Shree Warana Vibhag Shikshan Mandal's

WARANA UNIVERSITY,
WARANANAGAR

(A State Public University established under Section 3 (6) of MPUA, 2016)

|| विद्या सर्वस्य भूषणम् ||

Established:2025

Structure & Syllabus For
First Year Master of Technology (F.Y. M.Tech)
In
Computer Science Engineering

UNDER

Faculty of Science & Technology

(As Per National Education Policy – 2020)

With Effect from Academic Year 2025-26 Onwards

Shree Warana Vibhag Shikshan Mandal's

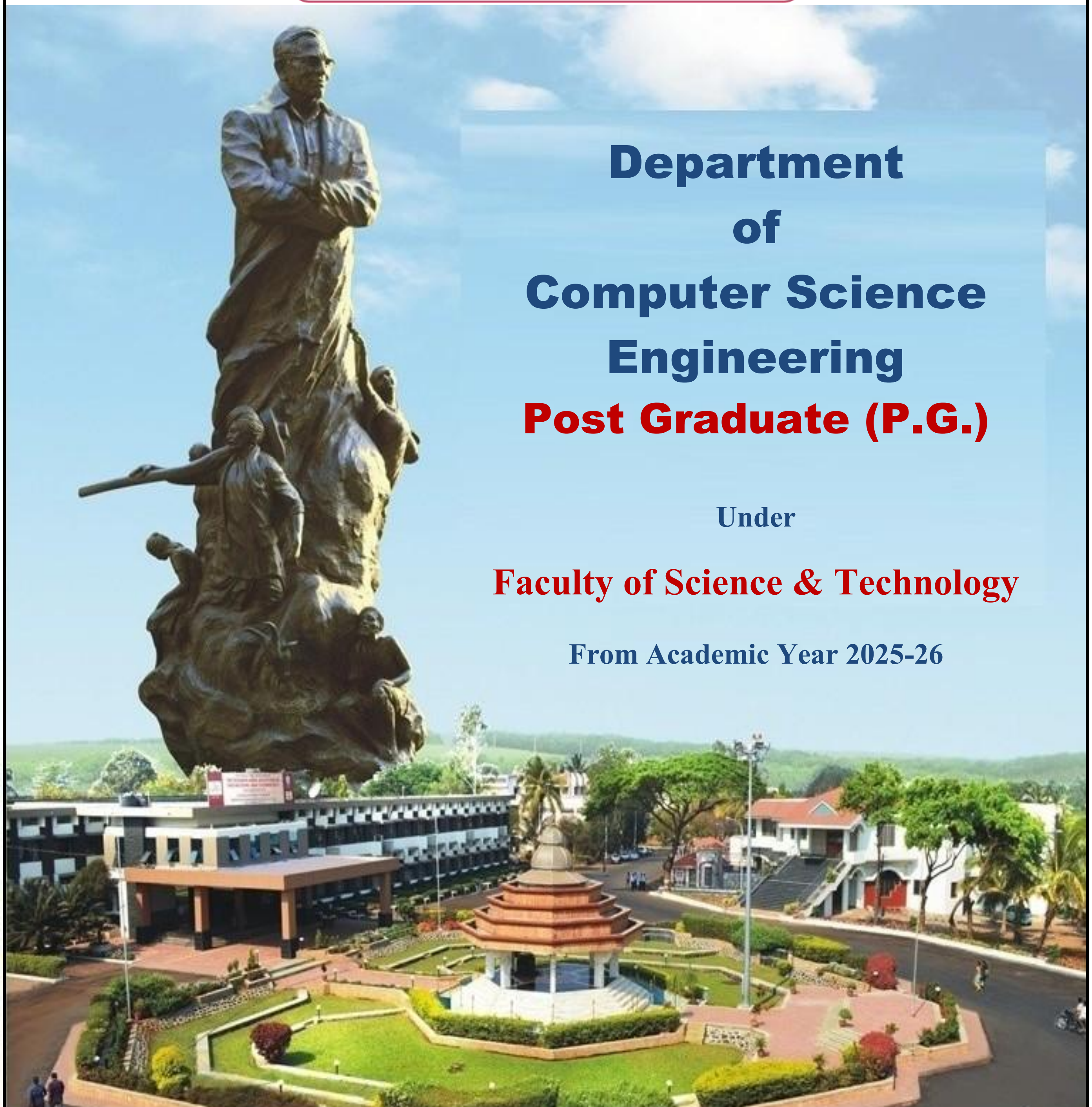
TATYASAHEB KORE INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), WARANANAGAR, KOLHAPUR

Estd. 1983

Lead Institute of

WARANA UNIVERSITY, WARANANAGAR

(A State Public University)



Department of Computer Science Engineering

Post Graduate (P.G.)

Under

Faculty of Science & Technology

From Academic Year 2025-26

M. Tech. in Computer Science Engineering

Structure and Syllabus under Autonomy as per NEP Policy 2020

Contents

Preface

The National Education Policy (NEP) 2020 has introduced significant reforms in India's higher education landscape with a strong focus on multidisciplinary education, flexibility in learning, research orientation, skill development, and industry relevance. In alignment with these progressive reforms, Tatyasaheb Kore Institute of Engineering & Technology (TKIET), Warana University, Warananagar, is committed to implementing the vision of NEP 2020 to cultivate competent, innovative, and ethically responsible professionals capable of addressing emerging technological challenges.

The Department of Computer Science and Engineering is pleased to present the Structure and Syllabus of the M. Tech Programme in Computer Science and Engineering. The curriculum has been carefully designed to provide students with a strong foundation in theoretical and applied aspects of advanced computing. The programme includes core courses such as Mathematical Foundations of Computer Science, Feature Engineering and Machine Learning, Deep Learning, and Parallel Algorithms, which strengthen the fundamental concepts required for modern computing systems and intelligent applications. In addition to the core courses, the programme offers a diverse set of program electives covering emerging areas such as network engineering and management, natural language processing, distributed systems and cloud computing, blockchain technology, pattern recognition, quantum computing, and cybersecurity. These courses enable students to explore specialized domains and develop expertise in cutting-edge technologies relevant to academia, research, and industry.

This syllabus provides comprehensive information regarding the course structure, credit framework, learning outcomes, evaluation mechanisms, and academic guidelines to ensure effective implementation of the programme. The Department sincerely acknowledges the valuable contributions of the Board of Studies members, faculty experts, and industry professionals whose insights and expertise have been instrumental in shaping this curriculum.

We are confident that the M. Tech Programme in Computer Science & Engineering, designed in accordance with NEP 2020, will equip students with advanced technical knowledge, research capabilities, and lifelong learning skills. The programme aims to prepare future-ready professionals who will contribute significantly to technological innovation, industry growth, and the digital transformation of society.

Program Outcomes

Program Outcomes (POs) are clear, measurable statements that describe what students are expected to know, understand, and be able to do by the time they complete an academic program. They define the competencies, skills, and professional abilities that graduates should possess at the end of the program. In India, POs for Engineering Programs are formally prescribed and monitored by the National Board of Accreditation (NBA). NBA has defined the following three POs for a graduate of PG Engineering Program:

PO1:	An ability to independently carry out research /investigation and development work to solve practical problems.
PO2:	An ability to write and present a substantial technical Report/document.
PO3:	Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

Duration

- The full time M. Tech Program is a 2 years post graduate program.
- The program is divided into 4 semesters.

Eligibility

1. The Candidate should be an Indian National.

2. Passed Bachelor's Degree in the relevant field of Engineering & Technology from AICTE or Central or State Government approved institutions or equivalent, with at least 50% marks (at least 45% marks in case of candidates of Backward Class categories, EWS and PWD).

3. Obtained Qualified score or non-qualified score in GATE conducted by the IIT for the current academic year.

OR

3. For sponsored candidates (Proforma P and Q), a minimum of two years of full-time work experience in a registered firm/ company/ industry/ educational and/ or research institute/ any Government Department or Government Autonomous Organization in the relevant field in which admission is being required.

Medium of Instruction

- The medium of instruction, examinations, assignments, and project reports is English.

Abbreviations

Acronym	Full Form
ISE	In-Semester Examination
ISE-I	In-Semester Examination I
ISE-II	In-Semester Examination II
ESE	End Semester Examination
ISA	In Semester Assessment
POE	Practical Oral Examination
L	Theory Lecture
T	Tutorial
P	Practical
C	Number of Credits
CH	Contact Hours
PCC	Program Core Course
PE	Program Elective Course
LC	Lab Course
OE	Open Elective Course
SW	Seminar Work
CV	Comprehensive Viva

Examination & Evaluation Pattern

Evaluation tools used for the evaluation of a student for each course is as follows:

For Theory Courses	In-Semester Examination (ISE) And End Semester Examination (ESE)
For Lab / Tutorial Courses	In-Semester Assessment (ISA) And / Or Practical and Oral Examination (POE)

Refer course structure for specific evaluation tools used for each course.

In-Semester Examination (ISE)

The In-Semester Examination (ISE) will be conducted at the departmental level. There will be two tests in each semester for every theory course: ISE-I and ISE-II.

- Each test will be of 40 marks.
- The duration of each test will be 1 hour and 30 minutes.

The total ISE marks will be calculated as the average of ISE-I and ISE-II. These rules may be modified from time to time as per the guidelines of the concerned regulatory authorities.

- ISE-I will cover Unit I and Unit II.
- ISE-II will cover Unit III and Unit IV.

  - **Minimum Passing Criteria**

    Students must score a minimum of 40% marks in the ISE. If a student fails to secure the minimum required marks, he/she must appear for a Make-up Examination.

    The Make-up Examination will be conducted in the same semester for:

    - Students who fail to secure minimum passing marks.
    - Students who were absent due to valid reasons such as medical issues, natural calamities, or participation in NSS, NCC, or similar activities (subject to verification of absence and recommendation from the Head of Department).

- **Special Provision**

  If a failed student appears for three tests (including the Make-up test) and scores more than 16 marks when calculating the average of the best two out of the three tests, the student will be awarded the minimum passing marks of 16 only.

  For students absent with valid reasons:

  - If absent in one test, the average of the attempted test and the Make-up test will be considered.
  - If absent in two tests, the decision will be taken after reviewing the reasons and based on the recommendation of the Head of Department.

End Semester Examination (ESE):

The End Semester Examination (ESE) will be conducted for 60 marks and will be based on the entire syllabus. The duration of each examination will be 2 hours.

Weightage of Units

The weightage of units in the ESE question paper will be as follows:

a) Units that are not covered in ISE-I or ISE-II will carry 30% weightage each.

b) Units that are covered in ISE-I and ISE-II will carry 10% weightage each.

Backlog Examination

Students who fail in the End Semester Examination (ESE) of either the odd or even semester within an academic year will be allowed to appear for the Backlog Examination, which will be conducted along with the regular ESE of the respective semester.

- **Re-Examination of ESE**

  A Re-Examination (Make-up Examination) for all courses (UG and PG), including both theory and laboratory courses, will be conducted once a year before the commencement of the odd semester of the next academic year.

  - A one-grade penalty will be applied to students appearing for the Make-up/Re-Examination.
  - However, no grade penalty will be applied if a student secures a ‘P’ grade in the Make-up/Re-Examination.
  - Grace marks will not be awarded for the Make-up/Re-Examination.
  - Exception: Grace marks may be considered if the student is appearing for the ESE for the first time.

- **Eligibility Criteria for ESE**

  To be eligible for the End Semester Examination (ESE), a student must:

  - Secure at least 40% marks in ISE and ISA of the concerned course.
  - Fulfil the attendance requirements as per the norms of Warana University, Waranangar.

  If a student does not meet these requirements, he/she will not be eligible to appear for the ESE.

Nature of Question Paper for ESE

Q. No.			**Marks**	**BL**	**CO**
1	**Attempt the following.**		**24**		
	a	**Unit -1**		II	1
	b	**Unit -2**		III	2
	c	**Unit -3**		IV	1
	d	**Unit -4**		I	1
2	**Attempt any Two of the following.**		**18**		
	a	**Unit -5**		VI	2
	b	**Unit -5**		II	3
	c	**Unit -5**		IV	3
3	**Attempt any Two of the following.**		**18**		
	a	**Unit -6**		IV	4
	b	**Unit -6**		III	4
	c	**Unit -6**		III	4

In Semester Assessment (ISA):

ISA for laboratory courses will be conducted as a continuous assessment throughout the semester. The assessment will be based on the following:

1. Performance in laboratory work.
2. Submission of experiments in the form of a properly maintained journal or report.
3. Timely completion of assigned experiments.
4. Attendance in laboratory sessions.

5. Understanding of the experiments conducted, evaluated through methods such as quizzes, oral examinations, case studies, field work, surveys, open-book tests, model preparation, programming, projects, or any other criteria specified by the course teacher.

Practical Oral Examination (POE):

POE for laboratory courses will be conducted immediately after the end of the semester. The duration of the practical examination will be as specified in the curriculum structure. The POE will be conducted jointly by an Internal Examiner and an External Examiner.

The examination may be conducted in any one of the following ways:

1. **Oral Examination Only**

   Both the Internal and External Examiners will ask questions based on the practical content of the course to assess the student's practical knowledge.

2. **Practical Examination Only**

   Students will be required to perform a given experiment, complete a workshop task, prepare a drawing, or develop a computer program, as applicable. In this case, the student's performance will be evaluated by the External Examiner only.

3. **Practical and Oral Examination**

   Students will first perform a given practical task. This will be followed by an oral examination (viva voce) based on the practical content of the course. The student's performance will be evaluated jointly by both the Internal and External Examiners.

Grading System

The University follows a **10-Point Grading System** to evaluate student performance.

- **Conversion of Marks into Grades**
  In every semester, the marks you get in each subject (out of 100) are converted into **grade points** as per the table below. You need at least **40% marks** to pass a subject.

Marks Obtained (Out of 100)	Grade Point	Letter Grade	Meaning
Absent	0	AB	Absent
0 – 39	0	F	Fail
40 – 44	4	P	Pass
45 – 49	5	C	Average
50 – 59	6	B	Above Average
60 – 69	7	B+	Good
70 – 79	8	A	Very Good
80 – 89	9	A+	Excellent
90 – 100	10	O	Outstanding

Note:

1. If decimal marks are 0.5 or more, they will be rounded off to the next higher number. (Example: 59.5 will become 60)
2. For courses of 50 marks or 200 marks, marks will be converted proportionally to 100 marks before assigning grade points.

- **Calculation of Semester Grade Point Average (SGPA)**
  SGPA is calculated at the end of each semester**.** It shows your average performance in one semester.

$$SGPA = \frac{\sum(Credit \times Grade\ Point)\ for\ each\ course\ of\ a\ Semester}{\sum(Credits)\ for\ a\ Semester}$$

- **Calculation of Cumulative Grade Point Average (CGPA)**
  CGPA is calculated after completing multiple semesters. CGPA reflects the overall academic performance of the student in the program.

$$CGPA = \frac{\sum(Total\ Credits\ of\ a\ Semester \times SGPA\ of\ Respective\ Sememster)\ of\ all\ semesters}{\sum(Course\ Credits)\ of\ all\ Semesters}$$

Note:

1. The SGPA & CGPA shall be rounded off to 2 decimal points.

First Year M. Tech. Computer Science Engineering

Curriculum Structure & Evaluation Scheme for Semester-I

Course Category	Course Code	Course Title	Teaching and Credit Scheme					Examination and Evaluation Scheme			
			L	T	P	C	CH	Component	Marks	Min for Passing	
PCC	2501PCSE PCC101	Mathematical Foundation of Computer Science	3	-	-	3	3	ESE	60	24	40
								ISE	40	16	
	2501PCSE PCC101T	Mathematical Foundation of Computer Science Tutorial	-	1	-	1	1	ISA	25	10	10
	2501PCSE PCC102	Feature Engineering and Machine Learning	3	-	-	3	3	ESE	60	24	40
								ISE	40	16	
	2501PCSE PCC102T	Feature Engineering and Machine Learning Tutorial	-	1	-	1	1	ISA	25	10	10
PE	2501PCSE PE103X	Program Elective-I	3	-	-	3	3	ESE	60	24	40
								ISE	40	16	
	2501PCSE PE104X	Program Elective-II	3	-	-	3	3	ESE	60	24	40
								ISE	40	16	
	2501PCSE PE105X	Program Elective-III	3	-	-	3	3	ESE	60	24	40
								ISE	40	16	
LC	2501PCSE LC106P	Lab Course	-	-	4	2	4	POE	25	10	20
								ISA	25	10	
SW	2501PCSE SW107T	Seminar-I	-	-	2	1	2	ISA	50	20	20
Total			15	2	6	20	23		650	260	260

Note: 'X' indicates the sequence number of PE/OE offered by the respective department.

First Year M. Tech. Computer Science Engineering

List of Program Electives for Semester-I

	Course Code	Course Title
Program Elective-I	2501PCSEPE1031	Network Engineering and Management Process
	2501PCSEPE1032	Linear Algebra
	2501PCSEPE1033	Optimization Techniques
Program Elective-II	2501PCSEPE1041	Semantic Web
	2501PCSEPE1042	Natural Language Processing
	2501PCSEPE1043	Distributed System & Cloud computing
Program Elective-III	2501PCSEPE1051	Block-chain Technology
	2501PCSEPE1052	Smart Technology and Internet of Things
	2501PCSEPE1053	High Dimensional Data Analysis

First Year M. Tech. Computer Science Engineering

Curriculum Structure & Evaluation Scheme for Semester-II

Course Category	Course Code	Course Title	Teaching and Credit Scheme					Examination and Evaluation Scheme			
			L	T	P	C	CH	Component	Marks	Min for Passing	
PCC	2501PCSEPC C201	Deep Learning	3	--	--	3	3	ESE	60	24	40
								ISE	40	16	
PCC	2501PCSEPC C201T	Deep Learning Tutorial	--	1	--	1	1	ISA	25	10	10
PCC	2501PCSEPC C202	Parallel Algorithms	3	--	--	3	3	ESE	60	24	40
								ISE	40	16	
PCC	2501PCSEPC C202T	Parallel Algorithms Tutorial	--	1	---	1	1	ISA	25	10	10
PE	2501PCSEPE2 03X	Program Elective-IV	3	--	--	3	3	ESE	60	24	40
								ISE	40	16	
PE	2501PCSEPE2 04X	Program Elective-V	3	--	--	3	3	ESE	60	24	40
								ISE	40	16	
OE	2501PCSEOE 205X	Open Elective Course	3	--	--	3	3	ESE	60	24	40
								ISE	40	16	
LC	2501PCSELC 206T	Lab Course	--	--	2	1	2	ISA	25	10	10
SW	2501PCSESW 207T	Seminar-II	--	--	2	1	2	ISA	50	20	20
CV	2501PCSECV 208P	Comprehensive Viva	--	--	2	1	2	POE	25	10	10
Total			**15**	**2**	**6**	**20**	**23**	**--**	**650**	**260**	**260**

Note:

- 'X' indicates the sequence number of Program Elective (PE) offered by Computer Science and Engineering Program.
- Students should opt for the Open Elective (OE) course from other departments. The list of OE courses offered by other departments is available in the structure. Although the OE course code is defined by the respective program in the structure, the actual opted OE course will appear on the mark card.

First Year M. Tech. Computer Science Engineering

List of Program Electives for Semester-II

	Course Code	Course Title
Program Elective-IV	2501PCSEPE2031	Cryptography and Network Security
	2501PCSEPE2032	Human and Computer Interaction
	2501PCSEPE2033	Parallel Computing
Program Elective-V	2501PCSEPE2041	Quantum Computing
	2501PCSEPE2042	Pattern Recognition
	2501PCSEPE2043	Deep Generative Model

First Year M. Tech. Computer Science Engineering

List of Open Electives Offered by All Programs

Sr. No.	OE Offered by Program	Course Code	Open Elective Course
1	Chemical Engineering	2501PCHEOE2051	Project Management
2		2501PCHEOE2052	Operations Research
3		2501PCHEOE2053	Energy Technology
4	Electronics & Telecommunication Engineering	2501PETCOE2051	Advanced Operating Systems
5		2501PETCOE2052	Cyber Security
6		2501PETCOE2053	Artificial Intelligence and Machine Learning
7	Construction Management (Civil Engineering)	2501PCCMOE2051	Water Power Engineering
8		2501PCCMOE2052	Waste to Energy
9		2501PCCMOE2053	Contracts & Tenders
10	Mechanical Design (Mechanical Engineering)	2501PMDEOE2051	Cryogenics
11		2501PMDEOE2052	Design for Manufacture & Assembly
12		2501PMDEOE2053	Enterprise Resource Planning
13	Structural Engineering (Civil Engineering)	2501PCSTOE2051	Cost Management of Engineering Projects
14		2501PCSTOE2052	Optimization Techniques in Civil Engineering
15		2501PCSTOE2053	Industrial Safety
16	Computer Science and Engineering	2501PCSEOE2051	Ethical AI & Explainability
17		2501PCSEOE2052	Computer Vision
18		2501PCSEOE2053	High Performance Computing for Multidisciplinary Research

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPCC101 Course Name: Mathematical Foundation of Computer Science				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

Discrete Mathematics, Automata Theory

Course Objectives:

1	To introduce fundamental concepts of discrete structures, formal logic, and proof techniques essential for computer science.
2	To develop an understanding of formal languages, grammars, and automata theory for modelling computational processes.
3	To explore Turing machines, computability, and decidability for analyzing problem solvability.
4	To study computational complexity classes and reduction techniques for classifying problem difficulty and efficiency.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Demonstrate understanding of discrete structures, relations, functions, and formal proof techniques used in computer science
CO2	Study different language classes, grammars, and finite state machines to model computational problems.
CO3	Design and evaluate context-free grammars and pushdown automata for language recognition and parsing applications.
CO4	Apply concepts of Turing machines, computability, and decidability to characterize solvable and unsolvable problems.
CO5	Assess computational complexity classes (P, NP, NP-complete, PSPACE) and apply reduction techniques to classify problem difficulty.

Course Description:

This course introduces the mathematical and theoretical foundations of computer science, covering discrete structures, automata, grammars, and formal proof techniques. It further explores Turing machines, computability, and computational complexity to develop a deep understanding of problem-solving and computational limits.

Course Contents

Unit-1	**Introduction to Discrete Structures**	**06 Hours**
Sets, Relations and Functions; Algebraic Structures, Morphisms, Lattices and Boolean Algebras. Theorems and types of proofs, formal proofs, deductive, reduction to definition, proof by construction, contradiction, induction, indirect, automatic, counter-examples.		

<table>
<tr><td colspan="3"></td></tr>
<tr><td>Unit-2</td><td>State Machines and Grammars</td><td>06 Hours</td></tr>
<tr><td colspan="3">Types of Languages, Types of grammar, recurrence relations, Regular expressions, Finite State Machines, DFA, NFA, Equivalence of DFA & NFA, Kleene’s Theorem, pumping Lemma, Applications.</td></tr>
<tr><td>Unit-3</td><td>Push down automata and CFG</td><td>06 Hours</td></tr>
<tr><td colspan="3">PDA, N-PDA, CFG, ambiguous grammar, non-ambiguous grammar, CNF, Parsers: Top-down, Bottom-up, applications.</td></tr>
<tr><td>Unit-4</td><td>Turing Machines</td><td>06 Hours</td></tr>
<tr><td colspan="3">Turing machines, variations of TMs, Combining TM’s, programming techniques for TMs, Universal Turing Machines, recursive and recursively enumerable languages.</td></tr>
<tr><td>Unit-5</td><td>Computability</td><td>06 Hours</td></tr>
<tr><td colspan="3">Church-Turing Thesis, Decision Problems, Decidability and Undecidability, Halting Problem of Turing Machines.</td></tr>
<tr><td>Unit-6</td><td>Computational Complexity</td><td>06 Hours</td></tr>
<tr><td colspan="3">Time Complexity, The class P, The class NP, NP-Completeness, Reduction, co-NP, Polynomial Hierarchy. Space Complexity -- Savich's Theorem, The class PSPACE.</td></tr>
<tr><td colspan="3">Learning Resources:</td></tr>
<tr><td colspan="3">Text Books:</td></tr>
<tr><td>1</td><td colspan="2">J.P. Trembley and R. Manohar -- Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill Book Co.</td></tr>
<tr><td>2</td><td colspan="2">“Introduction to Automata Theory, Language and Computations”, J.E. Hoperoft, Rajeev Motwani & J. D. Ullman, Pearson Education Asia, 2nd Edition.</td></tr>
<tr><td>3</td><td colspan="2">Introduction to Languages and Theory of Computation”, John. Martin MGH.3rd Edition</td></tr>
<tr><td colspan="3">Reference Books:</td></tr>
<tr><td>1</td><td colspan="2">Michael Sipser -- Introduction to The Theory of Computation, Thomson Course Technology.</td></tr>
<tr><td>2</td><td colspan="2">H.R. Lewis and C. H. Papadimitrou -- Elements of the Theory of Computation, Prentice Hall International.</td></tr>
<tr><td>3</td><td colspan="2">“Theory of Computer Science”, E. V. Krishamoorthy.</td></tr>
<tr><td colspan="3">MOOC / NPTEL/YouTube Links</td></tr>
<tr><td>1</td><td colspan="2"></td></tr>
</table>

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPCC102 Course Name: Feature Engineering and Machine Learning				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Linear Algebra and Probability Theory
2) Data Mining / Machine Learning Basics

Course Objective:

1	Introduce the foundations of feature engineering, machine learning paradigms, and their applications in solving real-world problems.
2	Develop an understanding of techniques for feature representation, transformation, selection, and construction to enhance model performance.
3	Provide knowledge of supervised, unsupervised, and probabilistic learning algorithms, along with their theoretical underpinnings and practical implementations.
4	Equip students with the ability to design, evaluate, and optimize machine learning pipelines using modern tools and frameworks.

Course Outcomes:

Cos	At the end of successful completion of the course, the students will be able to
CO1	Explain the importance of feature engineering and machine learning in data-driven decision making.
CO2	Apply techniques for feature cleaning, transformation, selection, and representation in diverse datasets.
CO3	Implement regression, classification, clustering, and probabilistic models to address practical machine learning problems.
CO4	Analyze the impact of features, algorithms, and hyperparameters on model performance using suitable evaluation measures.
CO5	Design and optimize end-to-end ML solutions by integrating feature engineering methods with appropriate algorithms and tools.

Course Description:

This course introduces advanced concepts and practical techniques in feature engineering and machine learning, emphasizing the critical role of data representation in building effective predictive models. Students will learn methods for transforming, selecting, and constructing features, along with core machine learning algorithms for regression, classification, clustering, and probabilistic modeling. The course blends theory with practical case studies, enabling students to design, implement, and evaluate end-to-end ML pipelines using modern tools and frameworks.

Course Content

Unit-1	**Introduction to Feature Engineering & Representation**	**6 Hours**

Importance of Feature Engineering in the ML pipeline, Feature Understanding, Feature Improvement, Feature Selection, Feature Construction, Feature Transformation: (1. Scalars, Vectors, and Feature Spaces 2. Dealing with Counts, Binarization, Quantization/Binning 3.Log Transformation 4. Scaling & Normalization techniques: Min-Max Scaling, Standardization (Variance Scaling), ℓ2 Normalization 5. Interaction Features), Feature Learning.		
Unit-2	**Text, Categorical Features & Feature Selection**	**6 Hours**
Feature Engineering for Text Data: Bag-of-X models: Bag-of-Words, n-Grams, and Phrases, TF-IDF and its role in weighting terms, Filtering and cleaning for meaningful features. **Categorical Feature Engineering:** Encoding techniques: One-Hot, Ordinal, Target/Mean encoding, Handling high-cardinality categorical variables. **Feature Selection in Machine Learning:** Importance & goals of feature selection, Impact of irrelevant features, overfitting risks, and validation considerations, Overview of methodologies [Filters, Wrappers (Recursive Feature Elimination, Stepwise), and Embedded methods]		
Unit-3	**Feature Transformations & Feature Learning**	**8 Hours**
Feature Transformation & Dimensionality Reduction: Intuition and derivation of linear projections, Principal Component Analysis (PCA): (Variance and empirical variance, Matrix-vector formulation & general solution, Transforming features, whitening & ZCA, Considerations, limitations, and use cases of PCA) **Feature Learning Approaches:** Parametric vs. non-parametric assumptions in data, Feature learning algorithms, Reconstructing data with unsupervised methods, Restricted Boltzmann Machines (RBMs), Extracting PCA and RBM components (MNIST case study overview), Word embeddings (Word2Vec, etc.) and their applications in information retrieval.		
Unit-4	**Introduction to Machine Learning & Regression**	**6 Hours**
Foundations of Machine Learning: Definition, terminology, and categories of ML problems, Types of learning: supervised, unsupervised, reinforcement, Machine learning architecture & lifecycle, Performance measures (accuracy, RMSE, precision/recall, etc.) **Regression Models:** Simple Linear Regression (hypothesis, cost function, gradient descent, learning rate), Matrix form representation of regression, Multivariate Linear Regression (hypothesis, gradient descent for multiple features, feature scaling), Polynomial regression.		
Unit-5	**Classification Algorithms & Neural Networks**	**6 Hours**
Logistic Regression: Hypothesis representation, decision boundary, Cost function & gradient descent for logistic regression, Multiclass classification (one-vs-rest, softmax), Regularization: overfitting, underfitting, regularized linear & logistic regression **Neural Networks:** Neuron representation & hypothesis, Cost function and gradient descent for a single neuron, Neural network architecture, multiclass classification with neural networks, Backpropagation algorithm.**Other Classifiers:** Decision Trees, Random Forests, Naïve Bayes Classifier, Instance-based classifier: K-Nearest Neighbour (KNN) basics and examples.		
Unit-6	**Probabilistic Models, Unsupervised Learning & Association Rules**	**6 Hours**

Probabilistic Models: Uncertainty & role of probability in ML, Normal distribution & geometric interpretations, Discriminative learning with maximum likelihood, Models with hidden variables: Hidden Markov Models (HMMs), Gaussian Mixtures, Expectation-Maximization (EM) methods **Clustering Methods:** Introduction to clustering, K-Means clustering: algorithm, applications, limitations, Hierarchical clustering: types, dendrograms, use cases **Association Rule Mining:** Overview of association rules and applications, Apriori algorithm for rule generation, Evaluation of candidate rules (support, confidence, lift), Case study: transactions in a grocery store, Validation, testing, and diagnostics of rules.

Learning Resources:	
Text Books:	
1	Sinan Ozdemir, Divya Susarla, "Feature Engineering Made Easy", Packt Publishing, ISBN 978-1- 78728-760-0 2.
2	Alice Zheng & Amanda Casari, "Feature Engineering for Machine Learning: Principles and Techniques for data scientist", Oreilly
3	Machine Learning with Python- an approach to applied ML, by Abhishek Vijayvargia, BPB publications
4	Practical Machine Learning by Sunila Gollapudi Packt Publishing Ltd
5	Machine Learning by Tom M. Mitchell, McGraw Hill Education; First edition
Reference Books:	
1	Max Kuhn, Kjell Johnson, "Feature Engineering and Selection: A Practical Approach for Predictive Models" 1st Edition, Chapman & Hall/CRC Data Science Series, ISBN 13-978-1-138-07922-9
2	Machine Learning for dummies John Paul Muller, Willey Publication
3	Ethem Alpaydin: Introduction to Machine Learning, PHI 2nd Edition-2013
4	Kevin Murphy, Machine Learning: a Probabilistic Approach, MIT Press, 1st Edition, 2012, ISBN No.: 978-0262-30616-4
5	C.M. Bishop, Pattern Recognition and Machine learning, Springer, 1st Edition, 2013, ISBN No.: 978- 81-322-0906-5

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1031 Course Name: Network Engineering and Management				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Fundamentals of Computer Networks
2) Basics of Operating Systems
3) Introductory knowledge of Security Concepts

Course Objectives:

1	To provide students with in-depth knowledge of computer networks and advanced network engineering concepts
2	To introduce methodologies and tools for efficient network planning, design, and management.
3	To familiarize students with emerging trends in network security, cloud networking, and software-defined networking (SDN).
4	To develop problem-solving and research skills for addressing real-world challenges in networking and its management

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain fundamental and advanced principles of computer networks and management systems.
CO2	Analyze network traffic, performance, and fault-tolerance strategies.
CO3	Design efficient, scalable, and secure enterprise networks
CO4	Apply tools, protocols, and frameworks for network monitoring and management.
CO5	Evaluate and adapt modern network technologies such as SDN, IoT networks, and 5G.

Course Description:

This course covers advanced aspects of network engineering and management including network architectures, traffic engineering, quality of service, performance management, fault analysis, and emerging paradigms such as SDN, IoT, and cloud networks. Emphasis will be placed on both theoretical foundations and practical tools used in real-world enterprise environments.

Course Content

Unit-1	**Advanced Networking Concepts**	**06 Hours**
Review of OSI & TCP/IP Models, High-speed networks: Gigabit Ethernet, MPLS, DWDM, Network topologies and architectures, IPv6 and transition mechanisms.		

Unit-2	Traffic Engineering & QoS	06 Hours
Traffic analysis and modelling, Quality of Service (QoS) parameters, Resource allocation and congestion control, Differentiated and integrated services.		
Unit-3	**Network Management Fundamentals**	**06 Hours**
FCAPS Model (Fault, Configuration, Accounting, Performance, Security), SNMP architecture and management information bases (MIBs), RMON, NetFlow, and sFlow tools, Case studies in enterprise network management.		
Unit-4	**Performance and Security Management**	**06 Hours**
Network monitoring tools (Wireshark, SolarWinds, Nagios), Performance metrics and optimization techniques, Fault-tolerant systems and redundancy, Security management: Firewalls, IDS/IPS, SIEM.		
Unit-5	**Emerging Networking Paradigms**	**06 Hours**
Software Defined Networking (SDN) and OpenFlow, Network Function Virtualization (NFV), Cloud Networking and Edge Computing, IoT and 5G network management.		
Unit-6	**Research & Case Studies**	**06 Hours**
Network simulation and modeling tools (NS2/NS3, Mininet), Real-time case studies of telecom and enterprise networks, Industry practices in large-scale network management, Future research trends and challenges.		

Learning Resources:	
Text Book	
1	Data Communications and Networking, Behrouz A. Forouzan, McGraw Hill, 5th Edition
2	Network Management: Principles and Practice, William Stallings, Pearson Education, 2nd Edition.
3	Network Management: Principles and Practice, Mani Subramanian, Pearson Education India, 2nd Edition.
Reference Book	
1	Computer Networks, Pearson Education, 5th Edition, Andrew S. Tanenbaum, David Wetherall, Pearson Education, 5th Edition.
2	Computer Networks: A Systems Approach, Larry L. Peterson & Bruce S. Davie, Morgan Kaufmann, 5th Edition.
3	Computer Networking: A Top-Down Approach, James F. Kurose & Keith W. Ross Pearson, 8th Edition,

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1032 Course Name: Linear Algebra				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Discrete Mathematics, Automata Theory

Course Objectives:

1	To introduce fundamental concepts of linear systems, vector spaces, and transformations.
2	To develop skills in matrix methods, eigenvalue problems, and quadratic forms.
3	To provide understanding of orthogonality, inner product spaces, and their applications.
4	To enhance problem-solving ability in complex variables and contour integration.
5	To apply linear algebra concepts in computer science and engineering domains such as optimization, image processing, and computer graphics.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Solve and interpret systems of linear equations using matrix methods.
CO2	Demonstrate understanding of vector spaces, basis, and dimension.
CO3	Apply linear transformations and inner product concepts, including Gram-Schmidt orthogonalization and least squares.
CO4	Compute eigenvalues, eigenvectors, diagonalization, and analyze quadratic forms for practical problems.
CO5	Evaluate and apply complex functions, Cauchy-Riemann equations, harmonic functions, and contour integrals in engineering applications.

Course Description:

This course provides mathematical foundations in linear algebra and complex variables, focusing on systems of equations, vector spaces, linear transformations, eigenvalues and eigenvectors, inner product spaces, quadratic forms, and functions of complex variables. The emphasis is on both theory and applications in optimization, computer graphics, and scientific computing.

Course Content

Unit-1	**System of Linear Equations**	**06 Hours**

Rank of a matrix, elementary matrices, system of linear equations, Gauss Jordan elimination, applications of systems of equations in science and engineering.

Unit-2	Vector Spaces	06 Hours
Vector space, subspaces, linear combination and spanning set, linear dependence and independence, basis and dimension, row space, column space, and null space of a matrix.		
Unit-3	**Linear Transformations**	**06 Hours**
Linear transformations, matrix representation of a linear transformation, rank and nullity, orthogonal transformations, geometric applications of linear transformations.		
Unit-4	**Inner Product Spaces**	**06 Hours**
Inner product, orthogonality, orthogonal complements, Gram-Schmidt orthogonalization, applications to least-squares fitting of data.		
Unit-5	**Eigenvalues and Eigenvectors**	**06 Hours**
Eigenvalues and eigenvectors, algebraic and geometric multiplicity, Cayley-Hamilton theorem, diagonalization of matrices, quadratic forms and definiteness, Sylvester's criterion, applications in optimization and computer science.		
Unit-6	**Complex Variables**	**06 Hours**
Functions of complex variables, analytic functions, Cauchy-Riemann equations, harmonic functions, Milne-Thompson's method, contour integrals, Cauchy's integral formula, applications in image processing and computer graphics.		

Learning Resources:	
Text Book	
1	Howard Anton & Chris Rorres, Elementary Linear Algebra, John Wiley & Sons.
2	David C. Lay, Linear Algebra and its Applications, Pearson.
3	Gilbert Strang, Linear Algebra and its Applications (4th Ed.), Cengage Learning.
Reference Book	
1	Seymour Lipschutz & Marc Lipson, Schaum's Outline of Linear Algebra (6th Ed.), McGraw-Hill.
2	David Poole, Linear Algebra: A Modern Introduction (4th Ed.), Cengage Learning.
3	Ron Larson & David C. Falvo, Linear Algebra: An Introduction, Cengage Learning.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering					
Course Code: 2501PCSEPE1033 Course Name: Optimization Techniques					
Teaching Scheme		**Credit**	**Evaluation Scheme**		
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**	
			ESE:	**60 Marks**	

Prerequisites:

1) Mathematics
2) Algorithms and Data Structures
3) programming language (such as MATLAB, Python, or C/C+)

Course Objectives:

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**

Course Description:

This course introduces the fundamental principles and methods of optimization, covering linear, nonlinear, dynamic, and integer programming techniques. It emphasizes modern approaches such as evolutionary algorithms and metaheuristics for solving complex engineering problems. The course equips students with analytical and computational skills to model, formulate, and optimize real-world systems efficiently.

Course Content

Unit-1	**Introduction to Optimization**	**08 Hours**
Definition, importance, and applications of optimization in engineering and science Classification of optimization problems: linear vs nonlinear, static vs dynamic, constrained vs unconstrained, deterministic vs stochastic		

General mathematical formulation of optimization problems Concepts of optimality, feasible solutions, global vs local optima		
Unit-2	**Linear Programming**	**06 Hours**
Formulation of linear programming problems Graphical method, Simplex method, Big-M and Two-Phase methods Duality theory and sensitivity analysis Transportation and assignment problems		
Unit-3	**Non-Linear Programming**	**06 Hours**
Unconstrained optimization: gradient methods, Newton's method, quasi-Newton methods, Constrained optimization: Lagrange multipliers, Karush–Kuhn–Tucker (KKT) conditions, Quadratic programming, Penalty and barrier methods		
Unit-4	**Dynamic and Integer Programming**	**06 Hours**
Principle of optimality, recursive equations, applications in shortest path and resource allocation problems, Integer programming: branch and bound, cutting plane methods Mixed-integer programming		
Unit-5	**Metaheuristic and Evolutionary Optimization**	**08 Hours**
Genetic Algorithms (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Ant Colony Optimization (ACO), Hybrid optimization techniques and their engineering applications,Comparative advantages over classical methods		
Unit-6	**Applications and Case Studies**	**06 Hours**
Network optimization, Resource allocation and scheduling problems, Multi-objective optimization basics (Pareto optimality, trade-offs)		
Learning Resources:		

Text Book	
1	Engineering Optimization: Theory and Practice – S. S. Rao Wiley, 2019 (4th Edition.
2	Operations Research: An Introduction – H. A. Taha Pearson, 2017 (10th Edition)
3	Optimization for Engineering Design: Algorithms and Examples – Kalyanmoy Deb, PHI, 2012
Reference Book	
1	Operations Research: Theory and Applications – Macmillan, J. K. Sharma – 2018
2	F. S. Hillier & G. J. Lieberman – Introduction to Operations Research – McGraw Hill, 2021 (11th Edition)
3	Genetic Algorithms in Search, Optimization and Machine Learning – David E. Goldberg Addison- Wesley, 1989

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1041 Course Name: Semantic Web				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Discrete Mathematical Structure
2) Database Systems
3) Basic knowledge of Web Technologies

Course Objectives:

1	Understand the foundational concepts and layered architecture of the Semantic Web
2	Develop the ability to model and describe web resources
3	Gain proficiency in querying and manipulating semantic data
4	Explore the principles of ontology engineering and utilize Semantic Web tools

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Describe the goals, architecture, and technologies of the Semantic Web.
CO2	Apply RDF and RDFS to model and describe web resources with semantic meaning.
CO3	Analyze SPARQL queries to retrieve, filter, and manipulate data from RDF graphs.
CO4	Evaluate the correctness and consistency of ontologies using OWL2 reasoning tools.
CO5	Create semantic web applications by designing ontologies and integrating tools for reasoning and data querying.

Course Description:

This course introduces the principles and technologies of the Semantic Web, focusing on standards like RDF, RDFS, OWL, and SPARQL. Students will learn to model data semantically, construct and query knowledge graphs, design ontologies, and use Semantic Web tools. The course also covers ontology engineering practices and the development of semantic web applications.

Course Content

Unit-1	**Foundation of Semantic Web Technologies**	**4 Hours**
Introduction, Semantic Web Technologies, A layered approach		
Unit-2	**Describing Web Resources: RDF**	**7 Hours**
Introduction, RDF: Data Model, RDF Syntaxes, RDFS: Adding Semantics, RDF Schema: The Language, RDF and RDF Schema in RDF, An Axiomatic Semantics for RDF and RDF Schema.		

Unit-3	**Querying the Semantic Web**	**7 Hours**
SPARQL Infrastructure, Basics: Matching Patterns, Filters, Constructs for Dealing with an Open World, Organizing Result Sets, Other Forms of SPARQL Queries, Querying Schemas, Adding Information with SPARQL Update, The Follow Your Nose Principle.		
Unit-4	**Web Ontology Language: OWL2**	**7 Hours**
Introduction, Requirements for Ontology Languages, Compatibility of OWL2 with RDF/RDFS, The OWL Language, OWL2 Profiles.		
Unit-5	**Ontology Engineering**	**7 Hours**
Introduction, Constructing Ontologies Manually, Reusing Existing Ontologies, Semiautomatic Ontology Acquisition, Ontology Mapping, Exposing Relational Databases, Semantic Web Application Architecture.		
Unit-6	**Semantic Web Software Tools**	**4 Hours**
Introduction, Metadata and Ontology Editors, Reasoners, Other Tools.		

Learning Resources:

Text Book	
1	A Semantic Web Primer, Third Edition by Grigoris Antoniou, Paul Groth, Frank Van Harmelen, Rinke Hoekstra, MIT Press
2	Semantic Web: Concepts, technologies and applications by Karin Breitman, Marco Antonio Casanova, Walt Truszkowski, Springer
Reference Book	
1	Semantic Web: Concepts, Technologies and Applications by Karin K. Breitman, Springer India
2	Semantic Web: Semantics for Data and Services on the Web by Kashyap, Bussler, Moran, Springer India
3	Semantic Web Explained by Peter Szeredi, Gergely Lukacsy, Tamas Benko, Cambridge University Press

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering					
Course Code: 2501PCSEPE1042 Course Name: Natural Language Processing					
Teaching Scheme		**Credit**	**Evaluation Scheme**		
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**	
			ESE:	**60 Marks**	

Prerequisites:

1) Data Structures & Algorithms, Artificial Intelligence / Machine Learning , Probability & statistics

Course Objectives:

1	**Introduce NLP Foundations:** Provide students with fundamental knowledge of natural language processing concepts, linguistic structures, and computational models.
2	**Develop Algorithmic Understanding**:Familiarize students with algorithms, statistical models, and deep learning techniques used in processing and understanding human language
3	**Hands-on Practical Skills**: Enable students to design, implement, and evaluate NLP systems for tasks such as tokenization, POS tagging, parsing, text classification, sentiment analysis, and machine translation.
4	**Encourage Research & Innovation**: Cultivate research aptitude by encouraging students to work on real-world problems in NLP such as chatbots, information retrieval, question answering, and speech-to-text systems.

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Demonstrate the understanding of basic text processing techniques in NLP 03 Build language models and demonstrate Word Sense Disambiguation using WordNet.
CO2	Design, implement and evaluate part-of-speech taggers and parsers for a language
CO3	Build language models and demonstrate Word Sense Disambiguation using WordNet
CO4	Analyse and build word embedding for different languages.

Course Description:

This course introduces the fundamental concepts, models, and algorithms of Natural Language Processing (NLP), focusing on the interaction between human language and computers. It covers linguistic foundations, text preprocessing, statistical and machine learning methods, and deep learning techniques for language understanding. Through hands-on projects and research-oriented assignments, students will gain practical experience in building NLP systems while considering ethical and societal issues such as bias, fairness, and multilingualism.

Course Content

Unit-1	Introduction	06 Hours

What is NLP, Fundamental and Scientific goals, Engineering goals, stages of NLP, problems in NLP, Applications of NLP, Empirical Laws of language, zipf's law, Heap's law.		
Unit-2	**Basic Text Processing**	**06 Hours**
Tokenization, word token, word type, sentence segmentation, feature extraction, issues in tokenization for different languages, word segmentation, text segmentation, normalization, case folding, Spelling Correction, Morphology, Stemming, Porters Algorithm, , lemmatization, spelling correction - dynamic programming approach for finding edit distance, N-gram Language Modeling-context sensitive spelling correction, probabilistic language model, auto completion prediction, Evaluation and perplexity, Smoothing techniques.		
Unit-3	**POS Tagging**	**06 Hours**
Sequence labeling tasks of NLP, POS tagging, POS tag sets, Hidden Markov Model Introduction, Markov Processes, HMM characterization -Likelihood of a sequence (Forward Procedure, Backward Procedure), Best state sequence (Viterbi Algorithm), Re-estimation(BaumWelch - Forward-Backward Algorithm) , Models for Sequential tagging – Maximum Entropy, Conditional Random Field.		
Unit-4	**Syntax**	**06 Hours**
Constituency and dependency parsing, Constituency parser -Syntactic structure, Parsing methodology, Different parsing algorithms, Parsing in case of ambiguity, Probabilistic parsing, CKY algorithm, Issues in parsing, Dependency parsing- Syntactic structure, Parsing methodology, Transition-Based Dependency Parsing, Graph-Based dependency parsing, Evaluation, Co-reference resolution, Named-entity recognition.		
Unit-5	**Knowledge Base and Semantics**	**06 Hours**
WordNet: Word Senses, Word relations, Word similarity and thesaurus methods, Word sense disambiguation, WordNet. Lexical and Distributional Semantics - Introduction, models of semantics, applications.		
Unit-6	**Word Embedding**	**06 Hours**
Introduction, one-hot vectors, methods of generating word embeddings, Skip-gram, CBOW, Glove model, Fast Text model, evaluation measures-rough scores.		
Learning Resources:		
Text Book		

1	Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Second Edition, Prentice Hall, 2008, ISBN: 978-0131873216.
2	Allen James, "Natural Language Understanding", Second Edition, Benjamin/Cumming, 1994, ISBN: 978-0805303346.
3	Chris Manning and Hinrich Schuetze, "Foundations of Statistical Natural Language Processing", MIT Press, ISBN: 978-0262133609
Reference Book	

1	Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
2	Conferences: Annual Meeting of the Association of Computational Linguistics (ACL),
3	Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT))

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1043 Course Name: Distributed System & Cloud Computing				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Computer Networks
2) Operating System-I
3) Information Security Course

Course Objectives:

1	To Understand the foundational concepts, models, and architectures of distributed systems.
2	To Learn communication, synchronization, consistency, and fault tolerance mechanisms in distributed environments.
3	To Understand cloud computing models, virtualization techniques, and cloud platform architecture.
4	To Explore programming environments and tools used in cloud-based distributed systems.

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Explain the goals, challenges, and architecture of distributed systems including middleware, RPC, and RMI.
CO2	Apply interprocess communication, mutual exclusion, and election algorithms in distributed settings.
CO3	Analyze consistency models, replication, and fault tolerance techniques including distributed file systems
CO4	Describe cloud architecture, virtualization layers, and resource management across platforms like AWS, Azure, and GCP.
CO5	Develop and deploy applications using cloud programming tools such as Google App Engine and AWS SDKs.

Course Description:

This course covers the fundamental principles and architectures of distributed systems, including communication, synchronization, consistency, and fault tolerance. It also explores cloud computing concepts such as service models, virtualization, and cloud infrastructure platforms like AWS and Azure. Students will gain practical exposure to distributed algorithms, middleware, and cloud-based application development.

Course Content

Unit-1	Foundations of Distributed Systems	7 Hours

Introduction to Distributed Systems: Definition, Goals , Challenges Architectural Models: Client-

Server, Peer-to-Peer (P2P), Service-Oriented Architecture (SOA), Microservices System Models: Physical Models, Architectural Models, Fundamental Models (Interaction, Failure, Security) Middleware: Role of middleware, Introduction to Remote Procedure Call (RPC), Remote Method Invocation (RMI), Message Queues.		
Unit-2	**Communication and Synchronization in Distributed Systems**	**7 Hours**
Interprocess Communication (IPC): External Data Representation and Marshalling, Client-Server Communication (Sockets, Connection-oriented vs. Connection-less) Remote Invocation: RPC, RMI, Message-Oriented Middleware (MOM), Group Communication: Unicast, Multicast, Broadcast, Ordered Multicast, Atomic Multicast Distributed Mutual Exclusion: Centralized, Distributed (Ricart-Agrawala), Token-based (Suzuki-Kasami) Election Algorithms: Bully Algorithm, Ring Algorithm.		
Unit-3	**Consistency, Fault Tolerance, and Distributed Resources**	**7 Hours**
Consistency Models: Data-Centric Consistency, Client-Centric Consistency Replication: Reasons for replication, Replica Management, Consistency Protocols. Fault Tolerance: Failure Models, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit (Two-Phase Commit, Three-Phase Commit), Recovery Distributed File Systems (DFS): Architecture, Design Issues, Case Studies (NFS, AFS, HDFS) Naming Services: DNS, X.500 Directory Service, LDAP.		
Unit-4	**Cloud Platform Architecture over Virtualized Data Centers**	**6 Hours**
Cloud Computing and Service Models, Data-Center Design and Interconnection Networks, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms: GAE, AWS, and Azure, Inter-cloud Resource Management, Cloud Security and Trust Management.		
Unit-5	**Virtual Machines and Virtualization of Clusters**	**5 Hours**
Implementation Levels of Virtualization, Virtualization Structures/Tools and Mechanises, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data-Center Automation.		
Unit-6	**Cloud Programming and Software Environments**	**5 Hours**
Features of Cloud and Grid Platforms, Parallel and Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments.		
Learning Resources:		
Text Book		
1	Distributed Systems: Concepts and Design by George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. (Pearson Education) (Unit 1,2)	
2	Distributed Systems: Principles and Paradigms by Andrew S. Tanenbaum and Maarten van Steen. (Pearson Education) (Unit 3)	
3	Distributed and Cloud Computing: From Parallel Processing to the Internet of Things by Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra (Unit 4, 5 and 6)	
Reference Book		

1	Distributed Algorithms by Nancy A. Lynch. (Morgan Kaufmann)
2	Mastering Cloud Computing: Foundations and Applications Programming by Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi. (McGraw Hill Education / Morgan Kaufmann)

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1051 Course Name: Blockchain Technology				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Cryptography Basics, Hashing (SHA-256, MD5), Public-key and private-key encryption, Digital signaturesOperating System-I
2) Data Structures, Linked lists, trees (Merkle trees), Hash maps, Graphs
3) Programming Languages : Python, JavaScript,
4) Database Concepts, Distributed databases

Course Objectives:

1	To introduce the fundamental concepts of blockchain technology
2	To explain the role of cryptography and distributed peer-to-peer networks
3	To explore various blockchain types and their applications
4	To develop an understanding of cryptocurrency and Bitcoin mechanisms
5	To study consensus algorithms and blockchain ecosystem components

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain the fundamentals and working principles of Blockchain technology and its real-world applications
CO2	Differentiate between types of blockchains, cryptocurrencies, and distributed ledger systems
CO3	Apply cryptographic techniques and hashing concepts to demonstrate secure data transactions in blockchain networks.
CO4	Evaluate the significance of consensus algorithms and their role in ensuring blockchain integrity.
CO5	Design a basic blockchain framework integrating blocks, wallets, and keys for a decentralized application

Course Description:

This course covers the fundamental principles and architectures of distributed systems, including communication, synchronization, consistency, and fault tolerance. It also explores cloud computing concepts such as service models, virtualization, and cloud infrastructure platforms like AWS and Azure. Students will gain practical exposure to distributed algorithms, middleware, and cloud-based application development.

Course Content		
Unit-1	**Introduction to Blockchain Technology**	**05 Hours**
Why blockchain matters more than you think,What is Blockchain? How does a Blockchain work,The origins of blockchain, Blockchain Applications		
Unit-2	**Basic Cryptography**	**06 Hours**
Blockchain came from Bitcoin, Why is Blockchain a Distributed P2P Network? Blockchain Vs Cryptocurrency, Types of Blockchain.		
Unit-3	**Cryptocurrency and Bitcoin**	**05 Hours**
What Are Different Blockchain Technologies?,Benefits of using Blockchain Technology,The Origin of Blockchain Completed, Blockchain came from Bitcoin.		
Unit-4	**Basic Understanding of Blockchain**	**08 Hours**
Why blockchain matters more than you think What is Blockchain and what is it going to change The Origin of Blockchain, A deeper dive into understanding Blockchain. Overview of Blockchain.		
Unit-5	**Basics of Blockchain**	**06 Hours**
Blockchain Technology, The Evolution of Blockchain Technology, Blockchain Technology – Basics, Introduction to the Decentralized Web Introduction to Distributed Ledgers, Merkle Tree and Hashing, Blocks, Wallets, and Addresses, Public and Private Key		
Unit-6	**Consensus Algorithms**	**06 Hours**
Cryptography and Cryptographic Algorithms, Transaction Execution and Distribution, Components of Blockchain Ecosystem, Blockchain Architecture.		
Learning Resources:		
Text Book		
1	The Basics of Bitcoins and Blockchains, Anthony Lewis, Two Rivers Distribution	
2	Blockchain Explained: A Pragmatic Approach, Srihari Kapu, Notion Press (8 December 2020)	
3	Blockchain Technology, Chandramouli Subramanian , University Press India	
Reference Book		
1	Arvind Narayanan, “Bitcoin and Cryptocurrency Technologies- A Comprehensive Introduction”, Princeton University Press, 2016.	
2	William Magnuson, “Blockchain Democracy- Technology, Law and the Rule of the Crowd”, Cambridge University Press, 2020.	

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1052 Course Name: Smart Technology and Internet of Things				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Fundamentals of Data CommunicationData Structures, Linked lists, trees (Merkle trees), Hash maps, Graphs
2) Basics of Computer Networks
3) Fundamentals of Python Programming

Course Objectives:

1	To provide foundational knowledge of IoT concepts and architecture
2	To equip students with IoT design and development skills
3	To enhance understanding of IoT communication protocols and technologies
4	To develop hands-on expertise with IoT hardware and software platforms
5	To enable application of IoT in real-world scenarios

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Identify and understand the unique characteristics and components of loT
CO2	Compare various development boards Arduino, Raspberry pi, Beagle bone
CO3	Illustrate a middleware for loT
CO4	Analyze various protocols for IoT
CO5	Compare various IoT communication technologies and Design various IoT applications

Course Description:

The Internet of Things (IoT) is a network of physical objects equipped with electronics, sensors, software, and connectivity to collect and exchange data. It enables remote sensing and control through existing networks, enhancing efficiency, accuracy, and economic benefits. When combined with sensors and actuators, IoT forms cyber-physical systems like smart grids, homes, transportation, and cities. Each object has a unique identifier and can communicate with others, with experts predicting around 50 billion connected devices by 2030.

Course Content

Unit-1	Fundamentals of Internet of Things (IoT)	06 Hours
Definition and characteristics of IoT, Technical Building blocks of IoT, Device, Communication		

Technologies, Data, Physical design of IoT, IoT enabling technologies, IoT Issues and Challenges: Planning, Costs and Quality, Security and Privacy, Risks.		
Unit-2	**IoT Design Methodology**	**06 Hours**
IoT systems management, IoT Design Methodology: Specifications Integration and Application Development.		
Unit-3	**Communication in IoT**	**06 Hours**
IoT Protocols: MQTT, CoAP, XMPP and AMQT, IoT communication models, IoT Communication technologies: Bluetooth, LTE-A, DTLS, BLE, Zigbee, Zwave, NFC, RFID, LiFi, Wi-Fi, Interfacing of wifi, RFID, Zigbee, NFC with development board.		
Unit-4	**Building IoT with Raspberry Pi and Intel Galileo/Arduino**	**06 Hours**
Physical device, Raspberry Pi Interfaces: Programming – APIs / Packages, Web services. Intel Galileo Gen2 with Arduino: Interfaces, Arduino IDE, Programming APIs and Hacks, IoT standards, Cloud computing for IoT, Bluetooth Low Energy, beacons		
Unit-5	**The Internet of Things to The Web of Things**	**06 Hours**
Resource-oriented Architecture and Best Practices: Designing RESTful Smart Things – Web enabling, Constrained Devices – The Future Web of Things		
Unit-6	**IoT Applications and Case Studies**	**06 Hours**
Various Real time applications of IoT, Case studies: Smart Agricultural: characteristics and applications - Scarecrow, Smart Irrigation System, Crop Water Management, Integrated Pest Management, Sensor-based field and resource mapping, Remote equipment monitoring, e-health: Characteristics of e-health and applications – monitoring of health parameters, smart medicine box, elderly people monitoring, challenges. Smart Metering, Smart Home Automation, Smart Cards, IoT in Sports, IoT in Smart Cities/Transportation, Smart parking.		
Learning Resources:		
Text Book / Reference Books		

1	Architecting the Internet of Things, Bernd Scholz-Reiter, Florian Michahelles, Springer
2	Getting Started with the Internet of Things, Cuno Pfister, O'Reilly Media
3	Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. Ovidiu Vermesan, Dr. Peter Friess River Publishers.
4	“The Internet of Things Connecting Objects to the Web” Hakima Chaouchi, Wiley Publications.
5	Intel Galileo and Intel Galileo Gen 2: API Features and Arduino Projects for Linux Programmers”, Manoel Carlos Ramon Apress

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE1053 Course Name: High Dimensional Data Analysis				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Linear Algebra and Probability Theory
2) Data Mining / Machine Learning Basics
3) Programming and Algorithms

Course Objectives:

1	Acquire a broad perspective on high-dimensional data, its challenges, and approaches to dimensionality reduction.
2	Gain exposure to discriminant analysis, clustering, factor analysis, and multidimensional scaling techniques.
3	Develop insight into non-Gaussian methods, independent component analysis, and feature selection strategies for high-dimensional data.
4	Familiarize with real-world applications of feature selection and understand emerging trends such as scalability, distributed methods, and real-time processing.

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	**Understand** the concepts of high-dimensional data, curse of dimensionality, and intrinsic versus extrinsic dimensions.
CO2	**Apply** discriminant analysis, clustering methods, and factor analysis to classify and group high-dimensional datasets.
CO3	**Analyze** feature extraction, independent component analysis, and dimensionality reduction techniques for data representation.
CO4	**Evaluate** feature selection methods and assess their effectiveness in real-world problems such as intrusion detection or medical diagnosis.
CO5	**Create** scalable and application-specific feature selection solutions for emerging challenges involving millions of dimensions and real-time data processing.

Course Description:

This course focuses on methods for handling and analyzing high-dimensional data, including dimensionality reduction, discriminant analysis, clustering, and factor analysis. It introduces non-Gaussian approaches, independent component analysis, and feature selection techniques. Applications to real-world problems and emerging trends such as scalability and real-time data processing are also highlighted.

Course Content		
Unit-1	**Introduction of High-Dimensional Data**	**4 Hours**

Overview of Dimensionality Reduction, High Dimension Data Acquisition, Curse of the Dimensionality, Intrinsic and Extrinsic Dimensions.		
Unit-2	**Discriminant Analysis**	**7 Hours**
Introduction, Classes, Labels, Rules and Decision Functions, Linear Discriminant Rules, Evaluation of Rules and Probability of Misclassification, Discrimination under Gaussian Assumptions, Bayesian Discrimination, Non-Linear, Non-Parametric and Regularised Rules, Principal Component Analysis, Discrimination and Regression		
Unit-3	**Factors and Grouping**	**7 Hours**
Norms proximities, features, and dualities: Vectors and matrix norms, measure of proximity, Features and feature maps, dualities of X and X Transpose Cluster analysis: Introduction, Hierarchal Agglomerative Clustering, k-means clustering, Principal component and cluster analysis Factor Analysis: Introduction, Population k-Factor model, Sample k-Factor model Multidimensional scaling: Introduction, Classical Scaling, Metric Scaling, Nonmetric scaling		
Unit-4	**Non-Gaussian Analysis**	**7 Hours**
Towards Non-Gaussianity: Introduction, Gaussianity and Independence, Skewness, Kurtosis and Cumulants, Entropy and Mutual Information, Training, Testing and Cross-Validation Independent Component Analysis: Introduction, Sources and Signals, Identification of the Sources, Mutual Information and Gaussianity, Estimation of the Mixing Matrix, Non-Gaussianity and Independence in Practice, Low-Dimensional Projections of High-Dimensional Data, Dimension Selection with Independent Components		
Unit-5	**Feature Selection for High-Dimensional Data**	**7 Hours**
The Need for Feature Selection, When Features Are Born, Intrinsic Characteristics of Data, Feature Selection, Feature Selection Methods		
Unit-6	**Application of Feature Selection to Real Problems and Emerging Trends Application of Feature Selection**	**4 Hours**
Classification in Intrusion Detection Systems, Tear Film Lipid Layer Classification, Cost-Based Feature Selection, Emerging Challenges: Millions of Dimensions, Scalability, Distributed Feature Selection, Real-Time Processing.		

Learning Resources:

Text Book

1	High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications by John Wright, Yi Ma, Cambridge University Press
2	Statistics for High-Dimensional Data: Methods, Theory and Applications by Peter Bühlmann, Sara van de Geer, Springer

Reference Books

1	High-Dimensional Covariance Estimation Mohsen Pourahmadi Wiley
2	Analysis of Multivariate and High-Dimensional Data by Inge Koch, Cambridge University Press
3	Statistical Inference from High-Dimensional Data by Carlos Fernández-Llano (Editor), MDPI
4	High-Dimensional Probability: An Introduction with Applications in Data Scienceby Roman Vershynin, Cambridge University Press
5	Fundamentals of High-Dimensional Statistics: With Exercises and R Labs by Johannes Lederer, Springer
6	High-Dimensional Statistics: A Non-Asymptotic Viewpoint by Martin J. Wainwright, Cambridge University Press

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSELC106P Course Name: Laboratory Course				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Practicals:	**04 Hours/Week**	**02**	**ISA:**	**25 Marks**
			POE:	**25 Marks**

Prerequisites:

1) Basics of Python/Java/C++ programming.
2) Knowledge of Data Mining or Machine Learning fundamentals.

Course Objectives:

1	Provide practical exposure to theoretical concepts learned in Mathematical Foundations and Machine Learning.
2	Develop hands-on skills in feature engineering, supervised/unsupervised learning, and complexity analysis.
3	Enable implementation of modern ML tools and frameworks on real-world datasets.
4	Integrate elective subject domains (NLP, IoT, Blockchain, Optimization, etc.) into small practical experiments.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Implement concepts of discrete mathematics, automata, and computational models through coding/simulation.
CO2	Apply feature engineering and machine learning algorithms to real-world datasets.
CO3	Evaluate and compare performance of regression, classification, and clustering models.
CO4	Integrate elective concepts such as NLP, Blockchain, or IoT into practical solutions.
CO5	Design a small-scale ML pipeline or simulation demonstrating end-to-end application.

Course Description:

This course is designed to strengthen practical understanding of mathematical foundations, feature engineering, machine learning algorithms, and program electives through implementation-based experiments.

Course Content

1	Set operations and relation properties (reflexive, symmetric, transitive)
2	Simulation of DFA/NFA and string acceptance
3	Pushdown Automata/CFG implementation for simple languages
4	Turing Machine simulation for palindrome or binary addition

5	Feature scaling, normalization, and encoding on dataset
6	PCA for dimensionality reduction on dataset
7	Linear & Polynomial Regression implementation with evaluation
8	Classification models: Logistic Regression, Decision Tree, KNN
9	Neural Network basics: single-layer and multi-class classification
10	Clustering (k-means, hierarchical) and association rule mining
11	Elective-based practical (e.g., NLP text preprocessing / Blockchain prototype / IoT data simulation)
12	Mini-project: End-to-end ML pipeline integrating feature engineering and ML model

Learning Resources:

Text Book

1	J.P. Trembley and R. Manohar – *Discrete Mathematical Structures with Applications to Computer Science*, McGraw Hill.
2	Sinan Ozdemir, Divya Susarla – *Feature Engineering Made Easy*, Packt Publishing.
3	Tom M. Mitchell – *Machine Learning*, McGraw Hill.

Reference Books

1	C.M. Bishop – Pattern Recognition and Machine Learning, Springer.
2	Max Kuhn, Kjell Johnson – Feature Engineering and Selection: A Practical Approach for Predictive Models, Chapman & Hall.
3	Kevin Murphy – Machine Learning: A Probabilistic Perspective, MIT Press.
4	C.M. Bishop – Pattern Recognition and Machine Learning, Springer.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSELC107T Course Name: Seminar-I				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Practicals:	**02 Hours/Week**	**01**	**ISA:**	**50 Marks**

Prerequisites:

1) Basic knowledge of technical communication and report writing.
2) Understanding of core engineering subjects in the area of specialization.
3) Familiarity with research papers, referencing styles, and presentation tools.

Course Objectives:

1	To enable students to carry out a detailed literature review on a focused research area.
2	To develop skills in technical writing, documentation, and report formatting (IEEE/standard styles).
3	To improve students' abilities to communicate scientific and engineering ideas effectively in oral and written forms.
4	To prepare students for research presentation, academic discussion, and dissertation planning.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Conduct a literature review and identify research gaps relevant to their area of specialization.
CO2	Organize and synthesize technical information effectively using appropriate referencing tools.
CO3	Prepare well-structured seminar reports in IEEE or other recognized technical formats.
CO4	Deliver professional oral presentations with clarity, coherence, and visual support.
CO5	Demonstrate teamwork, listening, and critical evaluation through peer seminar participation.

Course Description:

Each student will select a topic related to Design Engineering or their area of specialization. The topic should be based on a literature survey from reputed journals, conferences, and online academic databases and must help in identifying a potential M. Tech. dissertation topic. Students must prepare a technical report of 25–30 pages (A4 size) in IEEE format, submit two copies as term work, and present their seminar before the departmental faculty and peers. Assessment will be based on technical content, report quality, understanding, and presentation skills.

Course Content

Activity	Description	Expected Output
Topic Selection & Approval	Choose a topic relevant to Design/Engineering domain and get faculty approval.	Approved seminar topic.
Literature Review	Study at least 15–20 recent research papers from journals (IEEE, Elsevier, Springer).	Comprehensive review summary.
Seminar Report Preparation	Prepare a 25–30 page report in IEEE format covering objectives, methods, findings, and future scope.	Hard and soft copies of report.
Seminar Presentation	Present the seminar before faculty and peers using visual aids (slides, charts, etc.).	Oral presentation & Q&A.
Peer Attendance & Evaluation	Students must attend other seminars and actively participate.	Attendance & interaction marks.

Learning Resources:

Text Book	
1	M. Ashraf Rizvi – Effective Technical Communication, McGraw Hill Education India, 2nd Edition.
2	R.C. Sharma & Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw Hill, 5th Edition.
3	C.R. Kothari – Research Methodology: Methods and Techniques, New Age International Publishers, 4th Edition.
Reference Books	
1	S.P. Dhanavel – English and Communication Skills for Students of Science and Engineering, Orient BlackSwan, 2014.
2	Justin Zobel – Writing for Computer Science, Springer India, 3rd Edition.
3	Robert A. Day & Barbara Gastel – How to Write and Publish a Scientific Paper, Cambridge University Press, 8th Edition.
SWAYAM Courses	
1	NPTEL – Technical Communication for Scientists and Engineers (IIT Bombay).
2	NPTEL – Research Writing and Presentation Skills (IIT Madras).
3	SWAYAM – Communication Skills for Researchers (UGC).
Tools	
1	Access online tools such as Mendeley, Zotero, or LaTeX (Overleaf) for reference management and writing.
2	Practice oral presentation using PowerPoint or Google Slides with peer feedback.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPCC201 Course Name: Deep Learning				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Basics of Machine Learning and Neural Networks
2) Linear Algebra, Probability, and Optimization fundamentals
3) Python programming and exposure to ML frameworks (NumPy, TensorFlow, PyTorch preferred)

Course Objectives:

1	Introduce the fundamentals of deep learning and neural networks.
2	Explore architectures such as CNN, RNN, LSTM, GANs, and Autoencoders.
3	Provide practical exposure to building, training, and optimizing deep learning models.
4	Develop skills in applying deep learning for vision, sequential data, and generative tasks.
5	Enable understanding of challenges such as vanishing gradients, overfitting, and model generalization.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain the fundamentals of neural networks and deep learning architectures.
CO2	Apply feed-forward and convolutional neural networks for classification and recognition tasks.
CO3	Analyze sequential data using RNNs, LSTMs, and related architectures.
CO4	Design and implement generative models such as GANs and autoencoders.
CO5	Evaluate the performance of deep learning models and propose improvements using modern frameworks.

Course Description:

This course introduces the foundations and architectures of deep learning, focusing on neural networks, convolutional networks, recurrent models, and generative models. Students will learn both theoretical aspects and practical implementation techniques for solving real-world problems in vision, natural language, and generative modelling.

Course Content

Unit-1	**Introduction to Deep Learning**	**7 Hours**
Historical trends in Deep Learning, reasons for DL growth, Artificial Neural Networks, nonlinear		

classification examples such as XOR/XNOR, single and multiple layer perceptrons, feed forward networks, deep feed-forward networks, stochastic gradient–based learning, hidden units, architecture design, and backpropagation.		
Unit-2	**Convolutional Neural Networks (CNN)**	**7 Hours**
Introduction to CNNs and applications in computer vision, CNN basic architecture, activation functions including sigmoid, tanh, ReLU, and softmax layers, pooling layers and their types, training of CNNs in TensorFlow, popular CNN architectures such as VGG, GoogleNet, and ResNet, dropout, normalization, and data augmentation techniques.		
Unit-3	**Sequence Modelling**	**6 Hours**
Recurrent and recursive networks, unfolding computational graphs, recurrent neural networks, bidirectional RNNs, encoder-decoder sequence-to-sequence architectures, deep recurrent networks, recursive neural networks, and long short-term memory (LSTM) and other gated RNNs.		
Unit-4	**Recurrent Neural Networks (RNN)**	**7 Hours**
Introduction to RNNs and their applications in sequential data analysis, backpropagation through time (BPTT), vanishing gradient and gradient clipping, long short-term memory (LSTM) networks, gated recurrent units (GRUs), bidirectional LSTMs, and bidirectional RNNs.		
Unit-5	**Adversarial Networks (GANs)**	**6 Hours**
The Need for Feature Selection, When Features Are Born, Intrinsic Characteristics of Data, Feature Selection, Feature Selection Methods		
Unit-6	**Autoencoders**	**7 Hours**
Autoencoders and their architectures, encoder and decoder components, training autoencoders for data compression and reconstruction, relationship between autoencoders and GANs, and hybrid models including encoder-decoder GANs.		

Learning Resources:

Text Book

1	Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, MIT Press
2	Michael Nielsen – Neural Networks and Deep Learning, Determination Press
3	Satish Kumar – Neural Networks: A Classroom Approach, Tata McGraw Hill

Reference Books

1	François Chollet – Deep Learning with Python, Manning Publications
2	Charu Aggarwal – Neural Networks and Deep Learning: A Textbook, Springer

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPCC202 Course Name: Parallel Algorithm				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

Basics of Algorithms and Data Structures, Complexity analysis, Basic understanding of parallel computing concepts.

Course Objectives:

1	To introduce the need and motivation for parallel algorithms.
2	To study models of parallel computation and performance measures.
3	To design and analyze parallel algorithms for fundamental problems such as sorting, searching, selection, and graph problems.
4	To understand the trade-offs of time, space, and energy in parallel computing.
5	To apply parallel algorithmic strategies in real-world and research contexts.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain the motivation, models, and complexity measures for parallel algorithms.
CO2	Apply parallel computing techniques to sorting, searching, and selection problems.
CO3	Analyze the performance and efficiency of parallel algorithms across different models.
CO4	Design parallel algorithms for graph problems and evaluate their scalability.
CO5	Evaluate trade-offs in time, space, and energy complexity for parallel algorithm design.

Course Description:

This course introduces the design and analysis of parallel algorithms, covering foundational problems such as addition, sorting, selection, searching, and graph algorithms. Students will explore models of parallel computation, complexity measures, and performance evaluation. The course emphasizes both theoretical aspects and practical implications of parallelism in modern computing systems.

Course Content

Unit-1	**Introduction to Parallel Algorithms**	**6 Hours**

Motivation for parallelism, Models of parallel computation (EREW, CREW, SIMD, MIMD), Performance measures (speedup, efficiency, scalability), Parallel addition, Parallel implementation of Quick Sort, Basics of energy complexity, Asymptotic energy complexity derivation.

Unit-2	Parallel Selection	7 Hours
Sequential selection problem, Parallel selection algorithms, Parallel selection on EREW SM SIMD machines, Complexity analysis.		
Unit-3	**Parallel Searching**	**6 Hours**
Sequential search, Parallel search, K-ary search algorithm, Implementation on parallel models, Analysis of time and efficiency.		
Unit-4	**Parallel Sorting Algorithms**	**7 Hours**
Review of sequential sorting, Parallel Merge Sort, Parallel Quick Sort revisited, Bitonic Sort, Complexity analysis, Performance comparison across models.		
Unit-5	**Graph Algorithms in Parallel**	**6 Hours**
Parallel formulation for connected components, Parallel spanning tree algorithms, Parallel algorithms for maximum independent set, Parallel graph traversal methods (BFS/DFS in parallel).		
Unit-6	**Analysis and Applications**	**7 Hours**
Analysis of parallel algorithms (time, space, energy), Trade-offs and limitations, Applications of parallel algorithms in big data, image processing, machine learning, and scientific simulations.		

Learning Resources:

Text Book	
1	Selim G. Akl, Design and Analysis of Parallel Algorithms, Prentice Hall.
2	Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Pearson.
Reference Books	
1	Michael Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill.
2	Rajiv Chopra, Parallel Computing: Architectures, Algorithms and Applications,

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2031 Course Name: Cryptography and Network Security				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

Basic knowledge of Communication system and computer networks.

Course Objectives:

1	Explain different types of symmetric and asymmetric security techniques
2	Compare different types of cryptographic algorithms to ensure data integrity
3	Explain different types of security protocols in TCP/IP protocol suite
4	Understanding different types of security threats for computer system

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain classical encryption techniques and the basic model of network security.
CO2	Use symmetric and asymmetric cryptographic algorithms such as DES, AES, RSA, and Diffie-Hellman for secure communication.
CO3	Examine cryptographic authentication mechanisms including hash functions, MACs, and digital signature schemes to ensure data integrity.
CO4	Implement key management and user authentication protocols like X.509, PKI, and Kerberos.
CO5	Examine network and system security threats and identify appropriate mitigation strategies using firewalls, IDS, and security protocols.

Course Description:

This course introduces fundamental concepts of cryptography and network security, including classical and modern encryption, symmetric and asymmetric algorithms, key management, and authentication mechanisms. It covers internet and system security protocols such as SSL/TLS, SSH, IP security, intrusion detection, malware countermeasures, and firewalls. Students will analyze, implement, and evaluate security mechanisms to protect data and networks from evolving threats.

Course Content

Unit-1	**Foundations & Classical Encryption Techniques**	**6 Hours**

Overview of Information Security, Security Goals, Security Services and Mechanisms, OSI Security Architecture, Security Attacks, Symmetric Cipher Model, Classical Encryption Techniques – Substitution Techniques, Transposition Techniques, Basic Model for Network Security.

Unit-2	**Symmetric and Asymmetric Cryptography**	**7 Hours**
Block Cipher Structure, Data Encryption Standard (DES), Basic DES Example, Introduction to Advanced Encryption Standard (AES), Principles of Public-Key Cryptography, RSA Algorithm, Diffie-Hellman Key Exchange, Overview of Post-Quantum Cryptography and Motivation.		
Unit-3	**Cryptographic Authentication Functions**	**6 Hours**
Cryptographic Hash Functions, Applications of Hash Functions, Simple Hash Functions, Message Authentication Codes (MACs), HMAC, Digital Signatures, ElGamal Digital Signature, Digital Signature Standard (DSS).		
Unit-4	**Key Management and User Authentication**	**7 Hours**
Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public Key Infrastructure (PKI), Remote User Authentication Principles, Kerberos, Remote User Authentication Using Asymmetric Encryption.		
Unit-5	**Internet and Transport Security Protocols**	**6 Hours**
Transport-Level Security, SSL/TLS, HTTPS, Secure Shell (SSH), Email Security – PGP, S/MIME, IP Security Overview, IP Security Policy, Encapsulating Security Payload (ESP), Internet Key Exchange (IKE).		
Unit-6	**System Security and Threat Mitigation**	**7 Hours**
Intruders, Intrusion Detection Systems (IDS), Password Management, Malicious Software – Viruses, Worms, Trojans, Countermeasures, Firewalls, Trusted Systems, Basic Cloud Security Concepts, IoT Security Overview.		

Learning Resources:	
Text Book	
1	William Stallings – Cryptography and Network Security: Principles and Practices, 8th Edition, Pearson Education Limited, 2023
Reference Books	
1	Cryptography & Network Security B.A. Forouzan McGrawHill
2	Cryptography and network security – AtulKahate (TMGH)
3	Handbook of Applied Cryptography - Menezes, an Oorschot, and S.A. Vanstone

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2032 Course Name: Human Computer Interaction				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Basic knowledge of Computer Fundamentals and Operating Systems
2) Familiarity with Web Technologies (HTML/CSS/JavaScript) is helpful but not mandatory
3) Basic knowledge of Mathematics and Statistics for data visualization concepts

Course Objectives:

1	To understand the principles, importance, and evolution of Human–Computer Interaction (HCI) and User Interface (UI) design.
2	To apply human-centered design principles and effective screen layout techniques for creating intuitive interfaces.
3	To implement usability engineering, prototyping, and evaluation methods to enhance software usability.
4	To explore cognitive models and emerging technologies such as augmented reality, virtual reality, and ubiquitous computing for advanced interface design.

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Demonstrate an understanding of HCI principles, the evolution of user interfaces, and the importance of good design in software systems.
CO2	Apply human-centered design techniques, screen layout principles, and interface components to develop intuitive and effective user interfaces.
CO3	Use usability engineering, prototyping, and evaluation methods to analyze and improve the usability of software applications.
CO4	Design and develop innovative user interfaces by applying cognitive models and integrating emerging interaction technologies, including augmented reality, virtual reality, and ubiquitous computing.

Course Description:

This course provides an introduction to Human–Computer Interaction (HCI) and User Interface Design, focusing on the principles of usability, screen design, interface components, and evaluation techniques. It covers human-centered design, prototyping, cognitive models, and explores emerging technologies such as ubiquitous computing, augmented reality, and data visualization to prepare students for designing effective and user-friendly interactive systems.

Course Content

Unit-1	Introduction	6 Hours
Importance of user Interface –definition, importance of good design Benefits of good design. A brief		

history of Screen design. The graphical user Interface – popularity of graphics, the concept of direct manipulation, graphical System, Characteristics, Web user – Interface popularity, characteristics-Principles of user interface.		
Unit-2	**Design Process and Screen Design Principles**	**6 Hours**
Design process – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, and understanding business junctions. Screen Designing: Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and f low – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.		
Unit-3	**Windows, Controls, and Interface Components**	**6 Hours**
Windows – New and Navigation schemes selection of window, selection of devices based and screen- based controls. Components – text and messages, Icons and increases – Multimedia, colours, uses problems, choosing colours.		
Unit-4	**HCI in Software Process: Design Rules and Evaluation**	**6 Hours**
HCI in the software process, the software life cycle Usability engineering Iterative design and prototyping Design Focus: Prototyping in practice Design rationale Design rules Principles to support usability Standards Golden rules and heuristics HCI patterns Evaluation techniques, Goals of evaluation, Evaluation through ex-pert analysis, Evaluation through user participation, Choosing an evaluation meth-od. Universal design, Universal design principles Multi-modal interaction		
Unit-5	**Cognitive Models and Emerging Interaction Technologies**	**6 Hours**
Cognitive models Goal and task hierarchies Design Focus: GOMS saves money Linguistic models The challenge of display-based systems Physical and device models Cognitive architectures Ubiquitous computing and augmented realities Ubiquitous computing applications research Design Focus: Ambient Wood – augmenting the physical Virtual and augmented reality Design Focus: Shared experience Design Focus: Applications of augmented reality Information and data visualization Design Focus: Getting the size right.		

Learning Resources:

Text Book	
1	The essential guide to user interface design, Wilbert O Galitz, Wiley Dream Tech.
2	Human – Computer Interaction. Alan Dix, Janet Fincay, Gre Goryd, Abowd, Russell Bealg, Pearson Education.
Reference Books	
1	Designing the user interface. 3rd Edition Ben Shneidermann, Pearson Education Asia.
2	Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
3	User Interface Design, Soren Lauesen , Pearson Education

4	Human –Computer Interaction, D. R. Olsen, Cengage Learning.
5	Human –Computer Interaction, Smith - Atakan, Cengage Learning.
Web Reference	
1	https://onlinecourses.nptel.ac.in/noc25_cs38/preview
2	https://onlinecourses.nptel.ac.in/noc25_cs135/preview

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2033 Course Name: Parallel Computing				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Operating System
2) Computer Architecture
3) Computer Networks

Course Objectives:

1	Understand the motivation, models, and challenges of parallel computing and supercomputers.
2	Gain proficiency in message-passing paradigms and MPI programming for parallel systems.
3	Learn communication networks, interconnection topologies, and routing mechanisms for efficient parallelism.
4	Analyze performance factors including scalability, benchmarks, profiling, and network impact.
5	Design efficient parallel applications considering decomposition, load balancing, adaptivity, and parallel I/O bottlenecks.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain fundamental concepts of parallel computing, shared/distributed memory, and Amdahl's law
CO2	Apply MPI communication primitives to implement parallel algorithms
CO3	Analyze communication topologies, routing strategies, and mapping techniques in parallel systems
CO4	Evaluate scalability, benchmarking results, and profiling outcomes of parallel applications
CO5	Design parallel applications addressing decomposition, load balancing, and parallel I/O challenges.

Course Description:

Parallel programming is ubiquitous in today's multi-core era and solves many real-world scientific problems. Massive parallelism entails significant hardware and software challenges. The course is structured so that the participants understand challenges in efficient execution of large-scale parallel applications. The assignments will be designed to strengthen understanding of parallel programming.

Course Content

Unit-1	**Introduction**	**5 Hours**

Why parallel computing? Shared memory and distributed memory parallelism, Amdahl's law, speedup and efficiency, supercomputers.		
Unit-2	**Message passing**	**8 Hours**
MPI basics, point-to-point communication, collective communication, synchronous/asynchronous send/recv, algorithms for gather, scatter, broadcast, reduce.		
Unit-3	**Parallel communication**	**6 Hours**
Windows – New and Navigation schemes selection of window, selection of devices based and screen- based controls. Components – text and messages, Icons and increases – Multimedia, colours, uses problems, choosing colours.		
Unit-4	**Performance**	**7 Hours**
Scalability, benchmarking, performance modeling, impact of network topologies, parallel code analysis and profiling.		
Unit-5	**Designing parallel codes**	**7 Hours**
Domain decomposition, communication-to-computation ratio, load balancing, adaptively, case studies: weather and material simulation codes		
Unit-6	**Parallel I/O**	**6 Hours**
MPI I/O algorithms, contemporary large-scale I/O architecture, I/O bottlenecks.		

Learning Resources:

Text Book	
1	Peter S Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann, 2011.
2	DE Culler, A Gupta and JP Singh, Parallel Computer Architecture: A Hardware/Software Approach Morgan-Kaufmann, 1998.
3	Marc Snir, Steve W. Otto, Steven Huss-Lederman, David W. Walker and Jack Dongarra, MPI - The Complete Reference, Second Edition, Volume 1, The MPI Core.
4	William Gropp, Ewing Lusk, Anthony Skjellum, Using MPI : portable parallel programming with the message-passing interface, 3rd Ed., Cambridge MIT Press, 2014.
5	A Grama, A Gupta, G Karypis, and V Kumar, Introduction to Parallel Computing. 2nd Ed., Addison-Wesley, 2003.
Reference Books	
1	JL Hennessy and DA Patterson, Computer Architecture: A Quantitative Approach, 4th Ed., Morgan Kaufmann/Els India, 2006.
2	MJ Quinn, Parallel Computing: Theory and Practice, Tata McGraw Hill, 2002.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2041 Course Name: Quantum Computing				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Basic understanding of Linear Algebra and Complex Numbers.
2) Knowledge of Classical Computing and Algorithms.

Course Objectives:

1	Understand the limitations of classical computing and the need for quantum computing.
2	Learn the basic principles of quantum mechanics relevant to computation.
3	Explore quantum bits, gates, circuits, and simple quantum algorithms.
4	Study the fundamentals of quantum information and cryptography.
5	Gain awareness of quantum computing hardware and real-world applications.

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Explain the basic concepts and significance of quantum computing.
CO2	Describe fundamental quantum mechanical principles relevant to computation.
CO3	Apply quantum gates and circuit models to represent simple quantum algorithms.
CO4	Analyze quantum computing models and their advantages over classical models.
CO5	Evaluate quantum error correction techniques and their role in reliable quantum computation.

Course Description:

This course introduces the fundamental principles of quantum computing, including quantum mechanics concepts, quantum bits, gates, and circuits. Students will explore quantum algorithms, information theory, and error correction techniques. The course also highlights the comparison between classical and quantum computing models and provides insights into quantum hardware and real-world applications.

Course Content

Unit-1	Introduction to Quantum Computing	7 Hours

Classical computing vs. quantum computing, concept of qubits and quantum states, features of quantum computing including superposition, entanglement and interference, advantages of quantum computing in solving complex problems, real-world applications and significance in modern computer science.

Unit-2	Basics of Quantum Mechanics	7 Hours
Introduction to quantum mechanics with simple examples, basic linear algebra concepts such as vectors, matrices and complex numbers, quantum state representation, quantum measurement principles, postulates of quantum mechanics and their interpretations, operators and probability amplitudes.		
Unit-3	**Quantum Circuits and Gates**	**7 Hours**
Quantum logic gates: X, Y, Z, Hadamard, Phase, and CNOT, construction of simple quantum circuits using standard gates, visual representation through quantum circuit diagrams, concept of reversible computing, comparison of classical and quantum gate operations.		
Unit-4	**Quantum Algorithms**	**7 Hours**
Introduction to key quantum algorithms and their importance, conceptual overview of quantum Fourier transform, basics of Grover's search algorithm, introduction to Shor's algorithm and its application to integer factorization, case studies demonstrating quantum speed-up.		
Unit-5	**Quantum Information and Error Handling**	**7 Hours**
Quantum noise and sources of decoherence, basics of quantum error correction with simple examples, density matrices and Bloch sphere representation, introduction to entropy in quantum systems, challenges in building and maintaining stable quantum environments.		
Unit-6	**Quantum Cryptography and Future Trends**	**7 Hours**
Basics of quantum communication and security, quantum key distribution (QKD) and BB84 protocol, conceptual explanation of quantum teleportation, applications in cryptography and cybersecurity, overview of emerging platforms like IBM Q and Google Sycamore, introduction to quantum cloud services.		

Learning Resources:	
Text Book	
1	Michael A. Nielsen, Issac L. Chuang, "Quantum Computation and Quantum Information", Tenth Edition, Cambridge University Press, 2010.
Reference Books	
1	Scott Aaronson, "Quantum Computing Since Democritus", Cambridge University Press, 2013.
2	N. David Mermin, "Quantum Computer Science: An Introduction", Cambridge University Press, 2007.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2042 Course Name: Pattern Recognition				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Linear Algebra and Probability
2) Data Structures and Algorithms
3) Basic Machine Learning / Artificial Intelligence concepts

Course Objectives:

1	Gain a broad perspective on the fundamental principles of pattern recognition, classification, and clustering methods.
2	Develop familiarity with probabilistic, statistical, and machine learning approaches used in pattern classification and analysis.
3	Explore techniques of feature selection, feature generation, and dimensionality reduction for efficient representation of patterns.
4	Acquire insights into the design of advanced pattern recognition systems and their applications in domains such as image processing, speech recognition, and information retrieval.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Understand the concepts of pattern recognition, learning paradigms, and applications.
CO2	Apply Bayes decision theory, linear and nonlinear classifiers, and clustering algorithms to solve classification problems.
CO3	Analyze various feature extraction and dimensionality reduction techniques for pattern representation.
CO4	Evaluate the performance of classifiers, clustering methods, and template matching approaches for different datasets.
CO5	Create context-dependent classification models using Markov models, Hidden Markov Models, and neural networks for real-world problems.

Course Description:

This course introduces the fundamental concepts of pattern recognition and machine learning for data analysis and classification. It covers supervised, unsupervised, and semi-supervised learning methods, classifiers, clustering algorithms, and feature extraction techniques. Students will also learn advanced approaches such as template matching and Hidden Markov Models with applications in image, speech, and information retrieval.

Course Content

Unit-1	Introduction	4 Hours
What is Pattern recognition, Features, clustering vs. Classification, Supervised, Unsupervised, and Semi-Supervised Learning, Pattern Recognition Applications, Patterns and Feature Extraction with Examples.		
Unit-2	**Classifiers based on Bayes Decision Theory**	**7 Hours**
Bayes Decision Theory: Introduction, Bayes Decision Theory, Discriminant Functions and Decision Surfaces, Bayesian Classification for Normal Distribution, Estimation of Unknown Probability Density Functions, The Nearest Neighbor Rule, Bayesian Belief Networks Linear and Nonlinear Classifiers: Linear Discriminant Functions and Decision Hyperplanes, The Perceptron Algorithm, Least Squares Methods, Mean Square Estimation Revisited, Logistic Discrimination, Support Vector Machine The XOR Problem, The Two and Three Layer Perceptrons, Algorithm Based On Exact Classification of Training Set, The Back propagation Algorithm, The Cost Function Choice, Choice of the Network Size, Networks with Weight Sharing, Generalized Linear Classifiers, Polynomial Classifiers, Radial Basis Function Networks, Support Vector Machines		
Unit-3	**Clustering**	**7 Hours**
Clustering Basic Concepts, Clustering Algorithms: Sequential and Hierarchical algorithms.		
Unit-4	**Feature Selection and Generation**	**7 Hours**
Feature Selection: Introduction, Pre-processing, Feature Selection Based on Statistical Hypothesis Testing, The Receiver Operating Characteristics (ROC) Curve, Class Separability Measures, Feature Subset Selection, Optimal Feature Generation, Feature Generation/Selection, Feature Generation: Data Transformation and Dimensionality Reduction, Basis Vectors and Images , The Karhunen-Loeve Transform , The Singular Value Decomposition , Independent Component Analysis , Nonnegative Matrix Factorization, Nonlinear Dimensionality Reduction, The Discrete Fourier Transform (DFT) , The Discrete Cosine and Sine Transforms , The Hadamard Transfor, The Haar Transform , Discrete Time Wavelet Transform (DTWT), Applications Regional Features, Features for Shape and Size Characterization, A Glimpse at Fractals, Typical Features for Speech and Audio Classification		
Unit-5	**Template Matching**	**7 Hours**
Introduction, Measures Based on Optimal Path Searching Techniques, Measures Based on Correlations, Deformable Template Models, Content-Based Information Retrieval: Relevance Feedback.		
Unit-6	**Context Dependent Classification**	**4 Hours**
Introduction, The Bayes Classifier, Markov Chain Models, The Viterbi Algorithm, Channel Equalization, Hidden Markov Models, HMM with State Duration Modeling , Training Markov Models via Neural Networks , A discussion of Markov Random Fields.		
Learning Resources:		

Text Book	
1	Pattern Recognition by Sergios Theodoridis and Konstantinos Koutroumbas, Elsevier
2	Pattern Recognition Statistical, Structural and Neural Approaches by Robert Schalkoff, Wiley
3	Pattern Recognition and Machine Learning by Christopher Bishop, Springer
4	Neural networks for pattern recognition by Christopher M. Bishop, Oxford University Press
Reference Books	
1	Pattern Recognition: A Statistical Approach by P.A Devijver and J. Kittler, Prentice-Hall International
2	Introduction to Statistical Pattern Recognition, by K. Fukunaga, Academic Press.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEPE2043 Course Name: Deep Generative Model				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Probability and Statistics
2) Machine Learning
3) Deep Learning

Course Objectives:

1	Introduce the fundamental principles of generative modeling
2	Explore and derive the mathematical foundations of key generative models
3	Critically evaluate training strategies, convergence challenges, and evaluation metrics
4	Examine ethical considerations, societal impact, and current research trends in generative AI

Course Outcomes:

COs	At the end of the successful completion of the course, the students will be able to
CO1	Explain the theoretical foundations of generative modeling.
CO2	Derive and analyze key architectures of deep generative models.
CO3	Evaluate generative models using theoretical and quantitative metrics.
CO4	Compare and contrast different generative model families in terms of structure, training stability, interpretability, and data generation performance.
CO5	Critically assess the ethical, social, and research implications of generative AI, and identify applications and current trends in text, vision, and multimodal generation.

Course Description:

This course explores the theory and practice of generative deep learning models, including VAEs, GANs, autoregressive, flow-based, and diffusion models. It covers applications across vision, NLP, and audio, alongside ethical considerations and recent research trends in generative AI.

Course Content

Unit-1	Introduction to Generative Modeling	6 Hours

Discriminative vs. Generative Models, Joint and Conditional Probability Distributions, Latent Variable Models, Overview of Generative Models: VAE, GANs, Autoregressive, Diffusion Applications in Vision, NLP, Audio, and Bioinformatics.

Unit-2	Classifiers based on Bayes Decision Theory	6 Hours
Autoencoders: Vanilla and Denoising, Variational Inference and Latent Variables Evidence Lower Bound (ELBO) and KL Divergence, Reparameterization Trick, Extensions: Conditional VAEs, β-VAEs.		
Unit-3	**Clustering**	**6 Hours**
GAN Architecture: Generator and Discriminator, Adversarial Loss and Min-Max Game, Training Challenges: Non-Convergence, Mode Collapse, Theoretical Analysis of GAN Training, Variants: DCGAN, Conditional GANs (cGAN), WGAN, WGAN-GP.		
Unit-4	**Feature Selection and Generation**	**6 Hours**
Autoregressive Modeling and Maximum Likelihood, PixelRNN and PixelCNN, Sequence Models: LSTM, GRU, Transformer (Intro Only), Flow-Based Models: NICE, RealNVP, Glow Exact Likelihood and Invertible Transformations.		
Unit-5	**Diffusion Models & Evaluation Techniques I**	**6 Hours**
Introduction to Score-Based Generative Modeling, Forward and Reverse Diffusion Processes, Denoising Diffusion Probabilistic Models (DDPM), Evaluation Metrics: Inception Score (IS), Fréchet Inception Distance (FID), Precision/Recall Curves, Likelihood Estimation Metrics.		
Unit-6	**Applications, Ethics & Research Trends**	**8 Hours**
Applications: ChatGPT, DALL·E, Music Generation, Drug Discovery Deepfakes, Synthetic Data, and Generative AI in Art Hallucination, Bias, Fairness, and Alignment Responsible Deployment and Regulatory Challenges Survey of Recent Research and Open Problems.		

Learning Resources:

Text Book

1	Generative Deep Learning, David Foster, O’Reilly
2	Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press

Reference Books

1	“Generative Adversarial Nets” , Goodfellow et al. 2014
2	“Auto-Encoding Variational Bayes”, Kingma & Welling 2013

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEOE2051 Course Name: Ethical AI & Explainability				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Artificial Intelligence
2) Machine Learning or Data science fundamentals
3) Probability, Statistics, and Linear Algebra

Course Objectives:

1	Understand Ethical Principles in AI Design and Deployment
2	Identify and Mitigate Bias and Inequities in AI Systems
3	Apply Explainable AI (XAI) Techniques and Tools
4	Evaluate AI Systems Through Ethical and Regulatory Frameworks

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain key ethical concepts and frameworks applicable to the development and deployment of AI systems.
CO2	Identify sources of algorithmic bias and apply appropriate fairness metrics and mitigation techniques.
CO3	Demonstrate the use of Explainable AI (XAI) methods to interpret machine learning models.
CO4	Critically evaluate AI systems based on ethical, legal, and societal implications using real-world case studies.
CO5	Propose ethically sound and explainable AI solutions in response to complex, domain-specific challenges.

Course Description:

This course explores ethical issues in AI, such as bias, fairness, accountability, and transparency, with a focus on explainability and real-world case studies.

Course Content

Unit-1	**Foundations of Ethical AI**	**6 Hours**

Introduction to AI Ethics, History and evolution of ethical thought in AI, Core ethical principles: fairness, accountability, transparency, and privacy, Moral frameworks: utilitarianism, deontology, virtue ethics, Case studies: real-world ethical failures in AI.

Unit-2	Bias, Fairness, and Accountability	6 Hours
Types of biases in datasets and algorithms, Measuring and mitigating algorithmic bias, Fairness criteria in machine learning, Disparate impact and treatment, Legal and societal implications of biased AI systems, Tools for fairness assessment (e.g., IBM AI Fairness 360).		
Unit-3	**Explainability in AI Systems**	**6 Hours**
Importance and need for explainable AI (XAI), Human vs. model interpretability, Desirable properties of explanations (e.g., fidelity, interpretability, completeness), Explainability trade-offs (accuracy vs. transparency), Regulations and standards (e.g., GDPR "right to explanation").		
Unit-4	**Techniques and Tools for Explainable AI**	**8 Hours**
Post-hoc explanation techniques: LIME (Local Interpretable Model-agnostic Explanations),SHAP (SHapley Additive exPlanations),Counterfactual explanations, Partial dependence plots, Interpretable models: Decision trees, rule-based systems, linear models, Visualization and storytelling in explanations, Toolkits: LIME, SHAP, Interpret, ELI5.		
Unit-5	**Domain-Specific Challenges and Case Studies**	**6 Hours**
Healthcare: Explainability in diagnostic systems, Finance: Fair lending and credit scoring, Criminal justice: Recidivism prediction (e.g., COMPAS), Autonomous systems and robotics, Use of XAI in hiring and human resources.		
Unit-6	**Policy, Governance, and the Future of Ethical AI**	**6 Hours**
AI ethics guidelines (e.g., OECD, UNESCO, EU AI Act), Organizational AI governance and ethics boards, Standards and certifications (e.g., IEEE, ISO) Ethical AI in practice: risk assessment and impact evaluation.		

Learning Resources:	
Text Book	
1	“Ethics of Artificial Intelligence and Robotics" – edited by Vincent C. Müller (Part of the Stanford Encyclopedia of Philosophy)
2	Weapons of Math Destruction" – Cathy O’Neil
3	"Interpretable Machine Learning" – Christoph Molnar
Reference Books	
1	"Responsible Artificial Intelligence" – Virginia Dignum
2	"The Ethical Algorithm" – Michael Kearns & Aaron Roth

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEOE2052 Course Name: Computer Vision				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Artificial Intelligence, Machine Learning or Data science fundamentals
2) Probability, Statistics, and Linear Algebra

Course Objectives:

1	To introduce the fundamental concepts and techniques of computer vision.
2	To understand image formation, processing, and feature extraction.
3	To apply computer vision algorithms for object detection and recognition.
4	To evaluate and analyze modern applications of vision in AI and industry.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Understand image formation, representation, and transformation techniques.
CO2	Apply image processing methods for feature extraction and segmentation.
CO3	Evaluate object detection, recognition, and classification techniques.
CO4	Analyze motion, 3D vision, and tracking methods.
CO5	Assess real-world applications of computer vision in AI, robotics, and industry.

Course Description:

This course introduces the fundamental concepts and techniques of computer vision, focusing on image formation, processing, and analysis. Students will learn feature extraction, segmentation, object detection, and recognition methods along with advanced topics such as motion, 3D vision, and tracking. The course emphasizes both theoretical foundations and real-world applications in AI, robotics, and industry.

Course Content

Unit-1	**Fundamentals of Computer Vision**	**6 Hours**
Introduction to Computer Vision and its applications, Image formation, imaging geometry, and camera models, Digital image representation, pixel models, and sampling.		
Unit-2	**Image Pre-processing and Enhancement**	**6 Hours**

Image enhancement (spatial and frequency domain), Filtering, smoothing, and edge detection, Histogram equalization and image restoration.		
Unit-3	**Feature Detection and Description**	**6 Hours**
Corner detection (Harris, FAST), Feature descriptors (SIFT, SURF, ORB), Matching techniques and applications.		
Unit-4	**Segmentation and Classification**	**6 Hours**
Thresholding, region growing, clustering (k-means, mean-shift), Edge-based and region-based segmentation, Introduction to image classification using traditional ML methods.		
Unit-5	**Object Detection & Recognition**	**6 Hours**
Template matching, Viola–Jones face detection, Object detection using HOG and SVM, Introduction to deep learning for object detection (CNN basics, YOLO/R-CNN overview).		
Unit-6	**Advanced Topics & Applications**	**8 Hours**
Motion analysis, optical flow, and tracking, 3D vision and stereo imaging, Applications: autonomous vehicles, medical imaging, biometrics, AR/VR, industrial inspection.		

Learning Resources:	
Text Book	
1	Richard Szeliski, Computer Vision: Algorithms and Applications, Springer.
2	Rafael C. Gonzalez & Richard E. Woods, Digital Image Processing, Pearson.
Reference Books	
1	David Forsyth & Jean Ponce, Computer Vision: A Modern Approach, Pearson.
2	Jan Erik Solem, Programming Computer Vision with Python, O’Reilly.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSEOE2053 Course Name: High Performance Computing for Multidisciplinary Research				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Lectures:	**03 Hours/Week**	**03**	**ISE:**	**40 Marks**
			ESE:	**60 Marks**

Prerequisites:

1) Operating System,
2) Computer Organization and Microcontroller,
3) Digital System and Microprocessor

Course Objectives:

1	To introduce the fundamental concepts and motivation behind parallel and high-performance computing systems.
2	To understand different parallel hardware architectures and software models used in HPC.
3	To develop skills in distributed and shared memory programming using MPI, Pthreads, and OpenMP.
4	To provide practical knowledge of GPU and accelerator programming using CUDA and OpenCL.
5	To explore the applications and impact of high-performance computing across multiple scientific and engineering disciplines.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Explain the need for parallel computing and differentiate between parallel, concurrent, and distributed systems.
CO2	Analyze parallel hardware architectures and select appropriate programming models for given problems.
CO3	Develop and evaluate distributed-memory parallel programs using MPI.
CO4	Implement shared-memory parallel programs using Pthreads and OpenMP, ensuring correctness and performance.
CO5	Apply GPU and accelerator programming techniques to solve computational problems in multidisciplinary domains.

Course Description:

This course introduces the principles of high performance and parallel computing, focusing on architectures and programming models. It covers distributed and shared memory programming using MPI, Pthreads, and OpenMP, along with GPU and accelerator programming using CUDA and OpenCL. The course emphasizes performance evaluation and scalability of parallel applications. Applications of HPC in multidisciplinary research are also explored.

Course Content

Unit-1	Parallel Computing with Hardware and Software	7 Hours
Why Parallel Computing? Why We Need Ever-Increasing Performance, Why We're Building Parallel Systems, Why We Need to Write Parallel Programs, How Do We Write Parallel Programs?, What We'll Be Doing, Concurrent, Parallel, Distributed Parallel Hardware and Parallel Software Parallel Hardware: SIMD systems, MIMD systems, Interconnection networks, Cache coherence Parallel Software: Caveats, Coordinating the processes/threads, Shared-memory, Distributed-memory, Programming hybrid systems.		
Unit-2	**Distributed-Memory Programming with MPI**	**7 Hours**
Getting started: Compilation and execution, MPI programs, SPMD programs, Communication, Message matching, Status argument, semantics of MPI_send and MPI_recv Dealing with I/O: Output, Input, Collective Communication: Tree-structured communication, MPI_Reduce, Collective vs. point-to-point communications, MPI_Allreduce, Broadcast, Data distributions, MPI Derived Datatypes, Performance Evaluation of MPI Programs: Taking timings, Results, Speedup and efficiency, Scalability.		
Unit-3	**Shared-Memory Programming with Pthreads and OpenMP**	**6 Hours**
Shared-Memory Programming with Pthreads: Processes, Threads, and Pthreads, Hello World, Matrix-Vector Multiplication, Critical Sections, Busy-Waiting, Mutexes Shared-Memory Programming with OpenMP Getting Started: Compiling and running OpenMP programs, The program, Error Checking, Scope of variables, The reduction clause ,The parallel for Directive, More about loops in OpenMP , Scheduling Loops.		
Unit-4	**GPU programming: CUDA**	**6 Hours**
Introduction, CUDA's programming model: threads, blocks, and grids, CUDA's execution model: streaming multiprocessors and warps, CUDA compilation process, Putting together a CUDA project, Memory hierarchy.		
Unit-5	**GPU and accelerator programming: OpenCL**	**5 Hours**
The OpenCL architecture, The platform model, The execution model, The programming model, The memory model.		
Unit-6	**High Performance Computing for Multiple Disciplines**	**5 Hours**
High Performance Computing Disciplines, Impact of Supercomputing on Science, Society, and Security, Anatomy of a Supercomputer, Computer Performance.		
Learning Resources:		
Text Book		
1	An Introduction to Parallel Programming by Peter S. Pacheco, First Edition,Elsevier, 2011.(Unit 1,2,3)	
2	Multicore and GPU Programming: An Integrated Approach by Gerassimos Barlas, Second Edition, Morgan Kaufmann (Unit 4,5)	

3	High Performance Computing: Modern Systems and Practices by Thomas Sterling, Matthew Anderson, Maciej Brodowicz, Morgan Kaufmann (Unit 6)
Reference Books	
1	Parallel computing theory and practice by Michel J. Quinn by TMH
2	Computer Architecture & Parallel Processing by Kai Hwang & Briggs, McGraw Hill.
3	Parallel and Distributed Systems by Arun Kulkarni, Napur Prasad Giri, Wiley Publications, 2nd Edition

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSELC206T Course Name: Laboratory Course (Semester II)				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Practical:	**04 Hours/Week**	**02**	**ISA:**	**25 Marks**

Prerequisites:

1) Basics of Python programming with exposure to ML/DL libraries (NumPy, TensorFlow, PyTorch).
2) Knowledge of Algorithm Design, Complexity Analysis, and basics of Parallel Computing.

Course Objectives:

1	To provide practical exposure to Deep Learning and Parallel Algorithms.
2	To develop implementation skills for advanced DL architectures, parallel algorithms, and elec-tive-based applications.
3	To apply concepts from Program Electives (Cryptography, HCI, Pattern Recognition, Quantum, Generative Models) and Open Electives (Ethical AI, Computer Vision, HPC).
4	To evaluate scalability, performance, and ethical aspects of implemented models.
5	To integrate multi-domain knowledge through a mini-project.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Implement neural networks, CNNs, RNNs, and advanced DL models using frameworks.
CO2	Apply and analyze parallel algorithms for sorting, searching, and graph problems.
CO3	Integrate elective concepts (Cryptography, HCI, Pattern Recognition, Quantum, Generative Models) into practical tasks.
CO4	Evaluate models and algorithms in terms of scalability, efficiency, and ethical implications.
CO5	Design and demonstrate a mini-project integrating DL, parallelism, and elective concepts.

Course Description:

This laboratory course emphasizes implementation of Deep Learning models, parallel algorithms, and elective-based experiments, while encouraging interdisciplinary applications. Students will develop hands-on expertise in frameworks such as TensorFlow, PyTorch, and parallel computing platforms, culminating in a mini-project.

Course Content

1	Implement feed-forward neural network on a small dataset
2	Build a CNN for image classification (e.g., MNIST/CIFAR-10)

3	Implement RNN/LSTM for sequence modeling (e.g., text generation)
4	Train and evaluate a GAN or Autoencoder for image generation
5	Parallel addition and parallel Quick Sort
6	Parallel searching (k-ary search) and selection problem
7	Parallel graph algorithms: connected components / maximum independent set
8	Compare performance of sequential vs parallel sorting/searching
9	Elective-based practical (e.g., Cryptography with RSA, HCI prototype, Quantum algorithm simulation, Pattern Recognition classifier, Deep Generative Models)
10	Open Elective-based practical (e.g., Ethical AI bias detection, Computer Vision object detection, HPC performance benchmarking)
11	Case study: Analyze ethical, performance, or scalability trade-offs in chosen domain
12	Mini-project: End-to-end application integrating DL + Parallelism + Elective concept

Learning Resources:

Text Book

1	Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, MIT Press.
2	Ananth Grama et al. – Introduction to Parallel Computing, Pearson.
3	Relevant elective textbooks (Cryptography, HCI, Quantum, Pattern Recognition, Generative Models).

Reference Books

1	François Chollet – *Deep Learning with Python*, Manning Publications.
2	Selim G. Akl – *Design and Analysis of Parallel Algorithms*, Prentice Hall.
3	Research papers / online resources for electives and open electives.
4	C.M. Bishop – *Pattern Recognition and Machine Learning*, Springer.

Tatyasaheb Kore Institute of Engineering and Technology First Year M. Tech. Computer Science Engineering				
Course Code: 2501PCSESW207T Course Name: Seminar II				
Teaching Scheme		**Credit**	**Evaluation Scheme**	
Practicals:	**02 Hours/Week**	**01**	**ISA:**	**50 Marks**

Prerequisites:

1) Knowledge of core specialization subjects and familiarity with research methodologies.

Course Objectives:

1	To develop advanced skills in reviewing literature and identifying specific research gaps for dissertation.
2	To strengthen technical writing and formatting skills in line with IEEE/SCI indexed publications.
3	To train students in preparing extended seminar reports and pre-dissertation proposals.
4	To enhance oral communication and professional presentation abilities with effective use of visuals and tools.

Course Outcomes:

COs	**At the end of the successful completion of the course, the students will be able to**
CO1	Identify and analyze advanced research problems aligned with dissertation areas.
CO2	Critically evaluate literature, identify limitations, and synthesize findings for framing research scope.
CO3	Prepare structured seminar reports (30–40 pages) with technical rigor, proper referencing, and professional formatting.
CO4	Deliver impactful technical presentations with clarity, coherence, and effective handling of Q&A.
CO5	Engage in peer review, teamwork, and provide constructive evaluation of others' seminars.

Course Description:

Each student will select an advanced topic closely related to their M.Tech specialization and dissertation area. The seminar will require an in-depth literature review from reputed journals (IEEE, Elsevier, Springer, ACM, etc.), critical analysis of existing research, and formulation of possible research directions. Students must prepare a 30 - 40 page seminar report in IEEE format, submit two copies (soft + hard), and present their work before the departmental faculty and peers. Evaluation will consider technical depth, quality of documentation, presentation skills, and peer participation.

Course Content

Activity	Description	Expected Output
Topic Selection & Approval	Choose advanced research-oriented topic aligned with dissertation.	Approved seminar topic.
Extended Literature Review	Study at least 20–25 research papers from reputed sources, identify gaps.	Review matrix & summary.
Seminar Report Preparation	Prepare 30 - 40 page detailed report in IEEE format including methodology & research.	Hard and soft copies of report.
Seminar Presentation	Present seminar using professional slides, defend through Q&A.	Oral presentation & Q&A record.
Peer Review Participation	Attend peer seminars, provide constructive feedback, participate in academic discussion.	Peer review marks & participation.

Learning Resources:

Text Book

1	M. Ashraf Rizvi – Effective Technical Communication, McGraw Hill Education India, 2nd Edition.
2	R.C. Sharma & Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw Hill, 5th Edition.
3	C.R. Kothari – Research Methodology: Methods and Techniques, New Age International Publishers, 4th Edition.

Reference Books

1	Justin Zobel – Writing for Computer Science, Springer India, 3rd Edition.
2	Robert A. Day & Barbara Gastel – How to Write and Publish a Scientific Paper, Cambridge University Press, 8th Edition.
3	S.P. Dhanavel – English and Communication Skills for Students of Science and Engineering, Orient BlackSwan, 2014.