VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
(AUTONOMOUS) (Sponsored by Siddhartha Academy of General & Technical Education)
Approved by AICTE | Affiliated to JNTUK Kakinada | An ISO 9001:2015 Certified Institution

Institute Vision

To nurture excellence in various fields of engineering by imparting timeless core values to the learners and to mould the institution into a centre of academic excellence and advanced research.

Institute Mission

To impart high quality technical education in order to mould the learners into globally competitive technocrats who are professionally deft, intellectually adept and socially responsible. The institution strives to make the learners inculcate and imbibe pragmatic perception and proactive nature so as to enable them to acquire a vision for exploration and an insight for advanced enquiry.

Department Vision

The department vision is clearly defined and is in line with the college's vision. The vision of the department is "To evolve as a centre of academic excellence and advanced research in Computer Science and Engineering discipline."

Department Mission

The department vision is clearly defined and is in line with the college's vision. The vision of the department is: "To evolve as a centre of academic excellence, advanced research and innovation in the field of Artificial Intelligence and Data Science discipline."

Program Educational Objectives
(Undergraduate)

We have program educational objectives for our Computer Science and Engineering program. Program educational objectives are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.

Our program educational objectives are:

I. The graduates of the Program will have knowledge and skills for data analysis, including mathematics, science and basic engineering.
II. The graduates of the Program will have in-depth learning skills to function as members of multi-disciplinary teams and to communicate effectively using modern tools.
III. The graduates of the Program will have extensive knowledge in state-ofthe- art frameworks in Artificial Intelligence and be prepared for their careers in the software industry or pursue higher studies and continue to develop their professional knowledge.
IV. The graduates of the program will practice the profession with ethics, integrity, leadership and social responsibility

Program Outcomes

On successful completion of the B.Tech (Artificial Intelligence & Data Science) programme the student will be able to:

PO1 - *Engineering knowledge:* Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 - *Problem analysis:* Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 - *Design/development of solutions:* Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 - *Conduct investigations of complex problems:* Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 - *Modern tool usage:* Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 - *The engineer and society:* Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 - *Environment and sustainability:* Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 - *Ethics:* Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 - *Individual and team work:* Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 - *Communication:* Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 - *Project management and finance:* Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 - *Lifelong learning:* Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1: Develop software applications or solutions as per the needs of Industry and society.

PSO2: Adopt new and fast emerging technologies in computer science and engineering.

SCHEME OF INSTRUCTION

**VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
DEEMED TO BE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND ALLIED BRANCHES
SCHEME OF INSTRUCTION (SU24)**

Part A (Minimum 12 Credits)

Sl. No.	Course Name	Semester	Credits	Hours
1	Research Methodology (Common for all)	I	4	60
2	Course 1: Suggested by Research Supervisor related to research work to be carried out from a pool of courses prescribed by respective department.	I	3	45
3	Oral Presentation/Seminar: Technical Paper Writing relevant to research work / domain	II	2	–
4	Course 2: Suggested by Research Supervisor related to research work to be carried out from a pool of courses prescribed by respective department.	II	3	45
		TOTAL	**12**	**150**

S.No	Course Code	Title of the Course	L	T	P	C	SE	Total
1	24UC701	Research Methodology (Common for All)	4	0	0	4	100	100

S.No	Course Code	Title of the Course	L	T	P	C	SE	Total
1	24UC702	Oral Presentation/Seminar	0	0	0	2	100	100

S.No	Course Code	Title of the Course	L	T	P	C	SE	Total
1	24UC800	Ph.D Thesis	–	–	–	–	–	–

Ph.D. COMPUTER SCIENCE & ENGINEERING AND ALLIED BRANCHES

POOL I Courses:

S.No	Course Code	Title of the Course	L	T	P	C	SE
1	24CS710A	Data Science	3	0	0	3	100
2	24CS710B	Cloud Computing	3	0	0	3	100
3	24CS710C	Soft Computing	3	0	0	3	100
4	24CS710D	Introduction to Biostatistics	3	0	0	3	100
5	24CS710E	Social Network Analysis	3	0	0	3	100
6	24CS710F	Augmented Reality and Virtual Reality	3	0	0	3	100
7	24CS710G	Software Reliability	3	0	0	3	100
8	24CS710H	Introduction to Machine Learning (NPTEL)	-	-	-	3	100
9	24CS710I	Advanced Computer Networks (NPTEL)	-	-	-	3	100
10	24CS710J	Computer Vision and Image Processing (NPTEL)	-	-	-	3	100
11	24CS710K	Introduction to Internet of Things (NPTEL)	-	-	-	3	100
12	24CS710L	Advanced Distributed Systems (NPTEL)	-	-	-	3	100
13	24CS710M	Advanced Linear Algebra (NPTEL)	-	-	-	3	100
14	24CS710N	Cryptography and Network Security (NPTEL)	-	-	-	3	100
15	24CS710O	Geographic Information Systems (NPTEL)	-	-	-	3	100
16	24CS710P	Signal Processing Techniques and its Applications (NPTEL)	-	-	-	3	100

POOL II Courses:

S.No	Course Code	Title of the Course	L	T	P	C	SE
1	24CS720A	Wireless Ad Hoc and Sensor Networks	3	0	0	3	100
2	24CS720B	Generative AI and Large Language Models	3	0	0	3	100
3	24CS720C	Edge Computing	3	0	0	3	100
4	24CS720D	Quantum Computing	3	0	0	3	100
5	24CS720E	Evolutionary Computing	3	0	0	3	100
6	24CS720F	Cyber Forensics	3	0	0	3	100
7	24CS720G	Remote Sensing and GIS	3	0	0	3	100
8	24CS720H	Digital Speech Processing	3	0	0	3	100
9	24CS720I	Knowledge Graphics	3	0	0	3	100
10	24CS720J	Extended Reality and Metaverse	3	0	0	3	100
11	24CS720K	Secure Software Engineering	3	0	0	3	100
12	24CS720L	Deep Learning (NPTEL)	-	-	-	3	100
13	24CS720M	Natural Language Processing (NPTEL)	-	-	-	3	100
14	24CS720N	Introduction to Industry 4.0 and Industrial Internet of Things (NPTEL)	-	-	-	3	100
15	24CS720O	Blockchain and Its Applications (NPTEL)	-	-	-	3	100
16	24CS720P	Bioinformatics: Algorithms and Applications (NPTEL)	-	-	-	3	100

24UC701 RESEARCH METHODOLOGY
(Common for all)

Course Category:	**Credits:** 4
Course Type: Theory	**Lecture-Tutorial-Practice:** 4 - 0 - 0
Semester End Evaluation: 100	**Total Marks:** 100

COURSE CONTENT

Unit-I: Research Methodology – An Introduction (09 hours)
Function of Research – Meaning of Research – Motivation in Research – Significance of Research – Characteristics of Research – Steps involved in Research – Research in Pure and Applied Sciences – Interdisciplinary Research.
Literature review, Surveying, Synthesizing, critical analysis, reading materials, reviewing, rethinking, critical evaluation, interpretation, Research Purposes, Ethics in research – APA Ethics code.

Unit-II: Defining the Research Problem (09 hours)
Selecting the Research problem – Necessity of defining the problem – Goals and Criteria for identifying problems for research.
Perception of Research problem – Techniques involved in defining the problem – Source of problems – Personal consideration.

Unit-III: Research Design (15 hours)
Formulation of Research design – Need for Research design – Features of a good design – Important concepts related to Research design – Basic Principles of Experimental Designs.
Methods of Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, other methods – Warranty cards, Distributor or store audits, Pantry audits, Consumer panels, Use of mechanical devices, Projective techniques.

Unit-IV: Processing and Analysis of Data (15 hours)
Processing Operations, Some Problems in Processing, Elements/Types of Analysis, Statistics in Research, Measures of Central

Tendency, Measures of Dispersion, Measures of Asymmetry (Skewness), Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression.
Sampling Fundamentals: Need for Sampling, Some Fundamental Definitions, Important Sampling Distributions, Central Limit Theorem, Sampling Theory, Sandler's A-test, Chi-square Test, Analysis of Variance and Covariance, Multivariate Analysis Techniques, Time Series Analysis.

Unit-V: Interpretation and Report Writing **(12 hours)**

Meaning and Technique of interpretation – Precautions in interpretation – Significance of report writing – Different steps in writing a report – Layout of a Research report.

Types of report – Mechanics of writing a research report – Precautions for writing a research report – Conclusion.

TEXT BOOKS

1. Research Methodology Methods & Techniques, C.R. Kothari – New Age international Publishers, Reprint 2008.
2. A Hand Book of Methodology of Research, Rajammall, P. Devadoss and K. Kulandairvel, RMM Vidyalaya press, 1976.
3. Montgomery, D.C., 2017. Design and analysis of experiments. John Wiley & Sons.

REFERENCE BOOKS

1. Donald H. McBurney, Research Methods, 5th edition, Thomson Learning, ISBN: 81-315-0047-0, 2006.
2. Thesis and Assignment Writing, J. Anderson, Wiley Eastern Ltd., 1997.
3. Research Methodology, Mukul Gupta, Deepa Gupta – PHI Learning Private Ltd., New Delhi, 2011.
4. Fundamentals of Mathematical Statistics, S.C. Gupta and V.K. Kapoor, Sultan Chand & Sons, New Delhi, 1999.

POOL I — COURSES SYLLABUS

24CS710A DATA SCIENCE

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Data Structures	**Semester End Evaluation:** 100
Total Marks: 100	

COURSE CONTENT

UNIT I

Introduction: Relation among AI, ML, and Data Science, Importance of Data Science.

What is Data Science: Extracting Meaningful Patterns, Building Representative Models, Combination of Statistics, Machine Learning and Computing, Learning Algorithms, Associated Fields.

Data Objects and Attribute Types: What Is an Attribute?, Nominal Attributes, Binary Attributes, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes.

Basic Statistical Descriptions of Data: Measuring the Central Tendency: Mean, Median, and Mode. Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range.

UNIT II

Graphic Displays of Basic Statistical Descriptions of Data: Quantile Plot, Quantile–Quantile Plot.

Visualization: Univariate Visualization: Histogram, Quartile, Distribution Chart. Multivariate Visualization: Scatterplot, Scatter Multiple, Bubble Chart, Density Chart.

Visualizing High-Dimensional Data: Parallel Chart, Deviation Chart, Andrews Curves.

UNIT III

Data Similarity/Dissimilarity: Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Minkowski Distance, Proximity Measures for Ordinal Attributes, Dissimilarity for Attributes of Mixed Types, Cosine Similarity.

UNIT IV

Data Science Process: Prior Knowledge, Data Preparation, Modeling, Evaluation, Deployment.

Data Preprocessing: Data Quality: Why Preprocess the Data? Major Tasks in Data Preprocessing. **Data Cleaning:** Missing Values, Noisy Data, Data Cleaning as a Process.

Data Integration: Entity Identification Problem, Redundancy and Correlation Analysis, Tuple Duplication, Data Value Conflict Detection and Resolution.

Data Reduction: Overview of Data Reduction Strategies, Wavelet Transformation, Principal Components Analysis, Attribute Subset Selection. **Data Transformation and Data Discretization:** Data Transformation Strategies Overview, Data Transformation by Normalization, Discretization by Binning, Concept Hierarchy Generation for Nominal Data.

UNIT V

Regression: Introduction to Linear Regression and Logistic Regression for Classification Problems. Regression Metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2) Score, Root Mean Squared Error (RMSE).

Classification: Classification Principles, Classification Model Evaluation Metrics (Confusion Matrix), Precision, Recall, AUC-ROC.

Clustering: Clustering Principles, Clustering Evaluation Parameters, Clustering Algorithms (k-Means) and Evaluation Metrics for Assessing the Quality of Clustering Results.

Model Evaluation: Generalization Error – Out-of-Sample, Evaluation Metrics – Cross Validation – Overfitting – Underfitting.

TEXT BOOKS

1. Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, 2012. Third Edition (UNIT 1, UNIT 2, UNIT 4).

2. Vijay Kotu and Bala Deshpande, *Data Science Concepts and Practice*, 2019. Second Edition (UNIT 1, UNIT 2, UNIT 3, UNIT 4).

REFERENCE BOOKS

1. Joel Grus, *Data Science from Scratch*, 2019, 2nd Edition.

2. Tom M. Mitchell, *Machine Learning*, Tata McGraw Hill, 1997.

3. Sunila Gollapudi, *Practical Machine Learning*, First Edition, 2016, Packt Publishing Ltd.

E-RESOURCES AND OTHER DIGITAL MATERIAL

1. Google's Machine Learning Crash Course, last accessed on 11-05-2024, `https://developers.google.com/machine-learning/crash-course`

2. IBM's Machine Learning with Python, last accessed on 11-05-2024, `https://cognitiveclass.ai/courses/machine-learning-with-python`

24CS710B
CLOUD COMPUTING

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description
This course provides an introduction to cloud computing concepts, models, and architectures. It covers the fundamental principles of cloud computing, including service models (SaaS, PaaS, IaaS), cloud infrastructure, and emerging technologies in cloud services. Students will gain practical knowledge of leading cloud platforms, such as AWS and Google Cloud, and learn about security, privacy, and cost management in the cloud environment.

Course Objectives
The main objectives of this course are to:

- Understand the architecture and delivery models of cloud computing.
- Introduce leading cloud platforms like Google Cloud and AWS.
- Familiarize students with key cloud computing services (SaaS, PaaS, IaaS).
- Discuss security, confidentiality, and privacy issues in the cloud.
- Enable students to develop and manage applications on private and hybrid cloud infrastructures.

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

- CO1: Describe cloud computing architectures and delivery models.
- CO2: Analyze cloud infrastructure, focusing on leading providers like Google Cloud and AWS.
- CO3: Develop private and hybrid clouds for executing customized applications.
- CO4: Assess authentication, confidentiality, and privacy challenges in the cloud.
- CO5: Evaluate the financial and technological implications of cloud platform choices.

Unit – I: Cloud Computing Fundamentals (7 Hrs.)

Content:
Introduction to Cloud Computing: Architecture, characteristics, and service models (SPI Framework - SaaS, PaaS, IaaS). Cloud Delivery Models and Cloud Ecosystem. The Evolution of Cloud Computing vs. Traditional IT Model.
Cloud software architectures and defining Service-Oriented Architecture (SOA) in the cloud context.

Description:
This unit introduces the foundational concepts of cloud computing, including architecture, characteristics, and service models. Students learn about the SPI Framework (SaaS, PaaS, IaaS) and compare cloud computing with traditional IT infrastructure. The unit also introduces the service-oriented architecture (SOA) within the cloud context, equipping students with the knowledge to identify and describe cloud delivery models.

Examples / Applications / Case Studies:

1. Compare traditional IT infrastructure with cloud computing models.
2. Case study: Implementing SaaS in an educational setup.
3. Overview of Google Cloud and Amazon Cloud structures.

Exercises:

1. Design a diagram illustrating cloud computing layers and service models.
2. Analyze a real-world application to identify the advantages of cloud deployment.

Learning Outcome:
Describe cloud computing architecture, service models, and basic concepts.

Web Resources:

1. Introduction to Cloud Computing Architecture - Available on Google Cloud Platform (last accessed: 12-06-2024).
2. Cloud computing basics, Microsoft Azure (last accessed: 12-06-2024).

Unit – II: Cloud Infrastructure and Virtualization (7 Hrs.)

Examples / Applications / Case Studies:

1. Explore MapReduce in Google Cloud with a focus on search engine applications.
2. Case study: Implementing data storage on AWS S3.
3. Virtualization using VMware, Docker, and Kubernetes in cloud environments.

Exercises:

1. Create a virtual machine on AWS EC2 and deploy a basic web application.
2. Compare cloud storage options on Google Cloud and AWS.

Learning Outcome:
Analyze cloud infrastructure components, focusing on Google and AWS.

Web Resources:

1. Google Cloud Documentation (last accessed: 12-06-2024).
2. Amazon Web Services Documentation (last accessed: 12-06-2024).

Unit – III: Cloud Application Development and API Management (7 Hrs.)

Content:
REST APIs and SOAP APIs for cloud services.
Developing cloud-native applications with Platform as a Service (PaaS).
Combining SOA with cloud architectures.
Amazon Cloud Services: SimpleDB, CloudFront, Elastic MapReduce.

Description:
This unit explores cloud application development using REST and SOAP APIs and discusses cloud-native application design in a PaaS environment. Students learn to integrate SOA with cloud architectures to develop scalable cloud applications, with a focus on AWS and Google Cloud tools like SimpleDB, CloudFront, and Elastic MapReduce. **Examples / Applications / Case Studies:**

1. Building a RESTful API for a cloud-based e-commerce platform.

2. Case study: Creating a scalable cloud application using AWS Lambda.
3. Practical: Developing a microservice architecture using AWS and SOA.

Exercises:

1. Implement a cloud-native application on AWS using S3 and Lambda.
2. Develop a service-oriented application using Google App Engine.

Learning Outcome:
Develop and manage cloud-native applications using SOA and cloud APIs.

Web Resources:

1. API Management on AWS (last accessed: 12-06-2024).
2. Google App Engine - Building Applications (last accessed: 12-06-2024).

Unit – IV: Cloud Security and Privacy (7 Hrs.)

Security in Cloud Computing: Authentication, confidentiality, data integrity

- Privacy concerns and regulatory frameworks.
- Common cloud security standards and compliance requirements.
- Best practices for data protection and user privacy in the cloud.

Python Data Structures:

Tuple, List, Set, and Dictionary.

Description:

This unit examines the essential security aspects of cloud computing, focusing on authentication, confidentiality, and data integrity. It covers regulatory frameworks, common security standards, and best practices to protect data and privacy in cloud environments.

Examples / Applications / Case Studies:

1. Explore multi-factor authentication for cloud applications.
2. Case study: Managing data privacy in a healthcare cloud application.
3. Security protocols for cloud-based financial services.

Exercises:

1. Design a secure cloud storage solution for confidential data.
2. Evaluate a cloud service provider's security compliance for sensitive applications.

Learning Outcome:

- Assess security and privacy challenges in cloud computing.

Web Resources:

1. AWS Cloud Security Best Practices (last accessed: 12-06-2024).
2. Google Cloud Security and Compliance (last accessed: 12-06-2024).

Unit – V: Cloud Financial and Technological Implications (10 Hrs.)

- Cost Management in Cloud Computing: Cost models, budgeting, and pricing (AWS EC2, S3).
- Financial impacts of cloud adoption and cost-optimization techniques.
- Comparison of cloud platforms: AWS, Google Cloud, and Microsoft Azure.
- Technological implications of cloud adoption, including scalability, reliability, and high availability.

Description:

This unit focuses on managing costs in cloud computing, analyzing pricing models, and understanding the financial and technological implications of adopting cloud platforms. Students learn cost-optimization techniques and compare cloud platforms such as AWS, Google Cloud, and Azure.

Examples / Applications / Case Studies:

1. **Case study:** Cost-benefit analysis of migrating an enterprise application to AWS.
2. Optimizing cloud resources to reduce costs without compromising performance.
3. Evaluate Google Cloud, AWS, and Azure for different use cases and performance.

Exercises:

Inheritance:

1. Perform a cost analysis on AWS using EC2 pricing calculators.
2. Create a resource optimization plan for a cloud-based application.

Learning Outcome:

- Evaluate financial and technological considerations for cloud platform selection.

Web Resources:

1. AWS Pricing Calculator (last accessed: 12-06-2024).
2. Google Cloud Pricing (last accessed: 12-06-2024).

Textbooks:

1. Velte, A. T., Velte, T. J., & Elsenpeter, R. (2010). *Cloud Computing: A Practical Approach*. MGH.
2. Bahga, A., & Madisetti, V. (2014). *Cloud Computing: A Hands-on Approach*. Universities Press (India) Private Limited.
3. Shroff, G. (2010). *Enterprise Cloud Computing*. Cambridge.
4. Buyya, R., Broberg, J., & Goscinski, A. (2011). *Cloud Computing Principles and Paradigms*. Wiley.
5. Krutz, R. L., & Vines, R. D. (2010). *Cloud Security*. Wiley.

Reference Books:

1. Bernard Golden, *Amazon Web Services for Dummies*, John Wiley & Sons, 2013.
2. Joe Weinman, *Cloudonomics: The Business Value of Cloud Computing*, Wiley, 2012.
3. J. Hurwitz, R. Bloor, M. Kaufman, & F. Halper, *Cloud Computing for Dummies*, Wiley, 2010.

24CS710C
SOFT COMPUTING

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description:

This course focuses on giving an introduction to some new fields in soft computing with its principal components of fuzzy logic and neural networks, which help students differentiate traditional and neural networks. This course provides insightful study about problems incurred in various domains, and the comprehensive soft computing techniques provide solutions to these problems, benefiting students for the pursuit of allied research.

Course Aims and Objectives:

The aim of this course is to provide a comprehensive understanding of soft computing concepts, techniques, and applications.

Objectives:

1. To provide knowledge on principles of soft computing and its differences between soft computing and hard computing.
2. Learn the fundamentals of fuzzy logic and neural computing.
3. To impart skills in implementing soft computing solutions using pattern recognition tasks and back-propagation learning algorithms.
4. Apply soft computing techniques to solve real-world problems.
5. To enable students to design and manage hybrid systems integrating multiple soft computing paradigms.

Course Outcomes:

At the end of the course, the student will be able to:

1. CO1: To apply the soft computing techniques for solving the problems [K3].
2. CO2: To apply crisp sets and fuzzy sets in computer science and engineering [K3].

3. CO3: To apply crisp relations and fuzzy relations in computer science and engineering [K3].

4. CO4: To understand the artificial neural network and its applications [K2].

5. CO5: To understand back-propagation algorithm [K2].

Course Structure

Unit 1:

Introduction to Soft Computing

- Concept of computing systems
- "Soft" computing versus "Hard" computing
- Characteristics of Soft Computing
- Some applications of Soft Computing techniques
- Fuzzy logic:
  - Introduction to Fuzzy logic: Fuzzy set theory
  - Fuzzy versus crisp
  - Operations on crisp sets
  - Properties of crisp sets

Description:
This unit provides a foundational understanding of Soft Computing and Fuzzy Logic.

Soft Computing is an approach that tolerates uncertainty and imprecision. Fuzzy Logic is a method for handling vagueness and uncertainty, distinguishing between crisp (binary) and fuzzy (continuous) sets. The basics of fuzzy set theory, operations on crisp sets, and their properties are also covered.

Examples/Applications/Case Studies:

1. Google's image recognition algorithm uses soft computing techniques to identify objects in images.
2. Finance: Credit scoring and risk assessment using soft computing.

Unit II:

- Fuzzy versus Crisp
- Crisp sets
- Fuzzy sets
- Crisp relations
- Fuzzy relation

Description:
This unit focuses on fuzzy sets and membership functions to represent uncertain information. Fuzzy relations, rules, and inference engines enable decision-making, while defuzzification techniques convert fuzzy sets into crisp values.

Examples/Applications/Case Studies:

1. Image segmentation, using fuzzy relations to represent relationships between image pixels.
2. Case Study: A predicate logic-based system was developed for natural language processing, using predicate logic to represent and analyze the meaning of sentences.

Learning Outcomes:

- Understand the concept of fuzzy sets and their representation using membership functions.
- Analyze the properties of fuzzy set operations and their implications.
- Analyze Fuzzy relations like Fuzzy Cartesian product, Operations on Fuzzy relations.

Specific Resources:

- Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

Unit III:

- Crisp Logic
- Predicate Logic
- Fuzzy Logic
- Fuzzy Rule-based System
- Defuzzification

Description:
This unit focuses on Fuzzy Logic and Fuzzy Rule-based Systems. Fuzzy handles uncertainty. Defuzzification converts fuzzy outputs into crisp, actionable results. These logic systems have diverse applications in Artificial Intelligence, Control Systems, and Image Processing, enabling effective decision-making and problem-solving in complex environments.

Examples/Applications/Case Studies:

- Maintain optimal temperature (20-25°C) in a greenhouse, considering factors like outside temperature, humidity, and time of day. Using Fuzzy logic, Crisp Logic, and Predicate Logic ensures efficient decision-making.

Learning Outcomes:

1. Apply Predicate Logic in problem-solving.
2. Develop intelligent systems using Fuzzy Logic.

Specific Resources:

- Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

Unit IV:

- Basic Concepts of Neural Networks
- Human Brain
- Model of Artificial Neuron
- Neural Network Architecture
- Characteristics of Neural Networks
- Learning Methods

Description:
This unit focuses on Basic Concepts of Neural Networks, Model of Artificial Neuron, Neural Network Architecture. These fundamental concepts form the foundation of neural networks, enabling machines to learn, adapt, and make predictions or decisions.
Examples/Applications/Case Studies:

- Handwritten Digit Recognition Using Neural Networks on MNIST dataset, consisting of handwritten digit images.

Learning Outcomes:

1. Understand the biological inspiration behind neural networks and their connection to the human brain.
2. Learn the structure and functionality of artificial neurons and neural network architectures.

Specific Resources:

- Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

Unit V:

- Architecture of Back Propagation
- Back Propagation Learning
- Input Layer Computation
- Output Layer Computation
- Hidden Layer Computation
- Calculation of Error
- Training of Neural Networks
- Back Propagation Algorithm

Description:
This unit focuses on feed-forward neural networks with input, hidden, and output layers connected through weights, where backpropagation adjusts these weights during learning. A systematic process of forward prediction, error calculation, and backward adjustment of weights using the chain rule is used to optimize performance.

Examples/Applications/Case Studies:
Case Study: Predicting House Prices Using Neural Networks

- **Description:** A feed-forward neural network with multiple hidden layers is used to predict house prices based on features like size, location, and amenities.

Learning Outcomes:

1. Understand the structure and workflow of the backpropagation algorithm, including input, hidden, and output layer computations, and how they contribute to predictions.
2. Learn how errors are calculated and propagated backward to optimize weights, improving the network's learning and performance.

Specific Resources:

- Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

Text Books:

1. Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

Reference Books:

1. Daniel Jurafsky and James Martin, *Speech and Language Processing*, Prentice-Hall, Second Edition, 2008, ISBN: 0131873210.

2. Goldberg, D. E., *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading Mass.
3. Ross, T. J., *Fuzzy Logic with Engineering Applications* (3rd ed.), John Wiley and Sons.

E-Resources and Other Digital Material

1. Samanta, D. (2018, April 3). *Introduction To Soft Computing*. NPTEL Online Courses. https://onlinecourses.nptel.ac.in/noc22_cs54/preview. Last accessed on June 5, 2022.

2. Wall, M. (1996, August). *GALib: A C++ Library of Genetic Algorithm Components*. Massachusetts Institute of Technology. http://lancet.mit.edu/ga/dist/galibdoc.pdf. Last accessed on May 19, 2022.

3. http://cse.iitkgp.ac.in/~dsamanta/courses/sca/index.html

4. http://lancet.mit.edu/ga/dist/galibdoc.pdf

24CS710D
INTRODUCTION TO BIOSTATISTICS

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Engineering Mathematics	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course provides fundamental knowledge and skills that are essential to understand and implement numerical and statistical methods in biomedical and public health research. This course covers foundational statistical concepts, design study, collection of data, techniques for analysis of data, and elucidation of results. Profound implementation will be on practical application through hands-on exercises and real-time examples.

Course Objectives

The main objectives of this course are to:

- Apply basic statistical concepts commonly used in research.
- Use basic analytical techniques to generate results.
- Demonstrate statistical reasoning skills correctly and contextually.
- Communicate statistical analysis through written scientific reporting for research in various fields.

Course Outcomes

At the end of the course, the student will be able to:

CO1: Understand fundamental concepts of statistics, their relevance in research (K2).

CO2: Conduct different types of study, design and their strengths and limitations (K2).

CO3: Acquire skills in data collection, management, and quality assurance (K2).

CO4: Gain proficiency in understanding data various implementation techniques to extract data from various levels (K3).

CO5: Elucidate statistical results and communicate findings effectively (K3).

Course Content

Unit – I: Probability Distributions (8 Hrs.)
Probability and random variables - Probability concepts, Bayes's theorem, random variables. Discrete and continuous function of random variables. Moment generating function, Poisson Geometric, Uniform exponential distributions, normal distribution.

Unit – II: Random Variables (7 Hrs.)
Two-dimensional random variables, marginal and conditional distributions, expectational and conditional expectations. Transformation of two-dimensional random variables, central limit theorem (statement and problems).

Unit – III: Sampling Distributions (8 Hrs.)
Sampling concepts - methods of sampling. Simple random sampling, stratified random sampling, systematic sampling point and interval estimators, properties of estimators, sample size determination, testing of hypothesis, small samples (t-test), large samples (Z test, confidence limits).

Unit – IV: Some Specialty Probability Distributions (7 Hrs.)
Random processes - classification stationary and Markov process, Poisson processes, Markovian chains, Markovian Quenching models, infinite and finite with single server.

Unit – V: Non-Parametric and Distribution Free Statistics (7 Hrs.)
Design of experiment and non-parametric test - completely randomized design, Randomised block design, Latin square design sign test, Kolmogorov Smirnov Test, Mann-Whitney U test, Kruskal-Wallis test.

Text & Reference Books:

1. W. W. Daniel, *BIOSTATISTICS - A Foundation for Analysis in the Health Sciences*, 2013 by John Wiley and Sons, New Delhi.
2. CHAP T. LE, *INTRODUCTORY BIOSTATISTICS*, 2003 by John Wiley & Sons.
3. Veerarajan T, *Probability Statistics and Random Process*, 2008 by Tata Mc-GrawHill, New Delhi.
4. Indranil Saha, Bobby Paul, *Essentials of Biostatistics & Research Methodology*, 2021 by Academic Publishers.

Web Resources:

1. Dr Shamik Sen., (2017), “Introduction to Biostatistics”, IIT Mumbai, Last accessed on 17-11-2024
https://www.youtube.com/watch?v=Vz5jztR6QFM&list=PL0Noor1DIEikWkIRSwtu2g-zAS_NdHeV0

2. Mukesh Doble., (2016), “Biostatistics and design of experiments”, IIT Madras, Last accessed on 17-11-2024
https://www.youtube.com/watch?v=B3pAD8ie3t0&list=PLoNoar1DIEikPbM5cdpXOxDtQcrb4fQ5

3. Benish, Ali., (2020), “Introduction to Biostatistics”, Last accessed on 15-11-2024
https://www.youtube.com/watch?v=nBHw5izbDY&list=PLi7GKP9W2qzfd9kHpT3WrW1WF9h0yHFoK

24CS710E

SOCIAL NETWORK ANALYSIS

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Engineering Mathematics	**Semester End Evaluation:** 100
Total Marks: 100	

Course Content

Unit I

Introduction to Social Network Analysis: Definitions and history of SNA. Networks in social sciences vs. natural sciences. Overview of graph theory basics.
Network Data and Representations: Types of network data (ego networks, full networks, two-mode networks). Adjacency matrices, edge lists, and sociograms. Data collection techniques (surveys, APIs, observational methods).

Unit II

Network Visualization: Principles of effective network visualization. Tools for network visualization (Gephi, Cytoscape, R/igraph).
Measures of Network Structure: Node-level measures: degree, centrality (closeness, betweenness, eigenvector). Group-level measures: clustering, density, assortativity. Network-level measures: diameter, modularity.

Unit III

Theoretical Foundations: Structural holes and social capital (Burt). Weak and strong ties (Granovetter). Homophily and assortative mixing.
Network Models: Random networks (Erdös-Rënyi model), Small-world networks (Watts-Strogatz model), Scale-free networks (Barabási-Albert model).

Unit IV

Community Detection: Clustering and modularity. Algorithms for community detection (Girvan-Newman, Louvain). Applications of community detection.
Diffusion and Influence in Networks: Models of diffusion (SI, SIR, SIS), Cascades and thresholds. Influence maximization problems.

Unit V

Dynamic and Temporal Networks: Time-varying networks. Longitudinal data collection and analysis.
Applications of SNA: Networks in public health (e.g., epidemic spread). Networks in economics (trade, finance). Political networks (e.g., voting blocs, lobbying).

Text Books

1. Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications.*

2. Barabási, A.-L. (2016). *Network Science.*

3. Newman, M. (2010). *Networks: An Introduction.*

Reference Books

1. *Social Network Analysis: Methods and Applications* by Stanley Wasserman and Katherine Faust.

2. *Analyzing Social Networks* by Stephen P. Borgatti, Martin G. Everett, and Jeffrey C. Johnson.

3. *Exploratory Social Network Analysis with Pajek* by Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj.

4. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World* by David Easley and Jon Kleinberg.

E-Resources and Other Digital Material

1. Gephi: Open-source visualization and exploration tool for SNA. *[Gephi Official Website]*

2. UCINET: Widely used for advanced SNA metrics and visualizations. *[UCINET Official Website]*

3. NodeXL: A Microsoft Excel plugin for network analysis and visualization. *[NodeXL Website]*

4. NetworkX: Python library for the creation, manipulation, and study of complex networks. *[NetworkX Documentation]*

24CS710F

AUGMENTED REALITY AND VIRTUAL REALITY

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Engineering Mathematics	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course provides a foundational understanding of Augmented Reality (AR) and Virtual Reality (VR), exploring their concepts, technologies, and real-world applications. Students will learn the fundamental principles behind AR and VR, including the hardware, software, and design techniques used to create immersive experiences.

Course Objectives

The main objectives of this course are to:

- Learn the fundamental Computer Vision, Computer Graphics and Human-Computer Interaction Techniques related to VR/AR.
- Introduce Geometric Modeling Techniques.
- Understand the Virtual Environment.
- Explain and Examine VR/AR Technologies.
- Use various types of Hardware and Software in Virtual Reality systems.

Course Outcomes

At the end of the course, the student will be able to:

- CO1: Understand fundamental Computer Vision, Computer Graphics and Human-Computer Interaction Techniques related to VR/AR (K2).
- CO2: Explain Geometric Modeling Techniques (K2).
- CO3: Understand the Virtual Environment (K2).
- CO4: Apply VR/AR Technologies in real-world scenarios (K3).
- CO5: Apply various types of Hardware and Software in Virtual Reality systems (K3).

Course Content

Unit – I: Augmented Reality (7 Hrs.)

Augmented reality: Taxonomy, Technology and Features of Augmented Reality, AR Vs VR, Challenges with AR, AR systems and functionality, Augmented Reality Methods.Visualization Techniques for Augmented Reality Enhancing interactivity in AR Environments, Evaluating AR systems

Unit – II: Computer Graphics and Geometric Modelling (7 Hrs.)

The Virtual world space, positioning the virtual observer, the perspective projection, human vision, stereo perspective projection, Color theory, Conversion from 2D to 3D, 3D space curves, 3D boundary representation, Simple 3D modelling, 3D clipping, Illumination models, Reflection models, Shading algorithms, Geometrical Transformations: Introduction, Frames of reference, Modelling transformations, Instances, Picking, Flying, Scaling the VE, Collision detection.

Unit – III: Virtual Environment (8 Hrs.)

Input/Output Devices: Input (Tracker, Sensor, Digital Gloves, Movement Capture, Video-based Input, 3D Menus
3D Scanner, etc.), Output (Visual/Auditory/Haptic Devices)
Generic VR system: Introduction, Virtual environment, Computer environment, VR technology, Model of interaction, VR Systems, Animating the Virtual Environment: Introduction, The dynamics of numbers, Linear and Nonlinear interpolation, the animation of objects, linear and non-linear translation, shape
object in between, free from deformation, particle system Physical Simulation: Introduction, Objects falling in a gravitational field, Rotating wheels, Elastic collisions, projectiles, simple pendulum, springs, Flight dynamics of an aircraft.

Unit – IV: Introduction to Virtual Reality (VR) (7 Hrs.)

Virtual Reality and Virtual Environment: Computer graphics, Real time computer graphics, Flight Simulation, Virtual environment requirement, benefits of virtual reality, Historical development of VR, Scientific landmark.

Unit – V: Development Tools and Frameworks (7 Hrs.)

Human factors: Introduction, the eye, the ear, the somatic senses.
Hardware: Introduction, sensor hardware, Head-coupled displays, Acoustic hardware, Integrated VR systems.
Software: Introduction, Modelling virtual world, Physical simulation, VR toolkits, Introduction to VRML.

Textbooks

1. Coiffet, P., Burdea, G. C., (2003), *Virtual Reality Technology*, Wiley-IEEE Press, ISBN: 9780471360896

2. Schmalstieg, D., Höllerer, T., (2016), *Augmented Reality: Principles Practice*, Pearson, ISBN: 9789332557494

3. Norman, K., Kirakowski, J., (2018), *Wiley Handbook of Human Computer Interaction*, Wiley-Blackwell, ISBN: 9781118976135

4. LaViola Jr., J. J., Kruijff, E., McMahan, R. P., Bowman, D. A., Poupyrev, I., (2017), *3D User Interfaces: Theory and Practice*, Pearson, ISBN: 9780134034324

5. Fowler, A., (2019), *Beginning IOS AR Game Development: Developing Augmented Reality Apps with Unity and C#*, Apress, ISBN: 9781484246672

6. Hassaniien, A. E., Gupta, D., Khamra, A., Slowik, A., (2022), *Virtual and Augmented Reality for Automobile Industry: Innovation Vision and Applications*, Springer, ISBN: 9783030941017

Reference Books

1. Craig, A. B., (2013), *Understanding Augmented Reality, Concepts and Applications*, Morgan Kaufmann, ISBN: 9780240824086

2. Craig, A. B., Sherman, W. R., Will, J. D., (2009), *Developing Virtual Reality Applications, Foundations of Effective Design*, Morgan Kaufmann, ISBN: 9780123749437

3. John Vince, J., (2002), *Virtual Reality Systems*, Pearson, ISBN: 9788131708446

4. Anand, R., *Augmented and Virtual Reality*, Khanna Publishing House

5. Kim, G. J., (2005), *Designing Virtual Systems: The Structured Approach*, ISBN: 9781852339596

6. Bimber, O., Raskar, R., (2005), *Spatial Augmented Reality: Merging Real and Virtual Worlds*, CRC Press, ISBN: 9781568812304

7. OĆonnell, K., (2019), *Designing for Mixed Reality: Blending Data, AR, and the Physical World*, OŔeilly, ISBN: 9789352138731

8. Sami Siltanen, S., (2021), *Theory and applications of marker-based augmented reality*, Julkaisija – Utgivare Publisher, ISBN: 9789513874407

Web Resources

- Manivannan, M., (2018),*Virtual Reality Engineering*, IIT Madras, Last accessed on 15-11-2024 https://nptel.ac.in/courses/121106013

- Misra, S., (2019),*Industry 4.0: Augmented Reality and Virtual Reality*, IIT Kharagpur, Last accessed on 15-11-2024 https://www.youtube.com/watch?v=zLMgdYI82IE

- Dube, A., (2020), *Augmented Reality - Fundamentals and Development*, NPTEL Special Lecture Series, Last accessed on 15-11-2024 https://www.youtube.com/watch?v=MGuSTAqlZ9Q

24CS710G
SOFTWARE RELIABILITY

Course Category: Pool 1 - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:software Engineering	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course explores the principles, practices, and methodologies used to design, develop, and maintain reliable software systems. Software reliability focuses on ensuring that software performs its intended functions under specified conditions without failure over a defined period. The course covers key topics such as reliability modeling, Software Metrics for Reliability assessment, software testing techniques, and Neural Networks for Software Reliability.

Course Objectives

The main objectives of this course is to:

- To familiarize students with the core concepts of Software Reliability Engineering.
- To explore software reliability strategies and their application in real-world scenarios.
- To introduce software reliability models and predict software reliability in early lifecycle phases.
- To understand the role of software metrics in assessing reliability and gain insights into reliability estimation through software testing.
- To learn the role of neural networks in software reliability growth modeling.

Course Outcomes

At the end of the course, the student will be able to:

- CO1: Describe software reliability concepts, Strategies and operational profiles for reliability assessments.
- CO2: Apply various reliability models to predict software failures at early stages of software life cycle.

- CO3: Utilize static and dynamic complexity metrics to assess software reliability and estimate the reliability through operational profile-based testing.
- CO4: Integrate neural networks for advanced software reliability growth modeling.
- CO5: Apply SRE principles during requirements, implementation, and maintenance phases.

Course Content

Unit – I: Software Reliability Fundamentals (7 Hrs.)

Introduction: The Need for Reliable Software, Software Reliability Engineering Concepts, Basic definitions, Software practitioner's biggest problem, software reliability engineering approach, software reliability engineering process, defining the product.
The Operational Profile: Reliability concepts, software reliability and hardware reliability, developing operational profiles, applying operational profiles, learning operations and run concepts.

Description: This unit introduces the foundational concepts of Software Reliability, including software reliability engineering approach. Students learn the differences between software reliability and hardware reliability. Students will also be equipped to create and use operational profiles effectively in the reliability engineering process.

Examples / Applications / Case Studies:

1. Example: **E-commerce Platforms:** SRE principles are applied to ensure websites like Amazon can handle peak traffic (e.g., Black Friday sales) without system crashes.
2. Example: **Unclear Requirements:** Developers of an e-learning platform struggle to implement a feature because the specification does not define acceptable video buffering times.
3. Application: **Streaming Services:** Netflix uses reliability engineering to manage uptime and ensure smooth streaming experiences for millions of users simultaneously.
4. Case Study: **Toyota Prius Software Glitch:** A fault in the embedded software caused braking failures, highlighting the importance of understanding and addressing errors, faults, and failures.
5. Case Study: **Healthcare.gov Rollout:** The website faced significant issues due to unclear requirements and poor coordination, resulting in an unreliable system upon launch.

Exercises:

1. Write a short scenario describing a software issue and classify it as a failure, fault, or error, providing justification.

2. Discuss how unclear requirements can lead to reliability issues in software development. Provide an example.
3. Create a flowchart of the SRE process for a library management system services, model and basic concepts.
4. Differentiate between software reliability and hardware reliability with examples.

Learning Outcome:

- Understand the core concepts of Software Reliability Engineering principles and operational profiles.

Web Resources:

- Software Reliability concepts: https://www.javatpoint.com/software-engineering-software-reliability
- (https://www.geeksforgeeks.org/software-engineering-hardware-reliability-vs-software-reliability/)

Unit – II: Software Reliability Fundamentals (7 Hrs.)

Defining failure for the product, common measure for all associated systems, setting system failure intensity objectives, determining develop software failure intensity objectives, software reliability strategies, failures, faults and errors, availability, system and component reliabilities and failure intensities, predicting basic failure intensity. **Description:** This unit focuses on techniques like operational profiling, failure intensity prediction, and fault tolerance strategies with which the software meets user expectations and operates with minimal downtime. It also covers the strategies to enhance reliability. **Examples / Applications / Case Studies:**

1. Measuring failure intensity as the number of failures per operational hour for both software and hardware systems.
2. An e-commerce system may set a failure intensity objective of one failure per 1,000,000 transactions.
3. In an online payment system, a failure is defined as the system's inability to process transactions within a specified time limit (e.g., 5 seconds).
4. *Application:* In healthcare software, failure could be the incorrect dosage calculation in a drug administration system.
5. *Case Study: Therac-25 Incident:* A radiation therapy machine where software failures led to overdoses. The lack of clear definitions for failure conditions contributed to this catastrophic outcome.
6. *Application:* Bug-tracking systems (e.g., Jira) are used to identify and classify these issues during development.

7. *Case Study: Ariane 5 Failure:* A rocket failure caused by a software fault (integer overflow) resulted in a catastrophic explosion.

Exercises:

1. Identify the faults in the following system scenarios and propose strategies to mitigate them:
   - A server crashes under heavy traffic.
   - A mobile app displays incorrect data during offline mode.
2. Develop a fault-tolerance strategy for a critical banking application to handle unexpected downtimes.
3. For a critical healthcare monitoring system, set realistic failure intensity objectives. Justify your choices.

Learning Outcome: Develop software failure intensity objectives and measures for both software and hardware systems.

Web Resources:

1. https://wohlin.eu/softrel01.pdf
2. https://www.geeksforgeeks.org/reliability-attributes-in-software-development

Unit – III: Software Reliability Modeling Survey (7 Hrs)

Introduction,Historical Perspective and Implementation, Exponential Failure Time Class of Models, Weibull and Gamma Failure Time Class of Models, Infinite Failure Category Models, Bayesian Models, Model Relationship, Software Reliability Prediction in Early Phases of the Life Cycle.

Description: This unit introduces the need for software reliability modeling to predict and assess the dependability of software systems. These models use mathematical and statistical techniques to analyze failure behaviors and forecast reliability metrics, enabling informed decision-making during development. Each model type has specific applications and assumptions, enabling software engineers to address diverse reliability challenges effectively. Selecting an appropriate model involves considering system characteristics, failure patterns, and development phase requirements.

Examples / Applications / Case Studies:

1. Consider an online banking system where reliability modeling ensures the application can handle peak loads without failing. Predictions are made about system behavior under high transaction rates.
2. *Case Study: AT&T Network Systems:* Use exponential models to analyze failure rates in telecommunication switches.
3. *Case Study: AI in Healthcare:* Bayesian reliability modeling is applied in diagnostic software to adapt predictions as more clinical data becomes available.

4. *Practical:* Bayesian reliability modeling is applied in diagnostic software to adapt predictions as more clinical data becomes available.

Exercises:

1. Implement a cloud-native application on AWS using S3 and Lambda.
2. Develop a service-oriented application using Google App Engine.

Learning Outcome: Apply reliability models to predict and prevent service interruptions in cloud computing environments.

Web Resources:

1. https://handbook.reliability.space/
2. https://www.mdpi.com

Unit – IV: Software Metrics (7 Hrs)

Software Metrics for Reliability Assessment: Introduction, Static Program Complexity, Dynamic Program Complexity, Software Complexity and Software Quality, Software Reliability Modeling. Software Testing and Reliability: Introduction, Overview of Software Testing, Operational profiles, Time/Structure Based Software Reliability Estimation.

Description: This unit covers various software metrics for assessing reliability. Also Static and Dynamic program complexity, software complexity and software reliability modeling is covered. Students learn how to implement time and structure-based software reliability estimation.

Examples / Applications / Case Studies:

1. Measuring cyclomatic complexity to estimate the number of independent paths through the program's source code.
2. Implementing Musa's Execution Time Model to predict reliability during testing.
3. *Case Study:* AT&T applied software reliability models to enhance network switch dependability, reducing downtime significantly.
4. Using control flow graphs, identify critical failure points in the software's architecture.

Exercises:

1. Create an operational profile for a flight booking system to simulate real-world usage patterns.
2. Identify bottlenecks, improving overall game performance and reliability using structural analysis.

Learning Outcome: Assess software reliability by applying software metrics and testing strategies for healthcare domain.

Web Resources:

1. https://www.ijcaonline.org/archives/volume156/number5/patel-2016-ijca-912433.pdf
2. https://www.browserstack.com/guide/what-is-software-metrics

Unit – V: Best Practices and Application (10 Hrs.)

Best Practice of SRE: Benefits and approaches of SRE, SRE during requirements phase, SRE during implementation phase, SRE during Maintenance phase.
Neural Networks for Software Reliability: Introduction, Neural Networks, Neural Networks for software reliability, software reliability growth modeling.

Description: This unit explores the critical methodologies and benefits of Software Reliability Engineering (SRE) throughout the software lifecycle. Students can analyze the role of machine learning in improving software reliability. This unit equips learners with the skills to integrate SRE principles and leverage advanced technologies like neural networks, ensuring robust and reliable software systems.
Examples / Applications / Case Studies:

1. Case study: Create service-level objectives (SLOs) for a hypothetical e-commerce platform. Consider factors like response time, uptime, and failure rates.
2. Conduct a risk analysis for a cloud-based application. Identify potential failure points during design, implementation, and maintenance phases.
3. Build a simple neural network to predict software failure rates based on input variables such as code complexity, test coverage, and historical failure data.

Exercises:

1. Compare traditional reliability models (e.g., Musa's Execution Time Model) with neural network-based predictions. Analyze the differences in precision and applicability.
2. Predict software failures in space missions, improving mission-critical system reliability.

Learning Outcome:

- Analyze SRE and neural networks' role in software reliability.

Web Resources:

1. https://www.sciencedirect.com/science/article/pii/S0164121206001737
2. https://www.blameless.com/the-essential-guide-to-sre

Textbooks:

1. *Software Reliability Engineering*, John D. Musa, second edition, Tata McGraw-Hill, 2005.

2. *Handbook of Software Reliability Engineering*, edited by Michael R. Lyu, published by IEEE Computer Society Press and McGraw-Hill Book Company, 2023.

Reference Books:

1. *Practical Reliability Engineering*, Patric D. T. O'Connor, 4th Edition, John Wiley & Sons, 2003.

2. *Fault Tolerance Principles and Practice*, Anderson and PA Lee, PHI, 1981.

3. *Fault Tolerant Computing - Theory and Techniques*, Pradhan D. K (Ed.): Vol 1 and Vol 2, Prentice Hall, 1986.

4. *Reliability Engineering*, E. Balagurusamy, Tata McGraw-Hill, 1994.

POOL II — COURSES SYLLABUS

24CS720A: Wireless Adhoc & Sensor Networks

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Objectives:

The course should enable the students to:

1. Describe the concepts of ad hoc wireless networks.
2. Analyze different routing protocols of mobile ad hoc networks.
3. Apply the energy management policies in routing algorithms.
4. Implement protocols for location-based QoS.
5. Design and simulate sensor networks and evaluate performance.

Course Outcomes:

1. Students will be able to describe an ad hoc network and analyze various technologies associated with it.
2. Students will be able to analyze various transport layers and analyze various protocols associated with it.
3. Students will apply this knowledge to analyze ad hoc & sensor-based networks and compute various parameters associated with it.

Students will discuss the challenges in designing routing and transport protocols for wireless Adhoc/sensor networks.

Unit I: Ad Hoc Wireless Networks and MAC Introduction

Ad Hoc Wireless Networks and MAC Introduction: Issues in ad hoc wireless networks, issues in designing a MAC protocol for ad hoc wireless networks, design goals of a MAC protocol for ad hoc networks, classifications of MAC protocols.

Unit II: Routing Protocols in Ad Hoc Networks

Routing Protocols in Ad Hoc Networks: Issues in designing a routing protocol for ad hoc wireless networks, classifications of routing protocols, table-driven routing protocol, on-demand routing protocols, hybrid routing protocols, hierarchical routing protocols, and power-aware routing protocols.

Unit III: Energy Management in Ad Hoc Wireless Networks

Energy Management in Ad Hoc Wireless Networks: Energy-Efficient Communication in Ad Hoc Wireless Networks, Ad Hoc Networks Security, Self-Organized and Cooperative Ad Hoc Networking, Simulation and Modeling of Wireless, Mobile, and Ad Hoc Networks, Modeling.

Unit IV: Quality of Service in Ad Hoc Wireless Networks

Quality of Service in Ad Hoc Wireless Networks: Introduction, issues, and challenges in providing QoS in ad hoc networks, classification of QoS solutions, MAC layer solutions, QoS routing protocols, ticket-based, predictive location-based QoS routing protocols.

Unit V: Wireless Sensor Networks

Wireless Sensor Networks: Introduction, sensor network architecture, data dissemination, gathering, MAC protocols for sensor networks self-organizing, hybrid TDMA/FDMA, CSMA-based MAC, location discovery.

Text Books:

1. C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc Wireless Networks-Architectures and Protocols*, New Delhi: Pearson Education, 2013.
2. *Ad Hoc & Sensor Networks* by Stefano Basagni, Silvia Giordano, Ivan Stojmenovic. IEEE Press, A John Wiley & Sons, Inc., Publication 2004.

Reference Books:

1. Feng Zhao and Leonidas Guibas, *Wireless Sensor Networks*. Noida: Morgan Kaufmann Publishers, 2004.
2. C. K. Toh, *Ad Hoc Mobile Wireless Networks*. New Delhi: Pearson Education, 2002.
3. Thomas Krag and SebastinBuettrich, *Wireless Mesh Networking*. Mumbai: O'Reilly Publishers, 2007.

Web References:

1. https://www.elsevier.com/journals/ad-hoc-networks/1570-8705/guide-for-authors
2. https://en.wikipedia.org/wiki/Wireless_ad_hoc_network
3. www.oldcitypublishing.com/journals/ahwn-home/

24CS720B
Generative AI and Large Language Models

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course is designed for Ph.D. scholars to explore the theoretical and practical aspects of generative AI, focusing on models like GANs, LSTMs, VAEs, and Transformers. Scholars will gain in-depth knowledge of key AI/ML concepts, neural networks, and deep learning techniques, with a particular emphasis on generative models used in text, image, and music generation. The course covers advanced topics such as attention mechanisms, large language models like BERT, and fine-tuning for NLP tasks. Through hands-on applications, scholars will generate realistic content and critically evaluate the effectiveness of these models in solving complex, domain-specific challenges, while also exploring ethical considerations and future trends in AI-driven creativity.

Course Objectives

The main objectives of this course are to:

- Explain key concepts in AI/ML, neural networks, deep learning, and advanced models like GANs and LSTMs.
- Describe the concepts of generative AI, large language models, Transformer architecture, attention mechanisms, and fine-tuning BERT for NLP tasks.
- Apply generative AI models like GANs, VAEs, and Transformers to generate realistic images and text for practical applications.
- Apply generative AI techniques, including LSTM, Transformer, and MuseGAN, for generating and evaluating multi-track and polyphonic music compositions.
- Analyze and evaluate generative AI models and techniques for addressing complex challenges in text, music, and creative applications.

Course Outcomes

At the end of the course, the student will be able to:

- CO1: Demonstrate understanding of AI/ML foundations, neural network structures, and advanced models like GANs and LSTMs.
- CO2: Discuss key concepts of generative AI, large language models, Transformer architectures, attention mechanisms, and BERT for NLP tasks.
- CO3: Apply advanced techniques in generative AI for both image and text generation, using models such as GANs, VAEs, and Transformers to solve real-world problems.
- CO4: Apply generative AI models to create and assess complex multi-track music compositions with appropriate evaluation methods.
- CO5: Evaluate generative AI models and apply advanced techniques to adapt these models for addressing specific challenges across various domains.

Course Content

Unit I: Introduction to Neural Networks (7 Hrs.)

- **Foundations of AI and Neural Networks:** History and evolution of AI/ML, deep learning revolution, transfer learning, neural networks, NLP, backpropagation, and advanced architectures like GANs, VAEs, and LSTMs.
- **Advanced Neural Network Architectures:** Introduction to generative AI models: GANs, Variational Autoencoders (VAEs), Attention Mechanisms, and LSTMs.

Description: This unit provides a comprehensive overview of the foundations and advanced concepts in AI and neural networks, covering deep learning, transfer learning, and neural network structures.
Examples/Applications/Case Studies:

- Medical image synthesis using GANs for pulmonary chest X-rays.
- Implementing VAE for anomaly detection in medical imaging.

Exercises:

1. Apply an LSTM model for sentiment analysis on a text dataset and evaluate its classification performance.
2. Train a VAE for anomaly detection in a dataset and evaluate its detection accuracy.
3. Implement a GAN to generate synthetic images from a dataset and assess quality.

Learning Outcome: Understand AI/ML foundations, neural networks, training methods, and advanced models like GANs and LSTMs.

Web Resources:

- Sengupta, S. (2009). *Neural Networks and Applications.* NPTEL.
- Bhanusali, N. K. (2024). *Generative AI and Large Language Models.* SWAYAM.
- Babcock, J., and Ball, R. (n.d.). *Generative AI with Python and TensorFlow 2.* Educative LLC.

Unit II: Introduction to Large Language Models (7 Hrs.)

- **Foundation of Large Language Models:** Overview of generative AI and large language models, basics of attention mechanisms, Transformer architecture, pre-training techniques, and transfer learning strategies.
- **BERT and Advanced Techniques:** Understanding BERT architecture and pre-training objectives, fine-tuning BERT for downstream NLP tasks, and exploration of advanced Transformer architectures.

Description: This unit provides an overview of generative AI, large language models, Transformer architectures, attention mechanisms, pre-training techniques, and fine-tuning BERT for advanced NLP tasks.

Examples / Applications / Case Studies:

1. Implement a document classification system using BERT to categorize documents based on their content automatically. Evaluate the model's performance in terms of classification accuracy.

Exercises:

1. Implement a BERT using TensorFlow2 for text generation tasks. Train the BERT on a dataset of text sequences and generate new text samples.
2. Use transfer learning techniques to fine-tune a pre-trained BERT. Fine-tune the model on a domain-specific dataset and evaluate its performance for text generation tasks.

Learning Outcome:

- Understand generative AI, large language models, Transformer architecture, attention mechanisms, and BERT for NLP tasks.

Web Resources:

1. Chakraborty, T. & Chakraborty, S. (2024, July 28). *Large Language Models*. SWAYAM. Link

2. McMilan, E., Geisinger, V., Lin, J., & Galinkin, E. (2024, July 12). *Large Language Models (LLMs) & Text Generation*. Udacity. Link

Unit III: Image and Text Generation with Generative AI (7 Hrs.)

Topics:

- Image Generation with Generative AI: Introduction to Image Generation, implementing GANs for Image Generation, training and fine-tuning GANs, generating images with VAEs, VIT, advanced techniques in image generation, and image and video generation applications.

- Text Generation with Generative AI: Introduction to Text Generation, LSTM-based text generation, Transformer-based text generation, fine-tuning language models, and text generation applications.

Description: This unit covers image and text generation using generative AI, focusing on GANs, VAEs, LSTMs, and Transformers, along with their training, fine-tuning, and real-world applications.

Examples / Applications / Case Studies:

1. Develop image and video generation applications using trained generative AI models. Use the models to generate art, create deepfakes, or synthesize video content.

2. Develop applications for text generation tasks such as story generation, dialogue generation, or code generation using trained generative AI models.

Exercises:

1. Implement a ViT using TensorFlow2. Train the ViT model on a dataset such as MNIST or CIFAR-10 for image generation tasks.

2. Use transfer learning techniques to fine-tune a pre-trained language model (e.g., GPT, LSTM). Fine-tune the model on a domain-specific dataset and evaluate its performance for text generation tasks.

Learning Outcome:

- Apply generative AI techniques to generate images and text using GANs, VAEs, LSTMs, and Transformers for practical tasks.

Web Resources:

1. Obinguar, A. C. (2023, December 29). *Advanced Image Generation with Generative AI*. Omdena. Link
2. McMilan, E., Geisinger, V., Lin, J., & Galinkin, E. (2024, July 12). *Large Language Models (LLMs) & Text Generation*. Udacity. Link

Unit IV: Music Generation with Generative AI (7 Hrs.)

Topics:

- Music Generation with Generative AI: Introduction to Music Generation, music representation, and LSTM-based music generation.
- Transformer-based music generation, evaluation, fine-tuning, and music composition applications.
- MuseGAN: Overview of MuseGAN architecture, multi-track music generation using MuseGAN, training MuseGAN on polyphonic music datasets, and generating complex music compositions with MuseGAN.

Description: This unit covers generative AI techniques for music composition, focusing on LSTM, Transformer, and MuseGAN models for creating multi-track, polyphonic music and their applications in music generation.

Examples / Applications / Case Studies:

1. Implement a Transformer-based architecture (e.g., MuseGAN, MusicBERT) using TensorFlow2 for music generation. Fine-tune the model on a music dataset and generate novel music compositions.

Exercises:

1. Pre-process and represent music data in a suitable format for music generation tasks.
2. Explore MIDI or audio representations for training generative AI models.
3. Implement an LSTM network using TensorFlow2 for music generation. Train the LSTM model on a dataset of music sequences and generate new musical compositions.

Learning Outcome:

- Apply generative AI techniques, including LSTM, Transformer, and MuseGAN, to generate and compose multi-track, polyphonic music.

Web Resources:

1. Babcock, J., & Bali, R. (n.d.). *Generative AI with Python and TensorFlow 2*. Educative LLC. Link

2. Velardo, V. (The Sound of AI). (2023, November 1). *Generative Music AI Course*. YouTube. Link

Unit – V: Applications (10 Hrs.)

Generative AI Applications: Art and Creativity, Image and Video Generation, Text Generation, Music Composition, Healthcare, Finance. Real-world use cases and challenges in deploying generative AI models.

GPT Models and Applications: Study of GPT architecture and variants. Applications of GPT models in text generation and dialogue systems. Case study-based implementation of GPT-based tasks. GPT-based chatbot enhances E-Shop's customer support service.

Advanced Techniques and Applications: Transfer learning in music generation, Fine-tuning generative models for specific music genres or styles, Ethical considerations in AI-generated music, Future directions and emerging trends in AI-driven music composition.

Examples / Applications / Case Studies:

- ShopEaze implemented a GPT-4-based chatbot to handle customer queries, reducing response times and improving satisfaction. Discuss the benefits, challenges, and strategies for deploying AI chatbots in e-commerce customer support.

- PixelTunes used AI to generate medieval fantasy-style background music for their RPG game. Analyze the benefits, limitations, and strategies for effectively integrating AI-generated music in game development.

Exercises:

1. To design and conceptualize a GPT-based chatbot for a specific domain (e.g., healthcare, finance, or education).

2. Use a generative AI model (e.g., OpenAI Jukebox) to create a 30-second music sample in a specific genre, analyze its alignment with the intended style.

Learning Outcome:

- Utilize GPT and other generative models to create content, while evaluating their ethical impact and practical applications.

Web Resources:

- Dilmegani, C. (2024, November 18). *Top 100+ Generative AI Applications with Real-Life Examples.* Research AI Multiple. https://research.aimultiple.com/generative-ai-applications/
- Baghel, V. (2024, July 23). *Future-Oriented Generative AI Applications for Your Business.* OpenXcell. https://www.openxcell.com/blog/generative-ai-applications/

Books and Web Resources

Textbooks:

1. Rehmani, A. (2024). *Generative AI for Everyone: Understanding the essentials and applications of this breakthrough technology.* Kindle Edition.
2. Dinamani, N. (2024). *Introduction to Generative AI.* Manning Publications.
3. Aggarwal, C. C. (2018). *Neural networks and deep learning.* Springer.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* The MIT Press.
5. Babcock, J., & Bali, R. (2021). *Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs, Transformer models.* Packt Publishing Ltd.
6. Marr, B. (2022). *Generative AI in practice: How AI is changing business and society.* John Wiley & Sons.

Reference Books:

1. Kalin, J. (2018). *Generative Adversarial Networks Cookbook: Over 100 Recipes to Build Generative Models Using Python, TensorFlow, and Keras.* Packt Publishing Ltd.
2. Sprenkler, J. (2021). *Generative AI in Software Development.* Amazon Digital Services LLC.
3. Thakur, K., Baner, H. G., & Patna, A. K. (2024). *Artificial Intelligence and Large Language Models: An Introduction to the Technological Future.* Chapman & Hall.
4. Hay, J., & Walters, G. (2019). *Hand-On Generative Adversarial Networks with PyTorch.* Packt Publishing Ltd.
5. Lyu, Z. (Ed.). (2024). *Applications of Generative AI.* Springer International Publishing AG.

Web Resources:

1. Bansal, N. K. (2024, July 26). *Generative AI and Large Language Models*. SWAYAM. https://onlinecourses.swayam2.ac.in/bn24_ml16/preview

2. Sengupta, S. (2009, September 23). *Neural Networks and Applications*. NPTEL. https://archive.nptel.ac.in/courses/117/105/117105084/

3. Chakraborty, T., & Chakraborty, S. (2024, July 23). *Large Language Models*. SWAYAM. https://onlinecourses.nptel.ac.in/noc23_cs45/preview

4. Obigagior, A. C. (2023, December 29). *Advanced Image Generation with Generative AI*. Omdena. https://www.omdena.com/course/advanced-image-generation-with-generative-ai

5. McMinn, E., Geislinger, V., Liu, J., & Gamaleldin, E. (2024, July 12). *Large Language Models (LLMs) & Text Generation*. Udacity. https://www.udacity.com/course/large-language-models-and-text-generation--ud8138

6. Babcock, J., & Bain, R. (n.d.). *Generative AI with Python and TensorFlow 2*. Educative LLC. https://www.educative.io/courses/generative-ai-with-python-and-tensorflow2

7. Valerdo, V. *The Sound of AI*. (2023, November 1). *Generative Music AI Course*. YouTube. https://www.youtube.com/playlist?list=PL-x47aIMJ4AuPnPGHJTbEjhZMeMo3Jp-D

8. Dilmogeni, C. (2024, November 18). *Top 100+ Generative AI Applications with Real-Life Examples*. Research AllMultiple. https://research.allmultiple.com/generative-ai-applications/

9. Baghel, V. (2024, July 23). *Future-Oriented Generative AI Applications for Your Business*. OpenXcell. https://www.openxcell.com/blog/generative-ai-applications/

24CS720C EDGE COMPUTING

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Networking Fundamentals, Cloud Computing, IoT Basics, Distributed Systems	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

Edge Computing is a cutting-edge paradigm that brings computational capabilities closer to the data sources, minimizing latency and improving real-time processing. It serves as a critical technology for Internet of Things (IoT), 5G, and Artificial Intelligence (AI) applications. This course introduces the foundational concepts of edge computing, architecture, and integration with cloud systems. It emphasizes the design of edge applications and discusses the challenges and innovative solutions associated with edge deployment in various domains, such as smart cities, healthcare, and autonomous systems.

Course Objectives

1. Understand the evolution and significance of edge computing in modern distributed systems.
2. Explore the architecture, components, and enabling technologies of edge computing.
3. Design and implement edge-based solutions for low-latency and resource-constrained applications.
4. Analyze security and privacy challenges in edge computing and implement solutions.
5. Identify real-world use cases and evaluate the performance of edge computing solutions.

Course Outcomes

At the end of the course, the student will be able to:

- CO 1: Understand the need for edge computing and differentiate it from traditional and cloud computing paradigms. [K2]
- CO 2: Analyze edge architecture, protocols, and frameworks for distributed processing. [K4]

- CO 3: Apply edge computing solutions in IoT, AI, and 5G scenarios for real-time processing. [K3]
- CO 4: Develop edge-based applications considering security and privacy aspects. [K5]
- CO 5: Evaluate performance metrics for edge systems and recommend optimization strategies. [K4]

Course Content

UNIT I: Introduction to Edge Computing

- **Evolution of Computing Paradigms:** Overview of centralized (mainframe), distributed, cloud, and edge computing. Differences and comparative benefits of edge, cloud, and fog computing.
- **Role of Edge Computing in IoT and 5G:** How edge computing enables low-latency processing for IoT and 5G networks. Examples: Real-time analytics for IoT sensors and 5G applications.
- **Edge Computing Architecture:** Core components: edge devices (sensors, actuators), edge nodes (gateways, micro data centers). Edge-cloud interaction: data processing, storage, and offloading strategies.
- **Communication Protocols:** Overview of lightweight protocols: MQTT, CoAP, HTTP/REST for edge-to-cloud communication. Use cases: smart homes and industrial IoT.

UNIT II: Edge Computing Frameworks and Platforms

- **Edge Orchestration and Resource Management:** Managing edge resources: dynamic scheduling and load balancing. Role of orchestration tools like Kubernetes in edge environments.
- **Open-Source Frameworks:** EdgeX Foundry: Modular microservices architecture for device interoperability. Azure IoT Edge: Deployment of cloud workloads on edge devices. AWS Greengrass: Enabling AI, data processing, and synchronization on edge.
- **Virtualization and Containerization at the Edge:** Docker for lightweight edge deployments. Kubernetes for managing containerized applications in distributed edge environments.
- **Integration of Edge and Cloud:** Hybrid computing models: pushing tasks between edge and cloud based on latency and resource requirements.

UNIT III: Edge Applications and Use Cases

- **Real-Time Data Processing:** Applications in smart cities: traffic monitoring, pollution control, and public safety. Healthcare use cases: remote patient monitoring and diagnostics.
- **AI and Machine Learning on Edge Devices:** Running inference models (e.g., TensorFlow Lite) on edge devices for real-time decision-making.
- **Predictive Maintenance:** Analyzing sensor data at the edge to predict equipment failures before occurrence.

Case Studies Industrial IoT and smart manufacturing: use of edge for optimizing production lines. Autonomous systems: drones, self-driving cars leveraging edge for navigation and safety.

UNIT IV: Challenges in Edge Computing

- **Latency and Bandwidth Optimization:** Strategies for reducing latency: caching, edge analytics, and network optimization. Bandwidth savings through data filtering and aggregation.
- **Security and Privacy Concerns:** Challenges in securing edge devices: vulnerabilities due to resource constraints. Data privacy: encryption, anonymization, and compliance with regulations (e.g., GDPR).
- **Energy Efficiency and Resource Constraints:** Optimization techniques to reduce energy consumption in battery-operated devices.
- **Fault Tolerance and Resilience:** Techniques to ensure system reliability despite hardware or network failures.

UNIT V: Future Trends and Performance Evaluation

- **Emerging Technologies:** Federated learning: distributed AI model training on edge devices. 6G networks and neuromorphic computing at the edge.
- **Edge Computing for AR/VR and Gaming:** Real-time rendering and processing requirements for immersive experiences.
- **Performance Metrics:** Key metrics: latency, throughput, scalability, and energy efficiency.
- **Optimization Techniques:** Caching strategies to reduce repetitive data fetching. Load balancing and task offloading for optimized performance.

Text Books

1. Flavio Bonomi et al., *"Fog and Edge Computing: Principles and Paradigms,"* Wiley, 2019.
2. Amir M. Rahmani et al., *"Fog Computing in the Internet of Things: Intelligence at the Edge,"* Springer, 2018.

Reference Books

1. Mahadev Satyanarayanan, *"The Edge of Computing,"* Carnegie Mellon University Press, 2021.
2. Peng Zhang, *"Advanced Technologies in Edge Computing Systems,"* Elsevier, 2020.

E-Resources

1. NPTEL: Introduction to IoT: `https://onlinecourses.nptel.ac.in/noc22_c353/preview`
2. Edge Computing Tutorials by Microsoft Azure: `https://learn.microsoft.com/en-us/azure/iot-edge/?view=iotedge-1.5`
3. EdgeX Foundry Documentation: `https://docs.edgexfoundry.org/2.3/getting-started/`
4. EDGE AI FOUNDATION: `https://www.youtube.com/watch?v=VVfq5FkVk`

24CS720D: Quantum Computing

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Linear Algebra, Probability and Statistics, Discrete Mathematics, Design and Analysis of Algorithms	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

Quantum computing is an emerging field that combines principles of quantum mechanics with computational theory to create new ways of processing information. Unlike classical computing, which relies on bits as units of data, quantum computing uses qubits, which can exist in multiple states simultaneously through superposition and entanglement. This allows quantum computers to solve specific complex problems. This course will provide an introduction to Quantum Computation, starting with basic concepts to discussing the quantum circuit model of computation and basic Quantum algorithms that demonstrate the power of computing with quantum bits. Quantum error correction and quantum error mitigation are introduced, which are crucial for the success of quantum algorithms in the NISQ era.

Course Objectives

1. Explain the principles of quantum mechanics that form the basis of quantum computing and distinguish classical computing from quantum computing.
2. Understand quantum bits (qubits) and their representation using quantum states. Analyze basic quantum gates (e.g., Pauli, Hadamard, CNOT) and their operations on qubits.
3. Implement foundational quantum algorithms, such as Deutsch-Jozsa algorithm, Grover's search algorithm, and Shor's algorithm for factoring large numbers, etc., and evaluate the efficiency and real-world implications of quantum algorithms compared to classical counterparts.
4. Explain the significance of quantum error correction in maintaining reliable quantum computations and apply error correction principles in both classical and quantum systems.
5. Identify practical applications of quantum computing in areas like cryptography, optimization, and machine learning.

Course Outcomes

At the end of the course, the student will be able to:

1. Understand quantum mechanics concepts and the distinctions between traditional and quantum computing. [K2]
2. Analyze the quantum circuit model, quantum gates, unitary transformations, and universal gate sets. [K4]
3. Analyze key quantum algorithms like Deutsch, Deutsch-Jozsa, Simon's algorithm, and quantum Fourier transform. [K4]
4. Develop approaches for quantum search scenarios with unknown success probabilities. [K5]
5. Apply error correction principles in both classical and quantum systems, including fault-tolerant quantum computation methods. [K3]

Course Content

Unit I: Foundation
Overview of traditional computing — Church-Turing thesis — circuit model of computation — reversible computation — quantum physics — quantum physics and computation — Dirac notation and Hilbert Spaces — dual vectors — operators — the spectral theorem — functions of operators — tensor products — Schmidt decomposition theorem.

Unit II: Qubits and Quantum Model of Computation
State of a quantum system — time evolution of a closed system — composite systems — measurement — mixed states and general quantum operations — quantum circuit model — quantum gates — universal sets of quantum gates — unitary transformations — quantum circuits.

Unit III: Quantum Algorithms
Superdense coding — quantum teleportation — applications of teleportation — probabilistic versus quantum algorithms — phase kick-back — the Deutsch algorithm — the Deutsch-Jozsa algorithm — Simon's algorithm — quantum phase estimation and quantum Fourier Transform — eigenvalue estimation.

Unit IV: Quantum Algorithms
Order-finding problem — eigenvalue estimation approach to order finding — Shor's algorithm for order finding — finding discrete logarithms — hidden subgroups — Grover's quantum search algorithm — amplitude amplification — quantum amplitude estimation — quantum counting — searching without knowing the success probability.

Unit V: Quantum Computational Complexity and Error Correction
Computational complexity — black-box model — lower bounds for searching — general black-box lower bounds — polynomial method — block sensitivity — adversary methods — classical error correction — classical three-bit code — fault tolerance —

quantum error correction — three- and nine-qubit quantum codes — fault-tolerant quantum computation.

Text Books

1. Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge, 2010, 10th Anniversary Edition.
2. Noson S. Yanofsky and Mirco A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press, 2008, 1st Edition.

Reference Books

1. Eleanor G. Rieffel and Wolfgang H. Polak, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
2. *Quantum Computing in Practice with Qiskit® and IBM Quantum Experience®: Practical Recipes for Quantum Computer Coding at the Gate and Algorithm Level with Python*, Hassi Norlén, Packt Publishing, 2020.

E-Resources

1. https://nptel.ac.in/courses/106106232
2. https://onlinecourses.nptel.ac.in/noc21_cs103/preview
3. https://learning.quantum.ibm.com/
4. YouTube Link: *Introduction to Quantum Computing Complete Course - Quantum Soar*.
5. Quantum Computing Tools: QisKit, QSim, QuEST.

24CST20E: Evolutionary Computing

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Linear Algebra, Probability and Statistics, Discrete Mathematics, Design and Analysis of Algorithms	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

Evolutionary Computing solves complex problems, including optimization, modeling, and simulation problems, as well as search problems. The field of Evolutionary Computing draws inspiration from biology and the natural world, with its origins rooted in Darwinian evolution, genetics, and the concept of survival of the fittest. Evolutionary algorithms consist of several key components, including representation, evaluation functions, population, parent selection, variation operators, survivor selection, initialization, and termination conditions.

Course Aims and Objectives

1. To provide knowledge on basics of Evolutionary Computing and describe its key components.

2. To enable the importance of different types of evolutionary algorithms: evaluation functions, population, parent selection, variation operators, survivor selection, initialization, and termination conditions.

3. Apply evolutionary computing techniques to solve optimization problems, model complex systems, and search for solutions.

4. Analyze and evaluate evolutionary computing algorithms to evaluate their strengths and weaknesses, and select the most suitable algorithm for a given problem.

5. Design and implement evolutionary computing solutions to real-world problems, using programming languages and software tools.

Course Outcomes

At the end of the course, the student will be able to:

1. Understand the fundamentals of Evolutionary Computing. [K2]

2. Analyze and evaluate evolutionary computing algorithms. [K4]
3. Apply evolutionary computing techniques to solve problems. [K3]
4. Analyze and evaluate evolutionary computing algorithms to select the most suitable algorithm for a given problem. [K4]
5. Design and implement evolutionary computing solutions. [K4]

Course Structure

Unit 1:

Theory:

- Problems to Be Solved
- Optimization, Modeling, and Simulation Problems
- Search Problems
- Evolutionary Computing: The Origins
- Brief History
- The Inspiration from Biology
- Darwinian Evolution
- Genetics
- Putting It Together
- Evolutionary Computing: Why?

Description: This unit focuses on Evolutionary Computing, designed to solve complex problems that are difficult to address using traditional methods. These problems fall into three main categories: optimization, modeling, and simulation. Optimization involves finding the best solution among multiple possibilities, while modeling and simulation involve understanding and predicting the behavior of complex systems.

Examples/Applications/Case Studies: *Optimization of Wind Farm Layout using Evolutionary Computing:* Wind energy is a growing source of renewable energy. However, optimizing wind farm layouts to maximize energy production while minimizing costs is a complex problem.

Learning Outcomes:

- Understand the fundamentals of evolutionary computing and its applications.

- Comprehend the biological inspiration behind evolutionary computing (Darwinian evolution, genetics).

Specific Resources:

- E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Second Edition, Natural Computing Series, Springer, 2003.

Unit 2:

Theory:

- What Is an Evolutionary Algorithm?
- Components of Evolutionary Algorithms
- Representation (Definition of Individuals)
- Evaluation Function (Fitness Function)
- Population
- Parent Selection Mechanism
- Variation Operators (Mutation and Recombination)
- Survivor Selection (Replacement)
- Initialization
- Termination Conditions

Description: This unit focuses on Evolutionary Algorithms, computational methods inspired by natural evolution using iterative selection, variation, and survival to optimize solutions. Essential parts include representation, evaluation function, population, parent selection, variation operators, survivor selection, initialization, and termination conditions.

Examples/Applications/Case Studies: *Optimization of Wind Farm Layout using Evolutionary Computing:* Wind energy is a growing source of renewable energy. Optimizing wind farm layouts to maximize energy production while minimizing costs is a complex problem.

Learning Outcomes:

- Apply Evolutionary Algorithms to solve optimization problems.
- Analyze the role of representation, evaluation function, and population size.
- Compare different parent selection mechanisms and variation operators.

Specific Resources:

- E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Second Edition, Natural Computing Series, Springer, 2003.

Unit 3:

Theory:

- Recombination for Binary Representation
- Integer Representation
- Mutation for Integer Representations
- Recombination for Integer Representation
- Real-Valued or Floating-Point Representation
- Mutation for Real-Valued Representation
- Self-adaptive Mutation for Real-Valued Representation
- Recombination Operators for Real-Valued Representation
- Representation
- Permutation Representation
- Mutation for Permutation Representation
- Recombination for Permutation Representation

Description: This unit focuses on Single-Point Crossover, Two-Point Crossover, Uniform Crossover, and Permutation Representation.

Examples/Applications/Case Studies: *Warehouse Optimization:* A company operates a large warehouse with 10 aisles and 50 shelves. Management wants to optimize the layout to minimize travel distance for order picking. The objective is to:

- Assign 20 products to shelves.
- Determine the optimal aisle-shelf layout.

Learning Outcomes:

- Analyze the effects of different representation schemes on EA performance.
- Design and implement EA systems using various representation schemes.
- Evaluate the strengths and limitations of different representation schemes.

Specific Resources:

- E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Second Edition, Natural Computing Series, Springer, 2003.

Unit 4:

Theory:

- Fitness, Selection, and Population Management
- Population Management Models
- Parent Selection
- Survivor Selection
- Selection Pressure
- Multimodal Problems, Selection, and the Need for Diversity

Description: This unit covers Evolutionary Algorithms (EAs), fitness, selection, and population management as crucial components that work together to drive the optimization process.

Examples/Applications/Case Studies: *Wind Farm Optimization:* A wind farm operator wants to optimize the placement of 10 wind turbines on a 100-acre site to maximize energy production.

Learning Outcomes:

1. Understand survivor selection mechanisms (e.g., elitism, age-based selection).
2. Identify factors influencing selection pressure.

Specific Resources:

- E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Second Edition, Natural Computing Series, Springer, 2003.

UNIT 5:

Theory:

- Genetic Algorithms
- Evolution Strategies
- Evolutionary Programming
- Genetic Programming
- Learning Classifier Systems
- Differential Evolution
- Particle Swarm Optimisation
- Estimation of Distribution Algorithms

Description:

The Unit covers Genetic Algorithms (GAs), which optimize using selection, crossover, and mutation; Evolution Strategies (ES), for continuous optimization; Evolutionary Programming (EP), using mutation, selection, and competition; Genetic Programming (GP), evolving computer programs.

Examples/Applications/Case Studies:

A renewable energy company wants to optimize the layout of a new wind farm to maximize energy production.

Problem Description:

- 20 wind turbines
- 100-acre site
- Variable wind speeds and directions
- **Objective:** Maximize energy production

Use the above techniques.

Learning Outcomes:

- Recognize Genetic Programming (GP) and its evolutionary principles.
- Identify Learning Classifier Systems (LCS) and their machine learning integration.

Specific Resources:

1. E. Eiben, J. E. Smith, "Introduction to Evolutionary Computing," Second Edition, Natural Computing Series, Springer, 2003

Textbooks:

1. Jen-Shyang Pan, Jerry Chun-Wei Lin, Pei Hu, Zhigeng Pa, "Genetic and Evolutionary Computing," Proceedings of the Fifteenth International Conference on Genetic and Evolutionary Computing (Volume II), October 6-8, 2023, Kaohsiung, Taiwan.
2. "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications," S. Rajasekaran, G. A. Vijayalakshmi Pai, Prentice Hall of India, 2017.

24CST20F
CYBER FORENSICS

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This comprehensive course provides in-depth knowledge of cyber forensics across multiple platforms, combining theoretical foundations with practical applications. The course covers investigation techniques, evidence handling, and professional documentation across various digital environments.

Course Objectives

The main objectives of this course are to:

- Master fundamental and advanced cyber forensics techniques
- Conduct forensic investigations across multiple platforms
- Handle digital evidence professionally
- Perform network and device analysis
- Create professional forensic reports

Course Outcomes

At the end of the course, the student will be able to:

CO1: Apply fundamental concepts of cyber forensics and analyze digital evidence across different operating systems using standard forensic methodologies.

CO2: Perform network traffic analysis and investigate network-based attacks including email forensics using appropriate evidence collection techniques.

CO3: Execute forensic investigations on mobile devices, storage systems, and IoT devices using standard acquisition and analysis protocols.

CO4: Implement forensic procedures for cloud environments, web applications, and emerging technologies while addressing multi-jurisdictional challenges.

CO5: Demonstrate professional cyber forensic investigation practices through proper case management, documentation, and ethical considerations.

Course Content

Unit – I: Foundations of Cyber Forensics and Digital Evidence (7 Hrs.)

Cyber Forensics: Cybercrime, Basics of cyber forensics, cyber forensics investigation processes, digital evidence, challenges in cyber forensics, skills required for cyber forensics expert.
OS Forensics: Digital Evidence in Windows, File system, Timeline analysis, challenges, Digital Evidence on Macintosh, Digital Evidence on UNIX.

Unit – II: Network and Communication Forensics (7 Hrs.)

Network Forensics Fundamentals: Forensic footprints, Network traffic analysis, Protocol analysis, Network Evidence Collection, Device seizure procedures, Network artifacts, Traffic capture and analysis.
Attack Investigation: ICMP attack analysis, Network intrusion investigation, Email forensics - Header analysis, Email tracking, Evidence extraction.

Unit – III: Mobile and Device Forensics (8 Hrs.)

Mobile Device Forensics: Acquisition protocols, Android forensics, iOS forensics, Manual and physical extraction.
Storage Device Forensics: SSD forensics, Component analysis, Data recovery techniques, Traditional storage forensics, Memory analysis.
Embedded Device Forensics: IoT device investigation, Firmware analysis, Data extraction methods.

Unit – IV: Cloud and Web Forensics (7 Hrs.)

Cloud Forensics: Cloud computing fundamentals, Evidence collection in cloud environments.
Forensics-as-a-Service (FaaS): Multi-jurisdiction challenges.
Web Attack Forensics: Web server forensics, Database forensics, Web application attacks, Malware analysis, Virtual machine forensics.
Container forensics: Blockchain forensics.

Unit – V: Professional Practice and Reporting (7 Hrs.)

Professional Practice: Investigation methodology, Case management, Evidence preservation, Legal considerations.
Reporting: Report structure and format, Documentation standards, Technical writing for digital evidence.
Legal Challenges and Ethics: Current challenges in cyber forensics, Ethics in cyber forensic investigations.

Textbooks

1. Practical Cyber Forensics: An Incident-Based Approach to Forensic Investigations, by Niranjan Reddy.

2. Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, by Casey, E., Academic Press.

Reference Books

10. CyberForensics - Understanding Information Security Investigation by Jennifer Bayuk.

11. Handbook of Digital Forensics and Investigation by Eoghan Casey, 1st Edition.

12. Casey, E. Handbook of Digital Forensics and Investigation, Academic Press.

13. Cyber Forensics from Data to Digital Evidence by Albert J. Marcella, Jr., PhD, CISA, CISM, Frederic Guillossou, CISSP, CCE.

14. Nelson, B., Phillips, A., & Steuart, C. (2014). Guide to Computer Forensics and Investigations, Cengage Learning.

24CS720G
Remote Sensing & GIS

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Data Modelling, Data Integration	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course introduces students to the fundamental concepts of Remote Sensing and Geographic Information Systems (GIS). It covers the principles of remote sensing, satellite image processing, image enhancement, and spatial data analysis. Students will learn the basics of geospatial technologies, including map creation, spatial data management, and interpretation of geospatial data for problem-solving. The course also explores practical applications of GIS and remote sensing in various engineering domains, enabling students to analyze, visualize, and solve complex real-world problems using geospatial tools and techniques.

Course Objectives

The main objectives of this course are to:

- Introduce the fundamental concepts of Remote Sensing and GIS.
- Develop proficiency in analyzing and interpreting spatial and satellite data for decision-making.
- Equip students with practical skills to create, manage, and visualize geospatial data using advanced GIS tools.
- Solve real-world problems using remote sensing and GIS tools while addressing complex engineering and environmental challenges.

Course Outcomes

At the end of the course, the student will be able to:

CO1: Use GIS tools effectively to create, manage, and visualize geospatial data (K2).

CO2: Analyze and interpret spatial and satellite data for decision-making (K2).

CO3: Apply thermal, microwave, and LiDAR remote sensing techniques for environmental monitoring, disaster management, and urban studies (K3).

CO4: Apply geospatial technologies to develop innovative and sustainable solutions (K2).

CO5: Apply advanced image processing techniques to address and resolve complex image analysis challenges (K2).

Course Content

Unit – I: Introduction to Remote Sensing Science & Technology (7 Hrs.)

Physics of Remote Sensing: Principles of Remote Sensing, Physical basis of Remote Sensing, The nature and generation of Electromagnetic Radiation (EMR), Interaction of EMR with the atmosphere and earth's surface features.
Spectra and their characteristics: Spectral signatures and characteristic spectral reflectance curves for rocks, soil, vegetation, and water. Spectral quantities. Far and near-infrared and microwave remote sensing.
Platform and Sensors: Platforms: Types of platforms – ground, airborne, and spaceborne platforms; Orbital geometry, satellite characteristics; satellites for earth observation studies and planetary missions. Sensors: Types and classification of sensors, imaging modes, characteristics of optical sensors, sensor resolution – spectral, spatial, radiometric, and temporal.

Description: This unit is designed for beginners that covers basic concepts in remote sensing principles, EMR interactions, spectral characteristics, platforms, sensors, and their applications in Earth and planetary studies.

Examples / Applications / Case Studies:

1. Analyze spectral reflectance curves to identify vegetation health or soil properties.
2. Use satellite data to map land use/land cover changes over time.

Exercises:

1. Study of satellite imagery in different bands and visual interpretation.
2. Understanding spectral response patterns of different land cover objects.
3. Study of given area in B/W IR, color, and IR colour photographs.

Learning Outcome: Understand the fundamental principles of remote sensing and EMR interactions.

Web Resources:

- Canada Centre for Remote Sensing, *Fundamentals of Remote Sensing*. Notes here.

- Eswar R., *Remote sensing: Principles and Applications*, NPTEL. Last accessed on 17-11-2024, available at NPTEL Link.
- Bharti R., *Remote Sensing and GIS*, NPTEL last accessed on 17-11-2024, available at NPTEL Course.

Unit – II: Thermal, Microwave, LiDAR Remote Sensing (7 Hrs.)

Basics of Thermal Remote Sensing

- Physics of Thermal Remote Sensing
- Kinetic & Radiant temperature
- Emissivity of different materials
- Thermal systems (Characteristics of sensors, Resolutions)
- Characteristics of images and different types of available data products
- Thermal Image Interpretation

Information Extraction from Thermal Imagery

- Apparent thermal inertia mapping
- Brightness and temperature retrieval

Microwave and LiDAR Remote Sensing

- Principles of Microwave Remote Sensing (imaging and non-imaging)
- Microwave Bands
- Active and Passive Microwave Sensors
- Imaging Radar Geometry
- Synthetic Aperture Radar (SAR)
  - SAR versus other Earth Observation Instruments
  - Air-borne & Space-borne SAR Sensors
  - SAR Polarization
  - SAR Resolution Cell
  - Azimuth Resolution
  - Range Resolution

Description

This unit covers the fundamentals of thermal and microwave remote sensing, data products, different types of thermal and microwave sensors, and their resolutions.

Examples / Applications / Case Studies

1. **Urban Heat Island (UHI) Analysis:** Monitoring urban temperature variations to study the heat island effect and its impact on local climate.
2. **Surface Temperature Retrieval:** Calculating land surface temperatures for climate and environmental studies.

Exercises

1. Study of Thermal images and interpretation, computing radiance images from satellite data, derivation of brightness temperature.
2. UHI calculation using thermal data.
3. Study of SAR imagery and interpretation of features.

Learning Outcomes

- Apply thermal and microwave remote sensing data to inform and support decision-making in fields like urban planning, disaster management, and environmental monitoring.

Web Resources

1. Indu, J., *Microwave Remote Sensing in Hydrology.* NPTEL, last accessed on 17-11-2024, available at: https://archive.nptel.ac.in/courses/105/101/105101213/
2. N, Rao., *Thermal Remote Sensing*, last accessed on 17-11-2024, available at: https://incois.gov.in/documents/ITCOocean/itco0423/ppts/L3_1.pdf

Unit – III: Digital Image Processing (7 Hrs.)

Fundamentals: Digital Image, Image Histogram, Look-up Table, Optimal band selection methods (OIF etc.), Color Composites - FCC generation, Image Data formats: BIP, BIL, BSQ, and Common Interchangeable formats. Fundamentals of Image Rectification and Registration.
Image Rectification: Fundamentals of Image Rectification and Registration, Spatial Interpolation, Intensity Interpolation (Nearest neighbor, Bilinear interpolation,

Cubic convolution).

Description: This unit covers fundamentals of digital image processing, including basics of satellite image processing, band selection, and interpolation methods.

Examples / Applications / Case Studies:

1. Image Classification: Using band selection methods like OIF for selecting the most informative bands for classification tasks.
2. Image Registration: Aligning images from different sensors or time periods for change detection.

Exercises:

1. Satellite Image Rectification and Registration.
2. Change detection using satellite data.

Learning Outcome: Explain digital image processing for satellite data and their applications.

Web Resources:

- Arun, S., *Digital image processing of remote sensing data*, NPTEL, last accessed on 17-11-2024, available at NPTEL Course.

Unit – IV: Satellite Image Enhancement and Classification (7 Hrs.)

Image Enhancement: Contrast Enhancement: Linear, Non-Linear Enhancement (Min-Max Stretch, Percentile Stretch, Piecewise Linear Stretch, Histogram Equalization, Gaussian Stretch), Pseudo-color enhancement, Density slicing.
Spectral Indices: NDVI, RVI, SAVI, EVI, NDWI.
Image Classification: Principle of Image Classification, Types of class (Information classes and Spectral classes), Classification Schemes (USGS Classification scheme, LCCS - Land Cover Classification System), Types of Image classification: Supervised Classification, Unsupervised Classification.

Description: This unit covers satellite image enhancement techniques (linear and non-linear), spectral indices for environmental monitoring (NDVI, RVI, SAVI, EVI, NDWI), and land/land cover classification schemes (USGS, LCCS). It explores both supervised and unsupervised classification methods. The unit emphasizes real-world applications of these techniques in resource management and monitoring.

Examples / Applications / Case Studies:

1. NDVI and SAVI: Applied in vegetation monitoring, such as detecting drought conditions, assessing crop health, and forest management.
2. Supervised classification used for urban planning and land use mapping.

Exercises:

1. Spatial filtering and frequency domain filtering.
2. NDVI and NDWI calculation using satellite data.

Learning Outcome: Illustrate image classification and environmental indices calculation for environmental resource monitoring and management.

Web Resources:

- Iyengar, S., *The Joy of Computing using Python*, last accessed on 12-06-2024, available at NPTEL Course.

Unit – V: Principle of Geographic Information System (10 Hrs.)

Fundamental of GIS: Introduction to GIS, Understand the difference between GIS and Information Systems in general, GIS Components and functions of GIS, h/w & s/w requirements, Spatial data and attribute data, their sources, types of attributes (such as nominal, ordinal, interval, ratio), geographical data formats (such as coverage, geodatabase, shape file, grid, dxf, dwg, geo-tiff, GML). Representation of the real world via a vector and raster representation model.
Map Protection and Coordinate System: Data Sources, Data models, Creation of Databases (spatial and non-spatial), Spatial analysis - Interpolation, Buffer, Overlay, Terrain Modeling, and Network analysis.

Description: This unit covers basics of GIS, spatial and attribute data, data formats, coordinate systems, and spatial analysis techniques like interpolation, buffering, and network analysis.

Examples / Applications / Case Studies:

1. Disaster Management: Map hazards, assess risk, and optimize emergency response using terrain and network analysis.
2. Transportation: Plan routes, model traffic networks, and optimize logistics.

Exercises:

1. Geo-referencing and Projection.
2. Linking Spatial & Non-Spatial Data and DBMS.

3. Network Analysis.

4. Raster-based Spatial Analysis.

Learning Outcome: Use advanced problem-solving techniques to solve GIS-related problems.

Web Resources:

- Arun, S., *Geographic information system*, NPTEL, last accessed on 17-11-2024, available at NPTEL Course.

Group Tasks

Task 1: Flood mapping and monitoring using remote sensing and GIS

Task 2: Satellite data fusion, Spatial filtering

Task 3: Satellite image classification and change detection

Task 4: Environmental indices calculation using different satellite data and analysis

Task 5: Site Suitability using raster and vector dataset

Textbooks

1. Chrisman, N.R. (1997). *Exploring Geographic Information Systems*. John Wiley and Sons.

2. Longley, Paul A., Goodchild, Michael F., Maguire, David J., and David W. Rhind (2005). *Geographic Information Systems and Science* (2nd ed.). John Wiley and Sons.

3. Jensen, J. (1995). *Introductory Digital Image Processing: A Remote Sensing Perspective* (2nd ed.). Prentice Hall.

Reference Books

1. Mather, P. M. (1999). *Computer Processing of Remotely Sensed Images: An Introduction*. John Wiley.

2. *Remote Sensing III Edition: American Society of Photogrammetry and Remote Sensing*.

Web Resources

1. Eswar, R., *Remote sensing: Principles and Application*. NPTEL, last accessed on 17-11-2024, available at: https://archive.nptel.ac.in/courses/105/101/105101206/

2. Chinnasamy, S., *Remote sensing and GIS for rural development*. NPTEL, last accessed on 17-11-2024, available at: https://archive.nptel.ac.in/courses/105/101/105101221/

3. Indu, J., *Microwave remote sensing in hydrology*. NPTEL, last accessed on 17-11-2024, available at: https://archive.nptel.ac.in/courses/105/101/105101213/

24CS720H - Digital Speech Processing

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Digital Signal Processing, Signals	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description:

Oral speech is one of the most natural, common, and direct modes of human communication. Since the middle of the last century, speech has become an area of intense research and development (R&D) to facilitate Human-Computer Interaction (HCI). The advancement in computational power, available through devices like PCs, PDAs, and mobile handsets, has significantly boosted this field. Although the man-to-machine interaction in speech mode has not reached a widespread deployment threshold, spoken messages directly processed by machines necessitate further research in speech science and technology development.

This course provides foundational knowledge of speech production, perception, and signal processing in the digital domain.

Course Objectives:

1. Provide a thorough understanding of the fundamentals of speech processing.
2. Discuss various quantization methods and Pulse Code Modulation techniques for digital speech representation.
3. Acquire knowledge of techniques for transforming speech waveforms to the frequency domain using homomorphic processing.
4. Understand Text-to-Speech synthesis methods and Automatic Speech Recognition techniques.
5. Explore various speech processing systems for man-machine communication via voice.

Course Outcomes:

At the end of the course, the student will be able to:

- CO1: Understand the fundamentals of speech processing. (K1)
- CO2: Illustrate quantization methods used in the digital representation of the speech waveform. (K2, K3, K4)
- CO3: Gain knowledge of homomorphic speech processing and linear predictive analysis.
- CO4: Illustrate speech synthesis methods and Automatic Speech Recognition. (K2, K3)
- CO5: Study speech recognition and speaker recognition systems. (K1)

Course Structure:

Unit 1: Fundamentals of Digital Speech Processing

- Discrete time signals and systems
- Digital filters
- Sampling
- The process of speech production
- The acoustic theory of speech production
- Digital models of speech signals

Description:
This unit introduces the fundamental concepts of discrete time signals, sampling, and an overview of the process of speech production, along with the acoustic theory of speech production.

Example/Application/Case Studies: Basics of Acoustic Theory
Exercise/Project: Schematic model of the vocal tract system
Learning Outcomes:

- Learn various signals, filters, and sampling.
- Get a detailed understanding of the speech chain and phonetic representation.
- Study models of speech production.

Specific Resources:

1. Rabiner, L.R., & Schafer, R.W. (2003). *Digital Processing of Speech Signals*, Pearson Education India.
2. NPTEL course on Digital Speech Processing: `https://archive.nptel.ac.in/courses/117/105/117105145/`

Unit 2: Digital Representation of the Speech Waveform

- Instantaneous quantization
- Quantization for optimum SNR
- Adaptive quantization
- General theory of Differential quantization
- Delta modulation
- Differential PCM
- Short term Fourier analysis
  - Design of digital filter banks
  - Pitch detection
  - Analysis-Synthesis systems

Description:
This unit discusses the entire digital representation of speech waveform and its various quantization techniques particularly used in speech processing. It also covers some modulation techniques and short-term Fourier analysis so that students will understand pitch detection and synthesis systems.

Example/Application/Case Studies:
A case study about 2-bit quantizer characteristics and code word assignment.

Exercise/Project:
Implementation of a linear delta modulator.

Learning Outcomes:

- Learn quantization techniques and their importance in speech processing.
- Learn modulation techniques and short-term Fourier analysis.

Specific Resources:

1. Foundations and Strengths in Signal Processing, Vol 1, No 1-2, now Publishers Inc., USA.
2. Rabiner, L.R., & Schafer, R.W. (2003). *Digital Processing of Speech Signals*, Pearson Education India.
3. NPTEL course on Digital Speech Processing: `https://archive.nptel.ac.in/courses/117/105/117105145/`

Unit 3: Homomorphic Speech Processing

- Homomorphic systems for convolution
- The complex Cepstrum of speech
  - Definition of Cepstrum
  - Short time Cepstrum
  - Computation of Cepstrum
  - Short time homomorphic filtering of speech
  - The role of Cepstrum
- Linear predictive analysis
  - Linear prediction and speech model
  - Computing the prediction coefficients
  - The Levinson-Durbin recursion
  - LPC spectrum
  - Equivalent representations
  - The role of linear prediction

Description:
This unit details homomorphic speech processing and linear predictive analysis. It explores the use of the short-time Cepstrum as a representation of speech and as a basis for estimating the parameters of the speech generation model. Linear predictive analysis is one of the most powerful and widely used speech analysis techniques. It presents a formulation of the ideas behind linear prediction and discusses some of the issues involved in its practical applications.

Example/Application/Case Studies:
A case study on the impact of Cepstrum in Pattern Recognition.

Exercise/Project:
Implementation of the Levinson-Durbin algorithm.

Learning Outcomes:

- Gain knowledge about Cepstrum and its role in speech processing.
- Understand linear prediction and speech models and the Levinson-Durbin recursion.

Unit 4: Synthesis and Recognition Methods

- Text-to-Speech synthesis methods
  - Text analysis
  - Evolution of speech synthesis systems
  - Unit selection methods
  - TTS applications
  - TTS future needs
- Automatic speech recognition (ASR)
  - The problem of ASR
  - Building a speech recognition system
  - The decision process in ASR
  - Representative recognition performance
  - Challenges in ASR technologies

Description:
This unit discusses converting ordinary text messages into intelligible and natural-sounding synthetic speech by synthesis methods to transmit information from a machine to a human user. It also deals with the process of speech recognition by machine, essentially reversing text-to-speech problems.

Example/Application/Case Studies:
A case study on conceptual models of speech production and speech recognition.

Exercise/Project:
Execution of Baum-Welch algorithm training procedure.

Learning Outcomes:

- Acquire knowledge about various speech synthesis methods regarding text analysis and unit selection methods.
- Understand automatic speech recognition and the Baum-Welch algorithm.

Unit 5: Man–Machine Communication by Voice

- Voice response systems
- Speaker recognition systems
  - Speaker verification systems
  - Speaker identification systems
- Speech recognition systems
  - Isolated digit recognition system

Description:
This unit elaborates the broad area of speech applications, i.e., the various ways in which the basic models and the associated parameters derived from them are used in an integrated system whose purpose is to transmit or automatically extract information from the speech signal. It also discusses representative examples of digital speech processing systems and how digital processing techniques are used in such systems.

Example/Application/Case Studies:
A case study on a large vocabulary, speaker-dependent word recognition system.

Exercise/Project:
Develop the basic model of:

1. Voice response system
2. Speaker recognition system
3. Speech recognition system

Learning Outcomes:

- Learn the speech application systems.
- Gain knowledge about how digital processing techniques are used in speech processing.

Specific Resources:

1. Rabiner, L.R., & Schafer, R.W. (2003). *Digital Processing of Speech Signals*, Pearson Education India.
2. NPTEL course on Digital Speech Processing: https://archive.nptel.ac.in/courses/117/105/117105145/

Textbooks:

1. Rabiner, L.R., & Schafer, R.W. (2010). *Theory and Application of Digital Speech Processing*, Pearson Education India.

2. Rabiner, L.R., & Schafer, R.W. (2003). *Digital Processing of Speech Signals*, Pearson Education India.

3. Dr. Shaila D Apte (2019). *Speech and Audio Processing*, Wiley.

4. MacLoughlin, Ian (2009). *Applied Speech and Audio Processing with MATLAB Examples*, Cambridge University Press.

*References:

1. Quatieri, T.F. (2003). *Discrete-Time Speech Signal Processing: Principles and Practice*, Pearson Education India.

2. Foundations and Strengths in Signal Processing, Vol 1, No 1-2, now Publishers Inc., USA.

3. NPTEL course on Digital Speech Processing: `https://archive.nptel.ac.in/courses/117/105/117105145/`

4. University of California course material: `https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/speech%20course.html`

24CS7201 KNOWLEDGE GRAPHICS

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Data Modelling, Data Integration	**Semester End Evaluation:** 100
Total Marks: 100	

Course Content

UNIT I

- **Introduction to Knowledge Graphs:** Overview, Definition, and significance of KGs, Applications: Search engines, recommendation systems, natural language processing (NLP), and scientific research.

- **Fundamentals of Graph Theory:** Graph structures: Nodes, edges, and properties, Types of graphs: Directed, undirected, weighted, bipartite, etc., Graph traversal algorithms: BFS, DFS, Tools & Libraries: NetworkX, Neo4j.

UNIT II

- **Semantic Web Technologies:** RDF (Resource Description Framework), SPARQL (Query language for RDF), OWL (Web Ontology Language).

- **Knowledge Representation:** Ontologies: Design and construction, Taxonomies vs. ontologies, Entity-relation models.

UNIT III

- **Building Knowledge Graphs:** Data extraction: Structured, semi-structured, and unstructured sources, Entity linking and disambiguation, Relation extraction and knowledge fusion.

- **Querying and Reasoning:** Query optimization for KGs, Logical reasoning: Deductive, inductive, and abductive reasoning, Rule-based and machine learning-based reasoning.

UNIT IV

- **Machine Learning with Knowledge Graphs:** Graph embeddings (e.g., Node2Vec, TransE, DistMult), Knowledge graph completion, Applications in NLP and computer vision.

- **Knowledge Graph Evaluation:** Metrics for KGs: Precision, recall, coverage, and consistency, Benchmark datasets: Freebase, YAGO, Wikidata.

UNIT V

- **Scalability and Optimization:** Distributed graph processing (e.g., Pregel, GraphX), Optimizing storage and retrieval, Streaming graph updates.
- **Ethical Considerations and Bias in Knowledge Graphs:** Addressing bias in KGs, Ensuring data privacy and compliance, Transparency in automated reasoning.

Text Books

1. *"Knowledge Graphs: Fundamentals, Techniques, and Applications"*
   Authors: Mayank Kejriwal, Craig Knoblock, Pedro Szekely
2. *"Foundations of Semantic Web Technologies"*
   Authors: Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
3. *"Knowledge Graphs"*
   Authors: Dean Allemang, James Hendler

Reference Books

1. *"Knowledge Graphs: Fundamentals, Techniques, and Applications"*
   Authors: Mayank Kejriwal, Pedro A. Szekely, Craig A. Knoblock
   Overview: This book provides a solid foundation in knowledge graphs, covering core concepts, methods for construction, and real-world applications.
2. *"Knowledge Graphs and Semantic Web: Foundations, Technologies, and Applications"*
   Editors: Pascal Hitzler, Aidan Hogan, Markus Krötzsch
   Overview: A collection of foundational papers and methods in semantic web and knowledge graph development. Great for understanding the theoretical basis.
3. *"Building Knowledge Graphs"*
   Authors: Jesús Barrasa, Jim Webber
   Overview: Focuses on creating and utilizing knowledge graphs with practical examples using graph databases like Neo4j.

E-Resources and Other Digital Material

1. **Wikidata**
   Open, structured knowledge base by Wikimedia.
   URL: `https://www.wikidata.org`
   Use: Semantic data linking, entity extraction, general-purpose relationships.

2. **DBpedia**
   Structured data extracted from Wikipedia.
   URL: `https://www.dbpedia.org`
   Use: Ontology building, general domain queries.

3. **Freebase**
   Discontinued but available as a static dump.
   URL: `https://developers.google.com/freebase`
   Use: Large-scale entity relationship datasets.

24CS720J
EXTENDED REALITY AND METAVERSE

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites:	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course delves into Extended Reality (XR)—an umbrella term encompassing Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR)—and the emerging concept of the Metaverse, a shared digital universe of interconnected virtual spaces. Students will explore the technologies, design principles, and societal implications of XR and the Metaverse, with a focus on building immersive, interactive environments.

Course Objectives

The main objectives of this course are to:

- Teach the basic concepts of AR, VR, and MR technologies and their convergence into XR.
- Explain the challenges of Extended Reality and its evolution.
- To understand multi-technology perspective in Metaverse and its challenges.
- Discuss the application of Virtual Reality and Augmented Reality in Metaverse.
- To identify recent developments and focus on research initiatives in Metaverse.

Course Outcomes

At the end of the course, the student will be able to:

- **CO1:** Understand the foundational concepts of Extended Reality (XR), including AR, VR, and MR, and their integration into the Metaverse. (K2)
- **CO2:** Understand the evolution and challenges of Extended Reality. (K2)
- **CO3:** Understand the architecture and characteristics of Metaverse. (K2)
- **CO4:** Apply Virtual Reality and Augmented Reality in Metaverse. (K3)

- **CO5:** Use the Metaverse to solve real-world problems. (K3)

Course Content

Unit I: Extended Reality (7 Hrs.)

Introduction: XR at the forefront of technology, Extended reality spectrum, Incredible possibilities of XR.

Extended Reality: A world on XR definitions, Augmented reality: The most accessible of the XR technologies, stepping into immersive environments with Virtual Reality, merging the real and digital.

Unit – II: Evolution and Challenges of Extended Reality (7 Hrs.)

Evolution of Extended Reality: Tracing the XR Timeline, Rapid Evolution in XR Hardware, An Evolution Fueled by other Tech Trends

Challenges in XR: Legal and modern Concerns, Privacy and Security Concerns, Health Concerns, The need of responsible XR

XR in Everyday Life and Business: XR in Everyday Life, XR in the Workplace.

Unit – III: Understanding Metaverse (7 Hrs.)

Introduction: Fundamentals of metaverse, phases of development of the metaverse, characteristics, components and architecture of metaverse, metaverse engine, recognition and rendering.

Technologies of Metaverse: Content creation modes in metaverse, Technologies of metaverse, multi-technology convergence perspective, communication and computing infrastructure.

Unit – IV: Virtual Reality and Augmented Reality in Metaverse (7 Hrs.)

Virtual Reality in Metaverse: Recruitment, Coordination, New target, Preparation for Virtual Reality, Microsoft and the concept of Digital Twins.

Augmented Reality in Metaverse: AR: The Metaverse heart, AR updates via Spark, E-commerce using Augmented Reality, AR in military, Ubiquitous Computing.

Unit – V: Risks and Challenges of Metaverse (7 Hrs.)

Challenges of Metaverse: Legal Ramifications, Acceptance of Marketing, Data exchange, Export and localization of data, Rights to intellectual property, Act Digital services, Act on Digital Markets, Terrifying Dangers of Metaverse, Leaving Reality, Health concerns.

Future of Metaverse: Shopping and Business, Education, Advertising, Healthcare, Workplace and Office, Entertainment and Media.

Reference Books:

1. Park, Sang-Min, and Young-Gab Kim, "A metaverse: Taxonomy, components, applications, and open challenges", IEEE Access, 2022.

2. Chen, Shu-Ching, "Multimedia research toward the metaverse", IEEE MultiMedia, 2023.

3. C. Hackl, "Navigating the Metaverse: A Guide to limitless Possibilities in a Web 3.0 World", 2022, Wiley.

4. Terry Winters, "The Metaverse: Prepare Now for the Next Big Thing", 2021.

5. Terry, Quainstorm and Scott Keeney, The metaverse handbook: Innovating for the internet's next tectonic shift, John Wiley & Sons, 2022.

6. Rostami, Sajjad, and Martin Maier, "The metaverse and beyond: implementing advanced multiverse realms with smart wearables", IEEE Access, 2023.

7. Asra Abdelhafez, Dalia Ezzat, Ashraf Darwish, Aboul Ella Hassanien, Metaverse for Brain Computer Interface: Towards New and Improved Applications, Springer, 2023.

8. Aboul Ella Hassanien, Ashraf Darwish, Mohamed Torky, "The Future of Metaverse in Virtual Era and Physical World", Springer, 2023.

9. Lacity, Mary C., and Steven C. Lippin. Blockchain Fundamentals for Web 3.0, University of Arkansas Press, 2022.

Web Resources:

1. Prof Rashmi Gaur, *Online communication in the digital age-Virtual reality and the Metaverse*, IIT Roorkee, last accessed on 15-11-2024, available at `http://www.dicman.iitr.ac.in/met/courses/video/190107394L52.html`

2. *Metaverse Architecture: A Guide for Creators & Visionaries*, last accessed on 15-11-2024, available at `https://medium.com/@ibmsolutions08/metaverse-architecture-a-guide-for-creators-visionaries-eaa5366dd69`

3. *Extended reality and Everything you need to know*, Roundtable learning, last accessed on 15-11-2024, available at https://roundtablelearning.com/what-is-extended-reality-everything-you-need-to-know/

4. Rahmat M., *Course in Udemy: AR, VR, XR, MR, Metaverse and Spatial Computing Masterclass*, Unlocking The Future: Mastering Extended Reality (XR) and Navigating the Metaverse and Spatial Computing Era, last accessed on 15-11-2024, available at https://www.udemy.com/course/ar-vr-xr-mr-metaverse-and-spatial-computing-masterclass

5. Michael Nebeling, *Extended Reality for Everybody Specialization*, University of Michigan Course in Coursera, last accessed on 15-11-2024, available at https://www.coursera.org/specializations/extended-reality-for-everybody

Textbooks:

1. Marr, B. (2021). *Extended reality in practice: 100+ amazing ways virtual, augmented and mixed reality are changing business and society* (1st ed.). John Wiley & Sons.

2. Andrew, C. (2022). *Metaverse for Beginners: A Guide to Help You Learn About Metaverse, Virtual Reality and Investing in NFTs.*

3. Bin, M. (2022). *The metaverse and how it will revolutionize everything*, Liveright Publishing.

4. Klon, V. (2024). *Metaverse Made Easy: A Beginner's Guide to the Metaverse: Everything you need to know, Metaverses, NFT and Games.* Livr on Klon Inc.

24CS720K

SECURE SOFTWARE ENGINEERING

Course Category: Pool II - Ph.D	**Credits:** 3
Course Type: Theory	**Lecture-Tutorial-Practice:** 3 - 0 - 0
Prerequisites: Basic programming skills (e.g., Java, C++), Familiarity with software development life cycles, Basic knowledge of computer security concepts	**Semester End Evaluation:** 100
Total Marks: 100	

Course Description

This course provides an introduction to the principles and practices of secure software engineering, focusing on the development of secure software systems. Students will learn how to design, implement, and test software systems that are resistant to attacks and protect sensitive data.

Course Objectives

The main objectives of this course are to:

- Understand the problems faced by software practitioners and discuss the software reliability engineering approach.
- Design, develop, and test reliable software systems that meet their requirements and operate effectively in real-world environments, using a range of reliability engineering techniques and strategies.

Course Outcomes

At the end of the course, the student will be able to:

CO1: Understand software security, secure coding practices, and assurance in software systems.

CO2: Acquire skills to integrate essential security principles.

CO3: Develop the ability to efficiently plan, allocate time, and run tests, ensuring all critical scenarios are tested.

CO4: Apply UML diagrams for representing security requirements, business processes, and physical security considerations.

CO5: Identify and resolve security vulnerabilities efficiently using automated tools for UML.

Course Content

Unit – I: (7 Hrs.)
The Problem, System Complexity: The Context within Which Software Lives. Software Assurance and Software Security. The Role of Processes and Practices in Software Security. Threats to Software Security: Sources of Software Insecurity. The Benefits of Detecting Software Security Defects Early: Making the Business Case for Software Security. Current State.

Examples / Applications / Case Studies:

- Case Study: The SolarWinds Supply Chain Attack
- Automated Security Testing Tools

Exercises:

1. Learn how to spot and fix vulnerabilities in code.
2. Learn how to conduct penetration testing on a vulnerable application.

Learning Outcome:

Learners will be able to focus on secure software development, software assurance, and software security.

Web Resources:

1. http://study.com/articles/List_of_Free_Online_Software_Engineering_Courses.html
2. http://www.coursera.org/course/softwaresec

Unit – II: (7 Hrs.)

Introduction: Defining Properties of Secure Software: Core Properties of Secure Software, Influential Properties of Secure Software. How to Influence the Security Properties of Software: The Defensive Perspective, The Attacker's Perspective. How to Assert and Specify Desired Security Properties: Building a Security Assurance Case.

Description: This topic describes key aspects of secure software, focusing on core properties such as confidentiality, integrity, and availability. It explores how these properties can be influenced from both the defensive and attacker's perspectives. Additionally, it covers methods to assert and specify security properties, emphasizing the development of a security assurance case to justify the software's security features and demonstrate its effectiveness in mitigating vulnerabilities. This approach ensures systematic integration of security in software development.

Examples / Applications / Case Studies:

1. Security Assurance Case Example
2. Securing cloud applications by enforcing the least privilege principle and fail-safe mechanisms to prevent catastrophic system failures if an attack bypasses certain defenses.

Exercises:

1. Threat Modeling Exercise
2. Design Review Exercise

Learning Outcome: Able to design, develop, and evaluate software systems with robust security, ensuring they are resistant to known and emerging threats.

Web Resources:

1. http://study.com/articles/List_of_Free_Online_Software_Engineering_Courses.html
2. http://www.coursera.org/course/softwaresec

Unit – III: Testing (7 Hrs.)

Description: The process involves planning and allocating test time for the current release, running tests to identify failures, analyzing output to detect deviations, determining which deviations are failures, pinpointing where failures occurred, and guiding reliability improvements. It includes tracking reliability growth, estimating failure intensity, using failure patterns to refine tests, and certifying system reliability. Deploying SRE involves mastering core principles, persuading stakeholders of its value, executing deployment strategies, and leveraging consultants for expert guidance, ensuring a seamless integration of testing and reliability practices into operational workflows.

Examples / Applications / Case Studies:

1. Case Study: Agile Testing in a CI/CD Pipeline
2. Example: Failure Analysis in a Banking System
3. Application: SRE in E-commerce

Exercises:

1. Allocate Test Time: Design a test plan for a mobile app release, distributing test time across functional, performance, and regression tests.

2. Analyze Failures: Given test output data with deviations, identify which are critical failures and suggest corrective actions.

3. Track Reliability Growth: Simulate failure data across testing cycles and plot reliability improvements over time.

4. SRE Deployment Plan: Create a roadmap for introducing SRE into an organization, including stakeholder engagement, training, and initial focus areas.

Learning Outcome:

- Develop skills in planning and executing systematic testing workflows, including identifying and resolving failures.
- Gain proficiency in analyzing reliability metrics like failure intensity to guide testing and improve system robustness.
- Understand the principles and practical applications of SRE to enhance operational reliability.
- Learn how to effectively communicate the value of SRE to stakeholders and implement deployment strategies tailored to organizational needs.
- Acquire hands-on experience with exercises and case studies, translating theory into practical knowledge for real-world systems.

Web Resources:

1. https://cloud.google.com/SRE
2. https://www.linkedin.com/learning

Unit – IV: UML for Security (7 Hrs.)

Description: Using UML for security involves leveraging UML diagrams to define and analyze security requirements, model security-critical business processes, and address aspects like physical security, critical interactions, and system risks. It includes analyzing models with precise notations and formal semantics to identify security vulnerabilities and opportunities for enhancement. Model-based security engineering with UML, with the UMLsec profile, enables the integration of security considerations directly into system designs, guided by secure design principles and the application of established security patterns for robust and resilient system architectures.

Examples / Applications / Case Studies:

1. Example: Securing Online Payment Systems

2. Case Study: Access Control for Enterprise Systems
3. Case Study: E-Government System
4. Case Study: Secure Healthcare System

Exercises:

1. Diagramming Security Requirements
2. Security Analysis Using UML for Banking System

Learning Outcome:

- Develop the ability to model security requirements using UML diagrams effectively.
- Understand how to apply the UML profile to integrate security directly into system designs.
- Gain skills in identifying and analyzing vulnerabilities within UML models using formal semantics.
- Learn how to apply secure design principles and security patterns in creating robust systems.
- Build experience in securing real-world applications through exercises and case studies.

Web Resources:

1. https://www.umlsec.org
2. https://www.ibm.com/developerworks

Unit – V: Best Practices and Application (10 Hrs.)

Applications: Secure channel - Developing Secure Java programs - more case studies. Tool support for UML Sec - Extending UML CASE TOOLS with analysis tools - Automated tools for UML SEC. Formal Foundations - UML machines - Rely guarantee specifications - reasoning about security properties.

Description: Secure application development with UMLsec involves modeling and analyzing secure channels, implementing secure programming practices (e.g., in Java), and leveraging case studies to address real-world scenarios. Tools extend UML CASE tools with features like automated analysis to identify and mitigate security vulnerabilities. Formal foundations include UML machines, rely-guarantee specifications that provide theoretical support for reasoning about security properties, ensuring compliance with security requirements and enabling automated verification of models.

Examples / Applications / Case Studies:

1. Example: Secure Channel Design
2. Example: Developing Secure Java Programs
3. Case Study: Secure Banking Transactions
4. Case Study: E-Commerce Payment Gateway
5. Case Study: IoT Security

Exercises

1. Tool-Based Security Analysis
2. Secure Java Program Design
3. Secure Channel Modeling

Learning Outcomes

- Understand how to model secure channels and apply secure design principles using UML diagrams.
- Develop secure applications by integrating formal UML analysis with programming practices like secure coding in Java.
- Gain insights from real-world case studies into applying UML and formal foundations in different domains.
- Learn to use automated tools for UML analysis to identify and resolve security vulnerabilities efficiently.
- Acquire skills in formal reasoning about security properties using rely-guarantee specifications and UML machines.

Web Resources

1. Secure UML and UMLsec Overview
2. Secure Coding Guidelines for Java

Textbooks

1. *Software Security Engineering: A Guide for Project Managers* by Julia H. Allen, et al., May 11, 2008.

2. John Musa, *Software Reliability Engineering*, 2nd Edition, Tata McGraw-Hill, 2005 (Units I, II, and III).

3. Jan Jürjens, *Secure Systems Development with UML*, Springer, 2004 (Unit IV and V).

Reference Books

1. *Software Testing and Continuous Quality Improvement* by William E. Lewis: A practical guide for planning and executing software testing processes, Publisher: Auerbach Publications, First Edition: 2000; Second Edition: 2004.

2. *Foundations of Software Testing* by Aditya P. Mathur: Comprehensive coverage of test planning, analysis, and execution, Publisher: Pearson Education, 2008 (First Edition).

3. *Site Reliability Engineering: How Google Runs Production Systems* by Niall Richard Murphy et al.: A detailed look at SRE principles, case studies, and practices, Publisher: O'Reilly Media, 2016.

4. *The SRE Workbook* by Betsy Beyer et al.: Practical exercises and real-world applications of SRE concepts, Publisher: O'Reilly Media, Year of Publication: 2018.

5. *UMLsec: Extending UML for Secure Systems Development* by Jan Jürjens, Publisher: Springer, Year of Publication: 2004.

6. *Security Patterns in Practice* by Eduardo B. Fernandez, Publisher: Wiley, 2013.

7. *Secure Coding in Java* by Ed Seaver and Chris King, Publisher: Addison-Wesley, 2009.