creating AI model documentation for anomaly detection	2
06	Report on cloud security basics	2

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166123	Nature- inspired Optimization Algorithms Lab	-	2	-	-	1	-	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166123	Nature-Inspired Optimization Algorithms Lab	-	-	-	-	-	25	25	50

Lab Objectives

The laboratory aims to:

1. Understand the working principles of classical and nature-inspired optimization algorithms.
2. Implement single and multi-objective evolutionary algorithms such as GA, PSO, and NSGA-II.
3. Explore swarm intelligence algorithms for solving real-world optimization and classification problems.
4. Develop hybrid and novel optimization models to enhance convergence and accuracy.
5. Apply metaheuristic algorithms to optimize machine learning and deep learning model parameters.
6. Evaluate the performance of optimization algorithms for practical engineering and AI-based applications.

Lab Outcomes

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

1. Demonstrate the implementation and working of basic nature-inspired algorithms for optimization tasks.
2. Implement and analyze multi-objective optimization algorithms like NSGA-II and MOPSO.
3. Apply swarm-based algorithms (e.g., GWO, WOA) for feature selection and classification.
4. Design and test hybrid metaheuristic algorithms (e.g., GA-PSO) for faster and more robust convergence.
5. Integrate optimization algorithms with machine learning or edge computing systems for performance improvement.
6. Assess and compare the effectiveness of advanced optimization techniques in solving complex, real-world problems.

Suggested list of experiments

Sr. No.	Experiment Title	Mapped LO
1	Explain and demonstrate the functioning of a basic nature-inspired optimization algorithm (e.g., Genetic Algorithm or Particle Swarm Optimization) for solving simple optimization problems.	LO1

Draft Copy

Sr. No.	Experiment Title	Mapped LO
2	Implementation of a Multi-Objective Genetic Algorithm (NSGA-II).	LO2
3	Multi-Objective Particle Swarm Optimization (MOPSO) for Energy-Delay Trade-off.	LO2
4	GWO algorithm to select optimal subsets of features for a supervised ML model (e.g., Logistic Regression or SVM).	LO3
5	Implementation of Whale Optimization Algorithm (WOA) for Solving Continuous Optimization Problems.	LO3
6	Design and test a hybrid GA-PSO algorithm for faster convergence on benchmark problems.	LO4
7	Apply metaheuristic optimization (GWO or PSO) for fake news or sentiment classification.	LO3, LO5
8	Application of Water Wave Optimization (WWO) Algorithm for Parameter Tuning in ML/DL Models.	LO5
9	Implementation of Swarm Cyborg Intelligence for Adaptive Optimization in Autonomous Systems.	LO5
10	Application of Ant Lion Optimizer (ALO) Algorithm for Feature Selection in Data Classification Tasks.	LO3, LO6
11	Implementation of Harris Hawks Optimization (HHO) Algorithm for Energy-Efficient Task Scheduling in Edge Computing Systems.	LO5, LO6
12	Crime Forecasting Using PSO-Based SVM Model for Predictive Policing Applications.	LO6

Books:

1. Nature-Inspired Optimization Algorithms, Xin-She Yang,
2. Introduction to Evolutionary Computing, A.E. Eiben and J.E. Smith.
3. Swarm Intelligence, James Kennedy et al.
4. Swarm Intelligence and Evolutionary Computation, Georgios N. Kouziokas.
5. Advanced Optimization by Nature-Inspired Algorithms, O. Bozorg-Haddad et al.
6. Swarm Intelligence by Kuldeep Singh Kaswan.
7. K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, 2001.
8. Carlos Coello Coello, Gary B. Lamont, David A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007
9. Seyedali Mirjalili, Evolutionary Algorithms and Metaheuristics in Engineering Applications, Springer, 2020.
10. Coello, C.A. et al., Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2019.

Online References:

Sr. No.	Website Name
4.	https://onlinecourses.nptel.ac.in/noc25_me22
5.	https://www.youtube.com/watch?v=E2dJOYDz7b0

Draft Copy

Suggested list of assignments

Assignment 1: questions should be based on first three chapters

Assignment 2 : questions should be based on chapter four to six

Term Work: Term Work shall consist of at least 10 labs based on the above list. Also, Term Work Journal must include at least 2 assignments as mentioned in above syllabus.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (experiments) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166124	Edge Computing Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166124	Edge Computing Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To set up and configure a Linux environment, execute basic commands, manage files, and automate tasks using shell scripts for edge computing applications.
2. To understand network configurations, measure and analyze network performance parameters, and troubleshoot edge-to-cloud connectivity issues.
3. To deploy virtual servers on cloud platforms, host lightweight web services using Flask/FastAPI, and simulate cloud-based task offloading for edge applications.
4. To design and simulate edge and fog computing architectures using tools like iFogSim, configure edge nodes, and implement basic task-offloading strategies.
5. To simulate IoT sensor data, implement device-to-edge communication protocols such as MQTT and HTTP/WebSockets, and deploy lightweight ML models at the edge.
6. To compare edge vs cloud performance metrics, design simple dashboards for real-time monitoring, and optimize edge applications for resource-constrained environments.

Lab Outcomes:

1. Install and configure a Linux environment, execute command-line operations, and write shell scripts to automate tasks.
2. Analyze network performance metrics such as latency, bandwidth, and packet loss, and troubleshoot connectivity issues.
3. Deploy virtual servers, host Flask/FastAPI web services, and simulate cloud compute tasks for edge offloading.
4. Design and simulate edge/fog computing environments, configure edge nodes, and implement task-offloading strategies.
5. Simulate IoT sensor data, implement MQTT and HTTP/WebSocket communication, and deploy lightweight ML models at the edge.
6. Measure edge vs cloud performance, build basic dashboards for monitoring, and optimize edge applications.

DETAILED SYLLABUS:

Draft Copy

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Knowledge of Programming, Computer Networks, Web Programming.	
I	**Linux Fundamentals for Edge Computing**	Introduction to Linux OS and distributions suitable for edge computing, Installation of Linux (desktop/server) and basic configuration, Command-line basics: executing commands, navigating directories, file handling, and permissions, Shell scripting essentials for automating tasks, Setting up a development environment for Python and IoT experimentation **Self-learning Topics:** Linux Networking Configurations and commands	LO 1
II	Networking Fundamentals for Edge Systems	Introduction to computer networks and IP addressing, Measuring network latency, packet loss, and throughput, Edge-to-cloud connectivity and network considerations, Network troubleshooting for edge devices. **Self-learning Topics: Various** Network Performance parameters	LO 2
III	Cloud Virtualization and Web Services	Creating and deploying virtual servers on cloud platforms (AWS, Azure, or GCP), Introduction to Flask and FastAPI for lightweight web services, Hosting a basic web service inside a virtual machine, Simulating cloud compute for edge offloading scenarios, Security basics for cloud-hosted services **Self-learning Topics: Exploring** various Cloud computing and Edge computing platforms	LO 3
IV	Edge and Fog Computing Architecture	Introduction to edge and fog computing concepts and use cases,,Simulating edge and fog computing environments using iFogSim or similar tools,,Configuring local machines as edge nodes,,Designing basic edge and fog architectures for IoT data processing,Task offloading strategies: local vs cloud execution **Self-learning Topics:** Comparative analysis between Cloud, Edge and Fog Computing platforms and Services	LO 4
V	IoT Integration & Communication Protocols	Simulating IoT sensor data (temperature, humidity, light intensity) with Python, Implementing MQTT with HiveMQ Cloud and Raspberry Pi, Device-to-edge communication using HTTP POST and WebSockets, Data collection, logging, and real-time transmission, Lightweight ML model deployment at the edge (moving average, anomaly detection, simple regression) **Self-learning Topics: Applications of AI in Edge Computing**	LO 5

Draft Copy

VI	Edge Analytics, Performance, and Visualization	Comparing edge vs cloud performance: latency, bandwidth usage, processing time, Data storage and basic visualization techniques (CSV, console, HTML dashboards), Building simple edge dashboards for real-time monitoring, Optimizing edge applications for resource-constrained environments, Case studies and best practices in edge and fog computing. **Self-learning Topics:** Realtime applications in Edge and Fog Computing	LO 6

Text Books:

1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 1st Edition, 2019.
2. Wei Gao & Qi Zhang, "Machine Learning for Edge Computing", Wiley, 1st Edition, 2023
3. Peter Elger & Eóin Shanaghy, "AI as a Service: Serverless Machine Learning with AWS", O'Reilly, 2020
4. Joe Papa, "TinyML Cookbook: Over 80 Practical Recipes for Deploying ML on Microcontrollers and Edge Devices", O'Reilly, 2022

References

1. Chii Chang, Satish Narayana Srirama & Rajkumar Buyya, "Fog and Edge Computing: Principles and Paradigms", Wiley, 1st Edition, 2017.
2. Deepak Gupta & Aditya Khamparia, "Fog, Edge and Pervasive Computing in Intelligent IoT-Driven Applications", Wiley, 1st Edition, 2021.

Online Resources:

Sr. No.	Website Name
1.	https://github.com/Apress/AWS-IoT-With-Edge-ML-and-Cybersecurity
2.	NPTEL: Edge Computing: https://nptel.ac.in/courses/106104449
3.	Courseera: Implementing Security in IoT and Edge Devices: https://www.coursera .org/learn/security-at-the-edge-course-2

List of Experiments.

Sr No	List of Experiments	Hrs
01	Set up a basic Linux environment: installation, command execution, directory navigation, and file handling. (Foundation needed for Edge)	2
02	Network Basics for Edge: Test IP addressing, ping nodes, check latency, and explore edge–cloud connectivity using basic commands.	2
03	To create and deploy a virtual server on AWS	
04	Host a basic Flask / FastAPI web service inside the VM (simulating cloud compute).	2
05	Simulate basic IoT sensor data (temperature, humidity, light intensity) using a Python script.	2
06	Design and deploy an edge computing architecture using edge simulators such as ifogsim	2
07	Design and deploy a fog computing architecture using simulators such as iFogSim	2
08	To Implement MQTT using HiveMQ cloud and R-Pi	2
09	Configure a local machine as an Edge Node and deploy a small edge application.	2
10	Implement a basic task-offloading mechanism using conditional logic (local vs cloud execution).	2
11	Deploy a very lightweight ML model (moving average, anomaly detection, or simple regression) at the Edge.	2
12	Compare performance metrics (latency, bandwidth usage, processing time)	2

Draft Copy

	between edge and cloud execution.	
13	Implement device-to-edge communication using HTTP POST / WebSockets.	2
14	Build a simple data visualization or logging dashboard on the edge node (CSV / console / HTML).	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1	
01	Explain the limitations of cloud computing in supporting low-latency and real-time applications. How do fog computing and edge computing address these challenges?	2
02	Describe the architecture of an IoT Edge system. Explain the roles of sensors, gateways, communication interfaces, and cloud integration in the data flow of an IoT ecosystem.	
03	Discuss the hardware architecture of Raspberry Pi, highlighting its GPIO pinout, interfaces, and supported peripherals. Explain how SSH and VNC are used for remote access and configuration.	
	Assignment 2	
01	Explain the working principle of the MQTT protocol. Discuss its architecture, workflow, and why MQTT is preferred for lightweight IoT communication compared to HTTP and CoAP.	2
02	What makes Raspberry Pi suitable as an edge computing device? Explain with reference to low-latency processing, data acquisition, and the Edge–Fog–Cloud hierarchy.	
03	Describe any two real-world applications of edge computing in domains such as smart cities, healthcare, industrial automation, or transportation. Explain how edge nodes improve system efficiency.	

Assessment :
Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166125	**Digital Imaging Techniques and Analysis Lab**	**-**	**2**	**-**	**-**	1	**-**	**1**

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166125	**Digital Imaging Techniques and Analysis Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. To introduce students to image acquisition, representation, and basic operations using OpenCV.
2. To develop skills in applying spatial and frequency domain techniques for image enhancement.
3. To provide practical experience in noise modeling, filtering, and image restoration.
4. To apply segmentation, morphological, and feature extraction methods for image analysis.
5. To understand and evaluate lossy and lossless image compression techniques.
6. To solve real-world problems using digital image processing techniques.

Lab Outcomes:

1. Perform image acquisition, representation, and basic processing using OpenCV.
2. Apply spatial and frequency domain techniques to enhance and restore images.
3. Analyze and reduce different types of noise using appropriate filtering methods.
4. Implement segmentation, morphological, and feature extraction techniques for image analysis.
5. Apply and compare lossy and lossless image compression techniques.
6. Solve real-world problems in domains such as medical imaging, satellite imaging, and machine vision using image processing methods.

Draft Copy

List of Experiments.

Sr No	List of Experiments	LO Mapping
01	Study of image acquisition and basic operations using OpenCV.	LO1
02	Implementation of image enhancement using point processing techniques.	LO2
03	Perform histogram equalization and contrast stretching on grayscale images.	LO2
04	Apply smoothing and sharpening spatial filters.	LO2
05	Implement frequency domain filtering using Fourier Transform.	LO2
06	Study and remove noise using mean, median, and adaptive filters.	LO2, LO3
07	Perform geometric transformations: scaling, rotation, and translation.	LO3
08	Apply edge detection operators (Sobel, Prewitt, Canny).	LO4
09	Implement image segmentation using thresholding (global and adaptive).	LO4

10	Perform morphological operations (dilation, erosion, opening, closing).	LO4
11	Extract color, texture, and shape features from images.	LO4
12	Implement image compression using Huffman coding / RLE /DCT.	LO5,
13	Apply lossy and lossless compression techniques and compare performance.	LO5,LO6
14	Apply a complete image analysis pipeline to real-world data (Cat and dog classification, License Plate Number recognition, Facial expression and emotion recognition).	LO6

Online Resources:

Sr. No.	Website Name
1.	Virtual Lab https://cse19-iiith.vlabs.ac.in/List%20of%20experiments.html
2.	https://nptel.ac.in/courses/117105135
3.	https://nptel.ac.in/courses/108103174
4.	https://nptel.ac.in/courses/106105216

Sr No	List of Assignments / Tutorials	CO Mapping
01	Assignment covers the topics from the first three units.	CO1,CO2,CO3
02	Assignment covers the topics from the last three units.	CO4,CO5,CO6

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166126	Blockchain Technology Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166126	Blockchain Technology Lab	--	--	--	--	25	25	25

Lab Objectives:

1. Set up local and testnet blockchain environments.
2. Develop secure smart contracts using Solidity.
3. Deploy contracts on Ethereum and interact using Web3 frameworks.
4. Deploy chain code on Hyperledger Fabric.
5. Build a full-fledged DApp or cross-chain prototype.
6. Explore Layer-2 scaling or interoperability bridges.

Lab Outcomes:

1. To install, configure, and operate a local blockchain development environment using tools such as Ganache, Truffle, and Hardhat to compile, deploy, and test basic smart contracts.
2. To develop and analyze Solidity smart contracts using OpenZeppelin libraries, applying secure coding principles to identify and prevent vulnerabilities like reentrancy and unauthorized access.
3. To create and deploy ERC-20/721/1155 tokens on Ethereum testnets (Goerli/Sepolia) and verify smart contracts using tools such as Etherscan and Web3/Ethers libraries.
4. To design, deploy, and evaluate chaincode in Hyperledger Fabric by creating networks, channels, and endorsement policies, and performing invoke/query operations using CLI and SDK.
5. To build and integrate a blockchain-enabled front-end application using React and Web3/Ethers.js, enabling wallet connectivity, smart contract interactions, and event handling.
6. To design and develop a complete blockchain application including smart contract/chaincode, UI integration, deployment, documentation, and GitHub repository.

Draft Copy

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Computer Network: Concepts like P2P networks, IP addressing, ports, client-server architecture. Cryptography and security: Hash functions, public/private keys, digital signatures, encryption.	
I	Local Blockchain Environment Setup & Basics	● Install & configure development tools: Node.js, VS Code, Git ● Set up local blockchain: Ganache (GUI + CLI), Truffle initialization, Hardhat project setup ● Compile, deploy, and test a simple "Hello Blockchain" smart contract ● Write basic unit tests using Truffle/Hardhat	LO 1

		• Interact with local blockchain using scripts (JS or Python) **Self-learning Topics:** Deploy and Compare local blockchain simulators: Ganache vs Hardhat Node vs Anvil (Foundry).	
II	**Smart Contract Development with Solidity and OpenZeppelin**	• Solidity basics: state variables, functions, modifiers, events, mappings • Inheritance, libraries, interfaces • Using OpenZeppelin for secure contract development • Implement access control & secure coding practices • Write unit tests to avoid common vulnerabilities like Reentrancy : Integer overflow/underflow and Unauthorized access **Self-learning Topics:** Study how AI tools like ChatGPT, CodeQL are now used for smart contract audits.	LO 2
III	**Token Development & Deployment on Testnets**	• Create ERC-20, ERC-721, ERC-1155 tokens using OpenZeppelin • Mint & transfer tokens locally • Connect to Ethereum Testnets: Goerli/Sepolia • Use Infura/Alchemy for RPC access • Deploy via Truffle/Hardhat to testnet • Interaction with Web3.js or Web3.py scripts • Verify contracts on Etherscan **Self-learning Topics:** Explore the Automated NFT metadata generation using AI (e.g., DALL-E, Midjourney) and upload to IPFS.	LO 3
IV	**Hyperledger Fabric – Permissioned Chaincode Development**	• Install Fabric binaries & prerequisites (Docker, Go, Fabric Samples) • Launch local Fabric network using test-network • Create channels, peers, and certs • Write chaincode in Go/Node.js • Invoke & query chaincode via CLI and SDK • Endorsement policies & private data collections **Self-learning Topics:** Explore the Hyperledger Aries & Indy for Decentralized Digital Identity.	LO 4
V	**DApp Integration & Frontend (React/Node + Web3/Ethers)**	• Connect front-end with blockchain using Ethers.js/Web3.js • Wallet connection using MetaMask • Build React UI to interact with smart contracts • Read/write blockchain data from UI • Handle events & notifications **Self-learning Topics:** Build a chatbot-based DApp assistant using AI (LLM) that reads contract ABI & guides users how to use the DApp	LO 5
VI	**Capstone Mini-Project**	• Build a complete blockchain project using tools learned. • Students choose any one track:	LO 6

Draft Copy

		● Ethereum Dapp, NFT/Metaverse App Hyperledger Enterprise App and Layer-2 Dapp ● Deliverables: ● Working project with UI & smart contract/chaincode, GitHub repo and Report **Self-learning Topics:** Explore and Implement AI in blockchain projects	

Text Books:

1. Andreas M. Antonopoulos and Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media.
2. Elad Elrom, The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects, Apress.
3. Mayukh Mukhopadhyay, Ethereum Smart Contract Development: Build Blockchain-based Decentralized Applications Using Solidity, Packt Publishing.
4. Xun (Brian) Wu, Zhihong Zhang, and Liang Wang, Blockchain and Ethereum Smart Contract Solution Development: DApp Programming with Solidity, Apress.
5. Samanyu Chopra, Blockchain Developer's Guide: Develop Smart Applications with Blockchain Technologies – Ethereum, JavaScript, Hyperledger Fabric, and Corda, Packt Publishing.

References:

1. Roberto Infante, Building Ethereum DApps, Manning Publications.
2. Narayan Prusty, Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity, Packt Publishing.

Online Resources:

Sr. No.	Website Name
1.	Solidity (smart-contract language) — official docs: https://docs.soliditylang.org/
2.	OpenZeppelin (library for secure smart contracts) — docs: https://docs.openzeppelin.com/
3.	Hardhat (Ethereum dev tool) — tutorial: https://hardhat.org/tutorial
4.	Hyperledger Fabric (permissioned blockchain) — chaincode dev: https://hyperledger-fabric.readthedocs.io/en/release-1.4/chaincode4ade.html

List of Experiments.

Sr No	List of Experiments	Hrs
01	Write a program for Creating a block in Blockchain using Java/Python	2
02	Installation & Setup of Local Ethereum Development Environment: Install Node.js, VS Code, Git, MetaMask, Truffle, Hardhat, Ganache. Create first Hardhat project & compile a simple contract.	2
03	Deploy & Interact with Smart Contract on Local Blockchain: Deploy a sample contract using Truffle/Hardhat + Ganache. Write scripts to read/write data. View transactions in Ganache.	2
04	Build a Basic Solidity Smart Contract (CRUD Contract) : Functions, modifiers, events, data types, mappings. Test using Hardhat tests.	2
05	Secure Smart Contract Development using OpenZeppelin: Add Ownable, Pausable, AccessControl; prevent reentrancy & data leaks.	2
06	Smart Contract Testing & Debugging: Unit tests (Chai/Mocha), coverage report, Hardhat console debugging, fixing security issues.	2
07	Create & Test ERC-20 Token: Write ERC-20 token contract using OpenZeppelin; mint, transfer, burn tokens locally.	2
08	Create & Mint ERC-721 NFT: Create NFT with metadata/IPFS; mint NFT; view on OpenSea Testnet.	2

Draft Copy

09	Deploy Smart Contracts on Testnet (Goerli/Sepolia): Connect via Infura/Alchemy, verify on Etherscan, interact using Web3.js or Web3.py.	2
10	Hyperledger Fabric Network Setup & Chaincode Deployment: Install Fabric samples, set up test network, deploy sample asset chaincode.	2
11	Build a Simple DApp: Frontend + Smart Contract Interaction	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1 (Any two Questions from each Unit)	
01	**Unit I: Introduction to Blockchain & Block Design** **Q1:** Explain the structure of a blockchain block and the role of the block header. **Q2:** What is a Merkle Tree, and why is it used in blockchain? **Q3:** Differentiate between a blockchain and a traditional database.	1
02	**Unit II: Consensus Mechanisms & Blockchain Platforms** **Q1:** Describe the Proof-of-Work (PoW) consensus algorithm. **Q2:** Compare two blockchain platforms of your choice in terms of features and consensus mechanisms. **Q3:** What is a 51% attack, and how can it affect blockchain security?	1
03	**Unit III: Ethereum & Smart Contracts** **Q1:** What is the role of Gas in Ethereum transactions? **Q2:** List the basic components of a Solidity smart contract. **Q3:** Explain the working of an Ethereum transaction from sender to miner validation.	1
	Assignment 2 (Any two Questions from each Unit)	
01	**Unit IV: Private and Consortium Blockchains** **Q1:** Differentiate between private and consortium blockchains. **Q2:** What is the role of MSPs in Hyperledger Fabric? **Q3:** Briefly explain Paxos and Raft consensus mechanisms.	1
02	**Unit V: Cryptocurrencies and Digital Tokens** **Q1:** What are the differences between ERC-20 and ERC-721 tokens? **Q2:** Explain the double spending problem in blockchain. **Q3:** List two common security threats in blockchain and their mitigation strategies	1
03	**Unit VI: Emerging Trends of AI in Blockchain and Case Studies** **Q1:** How can AI be integrated with blockchain for fraud detection? **Q2:** What are AI-enabled smart contracts, and how do they enhance automation? **Q3:** Give one example of a real-world application where AI-based consensus is used.	1

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Draft Copy

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Draft Copy

Sd/-

Dr. Subhash K. Shinde
BoS-Chairman-Computer Engineering
Faculty of Technology

Sd/-

Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-

Prof. Shivram S. Garje
Dean
Faculty of Science & Technology