MASTER OF COMPUTER SCIENCE

2-YEAR FULL TIME PROGRAMME

RULES, REGULATIONS AND COURSE CONTENTS

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF MATHEMATICAL SCIENCES
UNIVERSITY OF DELHI
DELHI-110007
2024

MASTER OF COMPUTER SCIENCE
2-YEAR FULL TIME PROGRAMME

1. **AFFILIATION**

The proposed programme shall be governed by the Department of Computer Science, Faculty of Mathematical Sciences, University of Delhi, Delhi-110007.

2. **PROGRAMME STRUCTURE AND OUTCOMES**

The M.Sc. Computer Science programme is divided into two parts as under. Each part will consist of two semesters to be known as Semester-1 and Semester-2.

		Semester-1	Semester-2
Part-I	First Year	Semester-I	Semester-I
Part-II	Second Year	Semester-II	Semester-II

The Programme outcomes are as follows:

- Prepare the students to take up a career in the highly competitive IT industry with research and development skills.
- Equip the students with comprehensive knowledge of the current trends in computer science.
- The choice of courses from a wide list of specialized Courses would allow the students to opt for and follow the career path they have dreamed of.

3. **STRUCTURE OF THE PROGRAMME**

SUMMARY:

The schedule of Courses prescribed for various semesters shall be as follows:

Semester	Core Courses			Elective Course			Open Elective Course			Total Credits
	No. of Courses	Credits (L+T+P)	**Total Credits**	No. of Courses	Credits (L+T+P)	**Total Credits**	No. of Courses	Credits (L+T+P)	**Total Credits**	
I	6	15+0+7	**22**	0	0+0	**0**	0	0+0+0	**0**	**22**
II	5	12+0+6	**18**	1	3+0+1	**4**	0	0+0+0	**0**	**22**
III	1	0+0+4	**4**	3	9+0+3	**12**	1	3+0+1	**4**	**20**
IV	Major project	20	**20**	0	0+0	**0**	0	0+0+0	**0**	**20**
Total Credits for the Course			**64**			**16**			**4**	**84**

DETAILS:

Part-I Semester I

Semester I

	Number of core courses				5
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC101	Design and Analysis of Algorithms	3	0	1	**4**
MCSC102	Artificial Intelligence and Machine Learning	3	0	1	**4**
MCSC103	Information Security	3	0	1	**4**
MCSC104	Mathematical Foundations of Computer Science	3	0	1	**4**
MCSC105	Data Mining	3	0	1	**4**
MCSC106	Software Tools	0	0	2	**2**
	Total credits in core course				22
	Number of elective courses				0
	Total credits in elective course				0
	Number of open electives				0
	Total credits in elective course				0
	Total credits in Semester I				**22**

Part-I Semester II

Semester II					
	Number of core courses				4
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC201	Artificial Neural Networks	3	0	1	**4**
MCSC202	Deep Learning	3	0	1	**4**
MCSC203	Internetworking with TCP/IP	3	0	1	**4**
MCSC204	Cloud Computing	3	0	1	**4**
MCSC205	Reading Skills	0	0	2	**2**
	Total credits in core course				18
	Number of elective courses				1
		Theory	Tutorial	Practical	**Total**
	Elective 1	3	0	1	**4**
	Total credits in elective courses				4
	Number of open electives				0
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective 1	0	0	0	**0**
	Total credits in open elective				0
	Total credits in Semester II				**22**

List of Elective Courses

List of Electives for Semester II		
Course Code	**Course Title**	**L-T-P**
MCSE201	Digital Image Processing	3-0-1

MCSE202	Compiler Design	3-0-1
MCSE203	Natural Language Processing	3-0-1
List of Open Electives for Semester II		
Course Code	**Course Title**	**L-T-P**

Part-II Semester III

At least two electives out of those offered by the Department as mentioned in the list of electives and one elective offered by other Departments as approved by the Department.

Semester III					
	Number of core courses				1
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCSC301	Minor Project	0	0	4	**4**
	Total credits in core course				4
	Number of elective courses				3
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Elective course 1	3	0	1	**4**
	Elective course 2	3	0	1	**4**
	Elective course 3	3	0	1	**4**
	Total credits in elective courses				12
	Number of open electives				1
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective 1	3	0	1	**4**
	Total credits in open elective				4
	Total credits in Semester III				20

List of Elective Courses

List of Elective Courses for Semester III		
Course Code	**Course Title**	**L-T-P**
MCSE301	Cyber Physical Systems	3-0-1
MCSE302	Graph Theory	3-0-1
MCSE303	Network Science	3-0-1
MCSE304	Information Retrieval	3-0-1
MCSE306	Soft Computing	3-0-1
MCSE307	Quantum Computing	3-0-1
MCSE308	Software Quality Assurance and Testing	3-0-1
MCSE309	Social Networks	3-0-1
List of Open Courses for Semester III		
Course Code	**Course Title**	**L-T-P**
MCSO301	Data Analysis and Visualization	3-0-1
MCSO302	Data Science	3-0-1
XXXXXXX	Inter-Departmental Elective	X-X-X

* **L-T-P: Lectures -Tutorials- Practical**

** Only for students of other departments
***As per the elective offered by the concerned Department.

Part-II Semester IV

Semester IV		
	Number of core courses	1
Course Code	**Course Title**	**Credits in each core course**
MCSC401	Major Project	20
	Total credits in core course	20
	Number of elective courses	0
	Total credits in elective courses	0
	Number of open electives	0
	Total credits in open elective	0
	Total credits in Semester IV	20

4. SCHEME OF EXAMINATION

- English shall be the medium of instruction and examination.
- Examinations shall be conducted at the end of each semester as per the academic calendar notified by the University.
- The scheme of evaluation shall be as follows: performance of the students will be evaluated based on a comprehensive system of continuous and end-semester evaluation. For each course, there shall be one minor test, assignments/ laboratory work, and an end-semester examination: (Mid-Term Exam, Assignments/practical & laboratory work - 30% weightage; End-semester examination - 70% weightage), except for practical courses where Internal assessment and end-semester examination shall carry 50 marks each. For each course, the duration of written end-emester examination shall be three hours. Evaluation of the Practical courses will be based on internal assessment and the end-semester evaluation by a board of examiners appointed by the Committee of Courses. Evaluation of the Practical courses will be based on internal assessment and the end-semester evaluation by a board of examiners to be appointed by the Committee of Courses.
- The students will choose the elective courses out of the list of courses which are offered in a semester. An elective course offered by another department/ center/ institute may be taken subject to approval of the department. The minor project will be carried out in the department. The major project may be carried out either in the department or in the industry under the supervision of a teacher(s) to be approved by the Department. In case the project is carried out in an organization, a supervisor may also be appointed from the organization. The projects will be evaluated by the internal supervisor and an external examiner to be appointed by the department on the recommendation of the internal supervisor. The minor and the major projects shall carry 100 and 500 marks respectively distributed as follows:

  (a) Mid-semester evaluation: 30% weightage

(b) End-semester evaluation
   (i) Dissertation: 30% weightage
   (ii) Viva-voce: 40% weightage

(i) Examination for courses shall be conducted only in the respective odd and even Semesters as per the Scheme of Examinations. Regular as well as Ex-Students shall be permitted to appear/re-appear/improve in courses of odd semesters only at the end of odd semesters and courses of even semesters only at the end of even semesters.

5. PASS PERCENTAGE

In order to pass a course and earn credits prescribed for it, a student must secure at least 40% marks in the end semester examinations and 40% marks in the internal assessment.
Minimum Credit Requirement for Degree: 80

6. PROMOTION CRITERIA

Part I to Part II

For promotion from part I to part II a student must pass in at least seven courses and acquire at least 28 credits out of the courses prescribed for part I examinations. A student who fails to get promoted to part II shall be required to seek fresh admission in part I as per the admission procedure/ University rules.

Eligibility for award of Degree

In order to be eligible for the award of the degree of M.Sc. Computer Science, a student must earn at least 80 credits out of the courses prescribed for parts I & II examinations taken together.

7. Eligibility and Mode of Admissions and Number of seats in the M. Sc.. programme:

- **To be decided by the University in every academic year.**

8. Conversion of Marks into Grades:

Letter Grade	**Numerical Grade**	**Formula**	**Computation of grade cut off**
O (outstanding)	10	$m \geq \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.5\sigma$ or 90% whichever is lower
A+ (Excellent)	9	$\overline{X} + 2.0\sigma \leq m < \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.0\sigma$ or 80% whichever is lower

A (Very Good)	8	$\overline{X} + 1.5\sigma \le m < \overline{X} + 2.0\sigma$	the value of $\overline{X} + 1.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.5\sigma$ or 70% whichever is lower
B+ (Good)	7	$\overline{X} + 1.0\sigma \le m < \overline{X} + 1.5\sigma$	the value of $\overline{X} + 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.0\sigma$ or 60% whichever is lower
B (Above Average)	6	$\overline{X} \le m < \overline{X} + 1.0\sigma$	the value of $\overline{X}$ to be taken into account for grade computation will be actual $\overline{X}$ or 50% whichever is lower
C (Average)	5	$\overline{X} - 0.5\sigma \le m < \overline{X}$	the value of $\overline{X} - 0.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 0.5\sigma$ or 45% whichever is lower
D (Pass)	4	$\overline{X} - 1.0\sigma \le m < \overline{X} - 0.5\sigma$	the value of $\overline{X} - 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 1.0\sigma$ or 40% whichever is lower
F (Fail)	0	$\overline{X} - 1.0\sigma > m$	

9. **CGPA to Percentage Conversion:**
The formula for calculating the final percentage of marks from Cumulative Grade Point Average (CGPA) will be as per the University rules.

10. DIVISION CRITERIA

The candidates eligible for the award of M.Sc. Computer Science degree shall be classified on the basis of the marks obtained in the aggregate of best 89 credits acquired during parts I & II examinations taken together, as follows:

i) I Division: 60% or more marks in the aggregate
ii) II Division: 50% or more marks but less than 60% marks in the aggregate.
iii) Pass: 40% or more marks but less than 50% marks in the aggregate.

11. SPAN PERIOD

No student shall be admitted as a candidate for the examination for any of the Parts/Semesters after the lapse of four years from the date of admission to the Part-I/Semester-I of the programme.

12. ATTENDANCE REQUIREMENTS

No student shall be considered to have pursued a regular course of study unless he/she is certified by the Head of the Department of Computer Science, University of Delhi, to have attended 66.67% of the total number of lectures, tutorials, practicals, and seminars conducted in each semester, during his/her course of study. Provided that he/she fulfils other conditions, the Head, Department of Computer Science, may permit a student to attend the next semester who falls short of the required percentage of attendance by not more than 10 percent of the lectures, tutorials, and seminars conducted during the semester.

13. COURSE CONTENT FOR EACH COURSE

PART - I (SEMESTER - I)

MCSC101: DESIGN AND ANALYSIS OF ALGORITHMS [3-0-1]

Course Objectives

This course is designed to introduce advanced techniques of designing and analyzing algorithms. The course also familiarizes the students with some problems that are too hard to admit fast solutions. Some of the advanced algorithm design techniques provide good solutions to these problems.

Course Learning Outcomes

Upon successful completion of this course, the student will be able to:

CO1: understand advanced techniques to design algorithms like augmentation, randomization, parallelization and use of linear programming.
CO2: Analyse the strengths and weaknesses of each technique.
CO3: Identify and apply technique(s) suitable for simple applications.
CO4: Demonstrate correctness of algorithms and analyse their time complexity theoretically as well as practically.
CO5: Analyze algorithms in the probabilistic framework.
CO6: Understand and apply string matching to application at hand.
CO7: Understand what are parallel algorithms, their utility, and the notion of speedup.
CO8: be able to appreciate that certain problems are too hard to admit fast solutions and be able to prove their hardness.
CO9: understand what are approximation algorithms, their utility, and the notion of approximation ratio.

Syllabus:

Review: Review of Basic Sorting and Searching Algorithms, Greedy Algorithms Divide & Conquer and Dynamic Programming.
Augmentation: Maximum Flow and Min Cut Problems, Matching in bipartite graphs, Minimum weight matching.
String Processing: Finite Automata method, KMP.
Randomized algorithms: Introduction to Random numbers, randomized Qsort, randomized selection, randomly built BST, randomized min-cut.
Parallel Algorithms: Shared Memory Model, Distributed Memory Model, Speedup. Searching, sorting, selection, matrix-vector multiplication, prefix-sum.
Linear Programming: Formulating an LP, Feasible region and Convex Polyhedron, Simplex Algorithm, LP-rounding to obtain integral solutions, Primal-Dual Algorithm.

Introduction to Complexity Classes: Classes P, NP - Verifiability, NP-Hard - Reducibility, NP Complete.
Introduction to Approximation Algorithms.

Readings

1. J. Kleinberg and E.Tardos, "Algorithm Design", 1st Edition 2013., Pearson Education India,
2. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, "Algorithms", 1st Edition, 2017, Tata McGraw Hill.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.
4. Vijay V. Vazirani, "Approximation Algorithms", 2013, Springer.
5. Bernhard Korte and Jens Vygen, "Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics)", 6th edition, 2018, Springer.
6. Rajeev Motvani and Prabhat Raghavan, 2004, Cambridge University Press.

MCSC102: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING [3-0-1]

Course Objectives: Beginning with a comprehensive overview of the AI techniques, the course introduces the supervised and unsupervised ML techniques, alongwith their applications in solving real-world problems. The course also covers evaluation and validation methods for ML models.

Course Learning Outcomes:

On successful completion of this course, the student will be able to:

CO1: discuss Turing Test, and various methods of knowledge representation as applicable to a given context.

CO2: design and implement supervised and unsupervised machine learning algorithms for real-world applications while understanding the strengths and weaknesses.

CO3: analyse the computational complexity of various machine learning algorithms.

CO4: fine tune machine learning algorithms and evaluate models generated from data.

Syllabus:

Unit-I Introduction to Artificial Intelligence: Evolution of AI as a discipline, Definitions and approaches, Subject matter of AI, Foundations of AI, Philosophical issues, AI for all, Ethical Issues and Responsible AI.

Unit-II Introduction to Machine Learning: Hypothesis and target class, bias-variance tradeoff, Occam's razor, Approximation and estimation errors, Curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-III Regression: Linear regression with one variable, Linear regression with multiple variables, Gradient Descent, Logistic Regression, Polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-IV Classification: Decision trees, Naive Bayes classifier, Perceptron, multilayer perceptron, Neural network, back-propagation Algorithm, Support Vector Machine, Kernel functions.

Unit V Evaluation: Performance evaluation metrics, ROC Curves, Validation methods, Bias-variance decomposition, Model complexity.

Unit-VI Unsupervised Learning: Clustering, distance metrics, Mixture models, Expectation Maximization, Cluster validation methods.

Readings:

1. Alpaydin, Ethem, **Introduction to machine learning,** MIT press, 2014.
2. T. M. Mitchell, Machine Learning, McGraw Hill Education, 2017.
3. Christopher, M. Bishop, **Pattern Recognition And Machine Learning**, Springer-Verlag, 2016.
4. Shai Shalev-Shwartz, Shai Ben-David, **Understanding Machine Learning: From Theory to Algorithms,** Cambridge Press, 2014.
5. Michalski, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. **Machine learning: An artificial intelligence approach**, Springer Science & Business Media, 2013.

MCSC103: INFORMATION SECURITY [3-0-1]

Course Objectives: The course aims to train the students to maintain the confidentiality, integrity and availability of data. The student learns various data encryption protocols for transmitting data over unsecured channels in a network.

Course Learning Outcomes:

CO1 To be able to describe various security issues.
CO2 To be able to implement a symmetric and asymmetric cryptographic methods.
CO3 To be able to describe the role and implementation of digital signatures.
CO4 To be able to describe security mechanisms like intrusion detection, auditing and logging.

Syllabus:

Overview of Security: Protection versus security; aspects of security– confidentiality, data integrity, availability, privacy; user authentication, access controls, Orange Book Standard.

Security Threats: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats- tapping and piracy.

Computer Security Models: BLP Model, BIBA Model, HRU Model.

Cryptography: Substitution, transposition ciphers, symmetric-key algorithms: Data Encryption Standard, Advanced Encryption Standard, IDEA, Block cipher Operation, Stream Ciphers: RC-4. Public key encryption: RSA, ElGamal. Diffie-Hellman key exchange. Elliptic Curve, EC cryptography, Message Authentication code (MAC), Cryptographic hash function.

Digital signatures: ElGamal digital signature scheme , Elliptic Curve digital signature scheme, NISTdigital signature scheme.

Key Management and Distribution : Symmetric Key Distribution, X.509 Certificate public key infrastructures.

Intrusion detection and prevention.

Readings:

1. W. Stalling, **Cryptography and Network Security Principles and Practices** (7^{th} ed.), Pearson of India, 2018.
2. A.J. Elbirt, Understanding and Applying Cryptography and Data Security, CRC Press, Taylor Francis Group, New York, 2015.
3. C. Pfleeger and SL Pfleeger, Jonathan Margulies, **Security in Computing** (5th ed.), Prentice-Hall of India, 2015
4. M. Merkow and J. Breithaupt, **Information Security: Principles and Practices**, Pearson Education, 2006.

MCSC104: MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE [3-0-1]

Course Objectives:
This course will discuss fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science. The objectives of this course comprise of providing students knowledge of logic and boolean circuits, sets, functions, relations, deterministic and randomized algorithms. Furthermore, the students will learn analysis techniques based on counting methods, recurrence relations, trees and graphs.

<u>Course Learning Outcomes :</u>

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

CO1: perform operations on vectors; represent vectors geometrically; apply vector algebra to solve problems in sub-disciplines of computer science.
CO2: perform operations on matrices and sparse matrices; compute the determinant, rank and eigenvalues of a matrix; apply matrix algebra to solve problems in sub-disciplines of computer science.
CO3: perform data analysis in probabilistic framework
CO4: visualise and model the given problem using mathematical concepts covered in the course

Syllabus:

Vectors: Definition of Vectors, Vector Addition, Dot and Cross Products, Span, Norm of vectors, Orthogonality, geometry of vectors, Application of vectors in document analysis

Matrix Algebra
Matrices as vectors; Matrix-vector ,vector-matrix and matrix-matrix multiplications; Inner and outer products, triangular matrix, diagonal matrix, systems of linear equations, linear independence, determinant, rank of matrix, Eigen values and Eigen vectors, matrix transformations, geometry of transformations, Applications of matrix algebra in image representation and transformations.

Basic Probability Theory
Sample Space and Events, Probability axioms, Conditional Probability, Bayes' law

Basic Statistics
Introduction to Descriptive and Inferential Statistics, Describing Data Sets as Frequency tables, Relative frequency tables and graphs, Scatter diagram, Grouped data, Histograms, Ogives; Percentiles, Box Plot, Coefficient of variation, Skewness, Kurtosis;,

Distributions: Continuous and Discrete random variables, probability density function, probability mass function, distribution function and their properties, mathematical expectation, conditional expectation, Uniform (continuous and discrete), Binomial, Poisson, Exponential, Normal, $\chi 2$ distributions, weak Law of Large Numbers, Central Limit Theorem, Chebyshev's inequality.

Stochastic Processes
Introduction to stochastic process, Markov Chain, Transition probabilities, Birth-Death process

Readings:

1. Kishor S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, John Wiely, 2016.

2. Sheldon M. Ross, Probability Models for Computer Science, Academic Press, 2001.

3. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, CRC Press 2012. https://cs.nyu.edu/davise/MathTechniques/index.html

4. From Algorithms to Z-Scores: Probabilistic and Statistical Modeling in Computer Science Norm Matloff, University of California, Davis (Creative Common Licence) http://heather.cs.ucdavis.edu/~matloff/132/PLN/probstatbook/ProbStatBook.pdf

<u>MCSC105:</u> DATA MINING [3-0-1]

<u>**Course Objectives:**</u> In this course, the objective is to introduce the KDD process. The course should enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization, supervised and unsupervised mining techniques.

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

CO1: distinguish between the process of knowledge discovery and Data Mining.
CO2: play with basic data exploration methods to develop understanding of given data
CO3: identify suitable pre-processing method for give problem.
CO4: describe different data mining tasks and algorithms.
CO5: use programming tools (e.g. Weka/Python/R etc) for solving data mining tasks.
CO6: follow formal notations and understand the mathematical concepts underlying data mining algorithms

<u>**Syllabus:**</u>

Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, and unsupervised learning techniques.

Data preprocessing : Data cleaning, Data transformation, Data reduction, Discretization

Classification: Supervised learning/mining tasks , Decision trees, Decision rules, Statistical (Bayesian) classification, Instance-based methods (nearest neighbor), Evaluation and Validation methods.

Clustering : Basic issues in clustering, Partitioning methods (k-means, expectation maximization), Hierarchical methods for clustering, Density-based methods, Cluster Validation methods and metrics

Association Rule Mining: Frequent item set, Maximal and Closed itemsets, Apriori property, Apriori algorithm.

<u>**Readings:**</u>

1. J Zaki Mohammed and Wagner Meira, **Data Mining and Analysis: Fundamental Concepts and Algorithms,** Cambridge University Press, 2014.
2. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2006.
3. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques** (3^{nd} ed.), Morgan Kaufmann, 2011.
4. Charu C Agrawal, Data Mining: The Textbook, Springer, 2015

MCSC106: SOFTWARE TOOLS AND TECHNIQUES [0-0-2]

Course Objective:

To develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: master the command line interface
CO2: use features of version control systems
CO3: debug and profile code
CO4: manage dependencies

Syllabus:

Shell Tools and Scripting, Editors (Vim), Data Wrangling, Command-line Environment, Version Control (Git), Debugging and Profiling, Metaprogramming: Working with Daemons, FUSE, Backups, APIs, Common command-line flags/patterns, Window managers, VPNs, Markdown, Booting + Live USBs, Docker, Vagrant, VMs, Cloud, OpenStack, Notebook programming

Readings:

1. Newham C. Learning the bash shell: **Unix shell programming**. " O'Reilly Media, Inc."; 2005 Mar 29.
2. Shotts W. **The Linux command line: a complete introduction**. No Starch Press; 2019 Mar 5.
3. https://git-scm.com/book/en/v2

PART - I (SEMESTER - II)

MCSC201: ARTIFICIAL NEURAL NETWORKS [3-0-1]

Course Objectives: The course covers state-of-the-art techniques in neural network design, optimization, and specialized architectures. Students will gain hands-on experience through practical assignments and projects, enabling them to apply advanced neural network models to real-world problems.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: implement and analyze kernel methods, radial-basis function networks, and kernel regression.

CO2: implement and evaluate regularization networks and self-organizing maps.

CO3: develop information-theoretic models for the machine learning tasks.

Syllabus:

Unit I Kernel Methods and Radial-Basis Function Networks: Cover's Theorem on the Separability of Pattern, The Interpolation Problem, Radial-Basis-Function Networks, Recursive Least-Squares Estimation of the Weight Vector, Hybrid Learning Procedure for RBF Networks, Interpretations of the Gaussian Hidden Units, Kernel Regression and Its Relation to RBF Networks

Unit II Regularization Theory: Hadamard's Conditions for Well-Posedness, Tikhonov's Regularization Theory, Regularization Networks, Generalized Radial-Basis-Function Networks, The Regularized Least-Squares Estimator, Estimation of the Regularization Parameter, Manifold Regularization, Differentiable Manifolds, Generalized Regularization Theory, Laplacian Regularized Least-Squares Algorithm

Unit III Self-Organizing Maps: Two Basic Feature-Mapping Models, Self-Organizing Map, Properties of the Feature Map, Contextual Maps, Hierarchical Vector Quantization, Kernel Self-Organizing Map, Relationship Between Kernel SOM and Kullback–Leibler Divergence.

Unit IV Information-Theoretic Learning Models: Entropy, Maximum-Entropy Principle, Mutual Information, Copulas, Mutual Information as an Objective Function to be Optimized, Maximum Mutual Information Principle, Infomax and Redundancy Reduction, Spatially Coherent Features, Spatially Incoherent Features, Independent-Components Analysis, Sparse Coding of Natural Images and Comparison with ICA Coding, Natural-Gradient Learning for Independent-Components Analysis, Maximum-Likelihood Estimation for Independent-Components Analysis, Maximum-Entropy Learning for Blind Source Separation, Maximization of Negentropy for Independent-Components Analysis, Coherent Independent-Components Analysis, Rate Distortion Theory and Information Bottleneck, Optimal Manifold Representation of Data**.**

Unit V Stochastic Methods Rooted in Statistical Mechanics: Statistical Mechanics, Markov Chains, Metropolis Algorithm, Simulated Annealing, Gibbs Sampling, Boltzmann Machine, Logistic Belief Nets, Deep Belief Nets, Deterministic Annealing, Analogy of Deterministic Annealing with Expectation-Maximization Algorithm

Readings:

1. Simon O. Haykin, **Neural Networks and Learning Machines,** Pearson Education, 3rd Edition, 2016
2. C. M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2010.

MCSC202: DEEP LEARNING [3-0-1]

Course Objectives: The student learns various state-of-the-art deep learning algorithms and their applications to solve real-world problems. The student develops skills to design neural network architectures and training procedures using various deep learning platforms and software libraries.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the feedforward and deep networks.

CO2:design single and multi-layer feed-forward deep networks and tune various hyper-parameters.

CO3: analyze the performance of deep networks.

Syllabus:

Unit-I Introduction: Historical context and motivation for deep learning; deep feedforward neural networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Unit-II Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Unit-III Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks.

Unit-IV Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Unit V: Generative Adversarial Networks (GANs): Introduction to Generative Adversarial Networks, GAN Architectures (DCGAN, CycleGAN), Applications of GANs (Image Generation, Style Transfer)

Unit VI: Large Language Models: Introduction to Natural Language Processing (NLP), Traditional NLP Techniques, Transformer Architecture, Pre-training and Fine-tuning Language Models, Ethical Considerations and Bias in Language Models, Applications of Large Language Models (Text Generation, Sentiment Analysis, Question Answering)

Unit-VII Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labelled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Readings:

1. Ian Goodfellow, **Deep Learning**, MIT Press, 2016.
2. Jeff Heaton, **Deep Learning and Neural Networks**, Heaton Research Inc, 2015.
3. Mindy L Hall, Deep Learning, VDM Verlag, 2011.
4. Li Deng (Author), Dong Yu, **Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing)**, Now Publishers Inc, 2009.

MCSC203: INTERNETWORKING WITH TCP/IP [3-0-1]

Course Objectives:

This course is oriented to provide students, an understanding of the communication process of the Internet. This course will enable students to test and troubleshoot IP-based communications systems, and also the architecture, design and behaviors of the Internet and of the TCP/IP suite of protocols. Furthermore, this course will discuss various Flow , Error and Congestion control mechanisms of TCP and the principles of IPv6 Addressing ,IPv6 and ICMPv6 Protocols.

Course Learning Outcomes :

On successful completion of the course, the student will be able to:

CO1: be able to explain the TCP/IP architecture and utility of different layers

CO2: Analyze IP addressing requirements, routing architecture and choose appropriate routing methods;

CO3: Understand the working of internetworking devices and their network configuration;

CO4: be able to determine and evaluate selection of applications and protocols for data communication

Syllabus

Unit-I: TCP/IP Architecture and IP Packet, IP Addressing, Subnetting, and Subnet Routing.

Unit-II: Classless Interdomain Routing (CIDR), ARP, Fragmentation and Reassembly, DHCP, NAT, IPv6

Unit-III: Transmission Control Protocol: UDP and TCP, TCP: Three-way Handshake, TCP Flow Control and Data Transfer, TCP Congestion Control, RTT-based Congestion Control for a Datacenter.

Unit-IV: Advanced Topics: Mobile IP, Multicast Routing, OpenFlow, SDN, and NFV, Network Security Threats

Readings:

1. Douglas E Comer, "Internetworking with TCP/IP Principles, Protocol, and Architecture" , Volume I, 6th Edition, Pearson Education, 2015.
2. Internetworking with TCP/IP Volume II: ANSI C Version: Design, Implementation, and Internals, Pearson Education India; 3rd edition, 2015.
3. William Stallings, "Data and Computer Communications", 9th Edition, Pearson Education, 2011

MCSC204: CLOUD COMPUTING [3-0-1]

Course Objectives: This course aims to provide students with a solid understanding of parallel and distributed computing and cloud computing. Students will learn about cloud computing's characteristics, benefits, and historical developments, including distributed systems, virtualization, and service-oriented computing. They'll also grasp cloud computing architecture, service models (IaaS, PaaS, SaaS), deployment models, and emerging paradigms like Edge Computing and Mobile Cloud Computing.

Course Learning Outcomes :

On completing this course, the student will be able to:

CO1: Understand cloud computing's characteristics, benefits, and historical developments, including distributed systems and virtualization.
CO2: Master cloud computing architecture, service models, deployment models, and practical application of cloud technologies.

CO3: Analyze cloud economics, address open challenges, and comprehend emerging paradigms like Edge Computing and Mobile Cloud Computing, applying theoretical knowledge to real-world scenarios effectively.

Syllabus:

Introduction to Parallel and Distributed Computing; Introduction to Cloud Computing; Characteristics and benefits of cloud computing; Historical developments and evolution of cloud computing: Distributed Systems, Virtualization, Web 2.0, Service-oriented computing.

Utility Computing; Cloud Computing Reference Model. Introduction to virtualization; Characteristics of virtualized environments; Taxonomy of virtualization techniques; Virtualization and cloud computing; Pros and cons of virtualization; Technology examples: Xen: paravirtualization, VMware: full virtualization, Microsoft Hyper-V.

Cloud Computing Architecture; Service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS); Deployment models: Public, Private, Hybrid, Community; IaaS: Introduction to IaaS, Resource Virtualization i.e. Server, Storage and Network virtualization; PaaS: Introduction to PaaS, Cloud platform & Management of Computation and Storage; SaaS: Introduction to SaaS, Cloud Services, Web services, Web 2.0, Web OS; Case studies related to IaaS, PaaS and SaaS.

Economics of the cloud; Open Challenges in Cloud Computing; Introduction to emerging computing paradigms and research challenges: Edge Computing, Mobile Cloud Computing, Fog Computing etc.; Introduction to IoT Cloud; Study on simulators related to cloud computing and emerging computing paradigms.

Readings:

1. R. Buyya, C. Vecchiola, S. ThamaraiSelvi, Mastering Cloud Computing, McGraw Hill Education.

2. B. Sosinsky, Cloud Computing Bible, Wiley.

3. K. Hwang, G. C. Fox, J.Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann

MCSC205 READING SKILLS [0-0-2]

Course Objectives: The course aims to develop an important skills of independent reading.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: Develop a habit of independent reading.

CO2: Given a requirement, independently select sources of reading.

CO3: Read and assimilate independently.

This is a self-study course. The students will carry out extensive reading on a topic to be assigned by the department.

MCSE201: DIGITAL IMAGE PROCESSING

Course Objectives: The objective of this course is to study the concept of digital image processing. The course should also cover the image enhancement in the spatial and frequency domain followed by the image morphological operations such as dilation, erosion, and hit-or-miss transformations. The course also covers image segmentation and image compression.

Course Learning Outcomes :

CO1 Explains theoretical and practical concepts of image acquisition, enhancement, compression and segmentation.
CO2 Introduces the concept of feature extraction of segmented images.
CO3 Provides an overview of various multimedia tools.

Syllabus:

Fundamental Steps in Image Processing: Element of visual perception, a simple image model, sampling and quantization, some basic relationships between pixel, image geometry in 2D, image enhancement in the spatial domain.

Introduction to spatial and frequency methods: Basic gray level transformations, histogram equalization, local enhancement, image subtraction, image averaging, basic spatial, filtering, smoothing spatial filters, sharpening spatial filters.

Introduction to the Fourier transformation: Discrete fourier transformation, fast Fourier transformation, filtering in the frequency domain, correspondence between filtering in the spatial and frequency domain smoothing frequency-domain filters, sharpening frequency-domain filters, homomorphic filtering,

Some basic morphological algorithms: Line detection, edge detection, gradient operator, edge linking and boundary detection, thresholding, region-oriented segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region.

Representation and Description:

Introduction to Image Compression: JPEG, MPEG, Wavelets

Readings

1. Rafael C. Gonzalez and Richard E.Woods, **Digital Image Processing**, Prentice–Hall of India, 2002

2. William K. Pratt, **Digital Image Processing: PIKS Inside** (3rd ed.), John Wiley & Sons, Inc., 2001

3. Bernd Jahne, **Digital Image Processing**, (5th revised and extended edition), Springer, 2002

4. S. Annadurai and R. Shanmugalakshmi, **Fundamentals of Digital Image Processing**, Pearson Education, 2007

5. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach**, Prentice-Hall of India, 2006

6. B. Chanda and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2007

MCSE202: COMPILER DESIGN

Course Objectives: The course aims to develop the ability to design, develop, and test a functional compiler/ interpreter for a subset of a popular programming language.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe how different phases of a compiler work.

CO2: implement top-down and bottom-up parsing algorithms.

CO3: use tools like Lex and Yacc to implement syntax-directed translation.

Syllabus:

Unit- I Lexical and Syntactic Analysis: Review of regular languages, design of a lexical analyzer generator, context-free grammars, syntactic analysis: top-down parsing: recursive descent and predictive parsing, LL(k) parsing; bottom-up parsing: LR parsing, handling ambiguous in bottom-up parsers.

Unit-II Syntax directed translation: Top-down and bottom-up approaches, data types, mixed mode expression; subscripted variables, sequencing statement, subroutines and functions: parameters calling, subroutines with side effects.

Unit-III Code generation, machine dependent and machine-independent optimization techniques.

Readings:

1. Alfred V. Aho, Ravi Sethi, D. Jeffrey Ullman, Monica S. Lam, **Principles, Techniques and Tools**, Pearson Education India, 2nd edition,, 2013.
2. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers,** Pearson, 2016.
3. Dick Grune, Kees van Reeuwijk, Henri E .Bal, Ceriel J.H. Jacobs, K Langendoen, **Modern Compiler Design,** Springer, 2012.

MCSE 203: NATURAL LANGUAGE PROCESSING [3-0-1]

Course Objectives: The course provides a rigorous introduction to the essential components of a Natural Language Processing (NLP) system. The students will learn various statistical, machine learning, and deep learning techniques in NLP and apply them to solve machine translation and conversation problems.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1:Understand and describe the evaluation of NLP systems.

CO2:Understand deep learning techniques in NLP and apply them to solve machine translation and conversation problems.

CO3:Learn about major NLP issues and identify possible future areas of NLP research.

Syllabus:

UNIT I Introduction: Natural Language Processing (NLP), History of NLP, Neural Networks for NLP, Applications - Sentiment Analysis, Spam Detection, Resume Mining, Conversation Modeling, Chat-bots, dialog agents, Question Processing

UNIT II Language Modeling and Part of Speech Tagging: Unigram Language Model, Bigram, Trigram, N-gram, Advanced smoothing for language modeling, Empirical Comparison of Smoothing Techniques, Applications of Language Modeling, Natural Language Generation, Parts of Speech Tagging, Morphology, Named Entity Recognition

UNIT III Words and Word Forms: Bag of words, skip-gram, Continuous Bag-Of-Words, Embedding representations for words Lexical Semantics, Word Sense Disambiguation, Knowledge Based and Supervised Word Sense Disambiguation

UNIT IV Text Analysis, Summarization and Extraction: Sentiment Mining, Text Classification, Text Summarization, Information Extraction, Named Entity Recognition, Relation Extraction, Question Answering in Multilingual Setting; NLP in Information Retrieval, Cross-Lingual IR

UNIT V Machine Translation: Need of MT, Problems of Machine Translation, MT Approaches, Direct Machine Translations, Rule-Based Machine Translation, Knowledge Based MT System, Statistical Machine Translation (SMT), Parameter learning in SMT (IBM models) using EM), Encoder-decoder architecture, Neural Machine Translation

Readings:

1. Speech and Language Processing. Dan Jurafsky and James H. Martin. Pearson (2009).
2. Introduction to Natural Language Processing. Jacob Eisenstein. The MIT Press (2019).
3. Neural Network Methods for Natural Language Processing. Yoav Goldberg. Morgan and Claypool Publisher (2017).
4. Deep Learning for Natural Language Processing: Develop Deep Learning Models for Natural Language in Python. Jason Brownlee. Machine Learning Mastery (2019).
5. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Steven Bird, Ewan Klein and Edward Loper. O'Reilly (2009).

PART - II (SEMESTER – III)

MCSC301: MINOR PROJECT [0-0-4]

MCSE301: CYBER PHYSICAL SYSTEMS [3-0-1]

Course Objectives:

Cyber-physical systems (CPS) have a utility in many safety-critical areas such as automotive, avionics, trains, healthcare, atomic energy, power, and industrial automation. CPS are composed of integrated physical systems that are either controlled by software or are strongly integrated. The objectives of this course are to introduce students to the modelling of CPS, and to develop the ability to analyze and simulate different CPS systems. The student will also learn to develop skills to help them plan, implement, and monitor cyber security mechanisms to protect information technology assets.

Course Learning Outcomes (CO):

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

CO1: use the modeling software and related tools for the hybrid system.

CO2: to use comprehensive models of physical and cyber components to examine CPS.

CO3: to take up research work in multi-disciplinary areas keeping in mind the environment safety concerns

CO4: state the need and scope for cyber laws.

CO5: enumerate various network attacks, and describe their sources and mechanisms of prevention

Syllabus:

Unit I: Introduction and examples of cyber physical systems (CPS) in different domains, Important design aspects and quality attributes of CPS, Finite state machine, Characteristics of high confidence CPS, Discrete System Modelling, Continuous systems modelling, Extended state machines, Modelling of Hybrid systems, Various classes of Hybrid Systems, Analysis and Verification, Concepts of embedded systems, Input-outputs, Invariants and Temporal Logic, Linear Temporal Logic, Refinement and Equivalence, Model Development, Rechability Analysis and Model Checking

UNIT II: Cyberspace, Internet of things, Cyber Crimes, Cyber Security, Cyber Security Threats, Cyber laws and legislation, Law Enforcement Roles and Responses. Network Threat Vectors, MITM, OWAPS, ARP Spoofing, IP & MAC Spoofing, DNS Attacks, SYN Flooding attacks, UDP ping-pong and Fraggle attacks, TCP port scanning and reflection attacks, DoS, DDOS. Network Penetration Testing Threat assessment, Penetration testing tools, Penetration testing, Vulnerability Analysis, Threat matrices, Firewall and IDS/IPS, Wireless networks, Wireless Fidelity (Wi-Fi), Wireless network security protocols, Nmap, Network fingerprinting, BackTrack, Metasploit.

Readings:

1. R. Rajkumar, D. de. Niz and M. Klein, **Cyber Physical Systems,** Addision-Wesely, 2017

2. Rajiv Alur, **Principles of Cyber-Physical Systems**, MIT Press, 2015.

3. E.A.Lee and S A Shesia, **Embedded system Design: A Cyber-Physical Approach,** Second Edition, MIT Press, 2018

4. A. Platzer, **Logical Foundations of Cyber Physical Systems,** Springer, 2017.

5. Peter W. Singer and Allan Friedman, Cybersecurity and Cyberwar, Oxford University Press, 2014

6. Jonathan Clough, Principles of Cybercrime, Cambridge University Press, 24-Sep-2015

MCSE302: GRAPH THEORY

Course Objectives: This course will thoroughly introduce the basic concepts of graphs theory, graph properties and formulations of typical graph problems. The student will learn to model real-life problems such as graph coloring and connectivity as graph problems

Course Learning Outcomes :
On successful completion of the course, students will be able to:
CO1: model problems in different types of basic graphs like trees, bipartite graphs and planar graphs.
CO2: identify special graphs like Euler graphs and Hamiltonian graphs.
CO3: identify various forms of connectedness in a graph
CO4: examine different graph-coloring problems and their solutions.
CO5: model simple problems from real life as graph-coloring problems.

Syllabus:

Fundamental Concepts: Definitions, examples of problems in graph theory, adjacency and incidence matrices, isomorphisms, paths, walks, cycles, components, cut-edges, cut-vertices, bipartite graphs, eulerian graphs, vertex degrees, reconstruction conjecture, extremal problems, degree sequences, directed graphs, de Bruijn cycles, Orientations and tournaments.

Trees: Trees and forests, characterizations of trees, spanning trees, radius and diameter, enumeration of trees, Cayley's formula, Prüfer code, counting spanning trees, deletion-contraction, the matrix tree theorem, graceful labelling, minimum spanning trees (Kruskal's algorithm), shortest paths (Dijkstra's algorithm).

Matching and Covers: Matchings, maximal and maximum matchings, M-augmenting paths, Hall's theorem and consequences, Min-max theorems, maximum matchings and vertex covers, independent sets and edge covers,Connectivity, vertex cuts, Edge-connectivity.

Connectivity and Paths: Blocks, k-connected graphs, Menger's theorem, line graphs, network flow problems, flows and source/sink cuts, Ford-Fulkerson algorithm, Max-flow min-cut theorem.

Graph Coloring: Vertex colorings, bounds on chromatic numbers, Chromatic numbers of graphs constructed from smaller graphs, chromatic polynomials, properties of the chromatic polynomial, the deletion-contraction recurrence.

Planar Graphs: Planar graphs, Euler's formula, Kuratowski's theorem, five and four color theorems.

Readings:

1. Douglas B West, "Introduction to Graph Theory", II Edition, 2017, Pearson.
2. Gary Chartrand and Ping Zhang "Introduction to Graph Theory", 2017, Tata McGraw Hill.
3. Jonathan L. Gross and Jay Yellen, "Graph Theory and Its Applications", 2nd Edition, 2005, Chapman Hall (CRC).
4. The course will also be taught through various research Courses.

MCSE303: NETWORK SCIENCE

Course Objectives: The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn dynamics of and on networks in the context of applications from disciplines like biology, sociology, and economics

Course Learning Outcomes :

At the end of the course, the student will be

CO1: able to appreciate ubiquity of graph data model

CO2: able to understand the importance of graph theoretic concepts in social network analysis

CO3: able to understand the structural features of a network

CO4: familiar with the theoretical graph generation models

CO5: identify community structures in networks

CO6: able to write programs to solve complex network problems

Syllabus:

Introduction: Introduction to complex systems and networks, modelling of complex systems, review of graph theory.

Network properties: Clustering coefficient, centrality measures for directed and undirected networks.

Graph models: Random graph model, Small world graph model, Network evolution using preferential attachment

Community structure in networks: Communities and community detection in networks, Hierarchical algorithms for community detection, Modularity based community detection algorithms, Label Propagation algorithm

Readings:

1. Mohammed J. Zaki, Wagner Meira Jr.; Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014
2. Albert Barabasi, Network Science , Cambridge University Press, 2016
3. M.E. J. Newman, Networks: An Introduction, , Oxford University Press, 2010.
4. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010

MCSE 304: INFORMATION RETRIEVAL [3-0-1]

Course Objectives: This course aims to equip the students with basic techniques for information retrieval that find use in text analytics. The student will also learn to apply the tools for information extraction.

Course Learning Outcomes:

On completion of the course, the student will be able to

CO1: describe early developments in IR.

CO2: apply measures for evaluating retrieved information.

CO3: choose appropriate model for document processing.

CO4: develop simple information retrieval tools to solve real world problems.

Syllabus:

Unit 1- Introduction: Information, Information Need and Relevance; The IR System; Early developments in IR, User Interfaces.

Unit 2- Retrieval and IR Models: Boolean Retrieval; Term Vocabulary and Postings list; Index Construction; Ranked and other alternative Retrieval Models.

Unit 3- Retrieval Evaluation: Notion of Precision and Recall; Precision-Recall Curve, Standard Performance Measures such as MAP, Reciprocal ranks, F-measure, NDCG, Rank Correlation.

Unit 4- Document Processing: Representation; Vector Space Model; Feature Selection; Stop Words; Stemming; Notion of Document Similarity; Standard Datasets..

Unit 5- Classification and Clustering: Notion of Supervised and Unsupervised Algorithms; Naive Bayes, Nearest Neighbour and Rochio's algorithms for Text Classification; Clustering Methods such as K-Means.

Unit-6: Link Analysis: Page Rank, HITs, Web Crawling. Applications.

Readings:

1. R. Baeza-Yaets, B. Ribeiro-Neto, **Modern Information Retrieval: The Concept and Technology behind Search**, Latest Edition, Addison-Wesley, 1999.

2. C. D. Manning, P. Raghvan, H. Schutze, **Introduction to Information Retrieval,** Cambridge University Press, 2008.

3. D. A. Grossman, O. Frieder, **Information Retrieval: Algorithms and Heuristics,** 2nd Ed., Springer, 2004.

4. S. Buettcher, Charles L.A. Clarke, G. V. Carmack, **Information Retrieval: Implementing and Evaluating Search Engines,** MIT Press.

5. B. Croft, D. Metzler, T. Strohman, **Search Engines: Information Retrieval in Practice,** Addison Wesley

MCSE306: SOFT COMPUTING [3-0-1]

Course Objectives:

This course provides insights of soft computing frameworks applicable to bring its precision solutions for wide range of complex scientific applications.

Course Leaning Outcomes:

CO1: applying soft computing techniques towards various real-time case studies.

CO2: idea to design hybrid soft techniques over conventional computing methods.

CO3: Identify and select suitable Soft Computing methods to solve scientific complex problems where standard computing procedures are in intractable forms.

Syllabus:

UNIT-I Soft Computing: Introduction of Soft Computing, Soft Computing vs. Hard Computing, Various Types of Soft Computing Techniques, Applications of Soft Computing, Predicate Calculus, Rules of Interference, Overview of neural networks, estimating regularization parameter Kohnen's self-organizing networks, Hopfield network, applications of neural networks.

UNIT-II Fuzzy Logic Computing: Introduction of fuzzy sets and fuzzy reasoning, Basic functions on fuzzy sets, relations, rule based models and linguistic variables, fuzzy controls, Fuzzy decision making, , inferencing, defuzzification, fuzzy clustering, fuzzy rule based classifier, applications of fuzzy logics.

UNIT-III Evolutionary Algorithms: Introduction to evolutionary algorithms, Basic principles of Evolutionary Algorithms, Evolutionary strategies, Genetic Algorithm, Fitness Computations, Cross Over, Mutation, Evolutionary Programming, Classifier Systems, Genetic Programming Parse Trees, Variants of GA, Applications, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization, concept of multi-objective optimization problems (MOOPs), Multi-Objective Evolutionary Algorithm (MOEA), Non-Pareto approaches to solve MOOPs, Pareto-based approaches to solve MOOPs, Some applications with MOEAs.

Readings:

1. Simon S. Haykin, Neural Networks, Prentice Hall, 2nd edition.
2. B. Yegnanrayana, “Artificial Neural Networks” , PHI.
3. Jacek M. Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House,1994
4. Zimmermann, “Fuzzy Set Theory and its Application”, 3rd Edition.
5. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall, 1998.
6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, 1997.
7. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

MCSE307: QUANTUM COMPUTING AND ITS APPLICATIONS [3-0-1]

Course Objectives: This course provides a foundation for quantum computing, Post-Quantum Cryptography and quantum machine learning. It covers the fundamental concepts of quantum mechanics, quantum algorithms, and their applications in various areas, including cryptography, cybersecurity, machine learning, finance, the energy sector, etc. Students will gain a theoretical understanding of quantum computing and practical skills in implementing quantum algorithms for various tasks.

Course Learning Outcomes: On completing this course, the student will be able to:

CO1: Understand the basic principles of quantum mechanics and their relevance to quantum computing.
CO2: Comprehend quantum algorithms and their applications.
CO3: Apply quantum optimization techniques in problem-solving.
CO4: Demonstrate practical skills in quantum computing in various areas, including cryptography and machine learning.

Syllabus:

Unit-I Fundamentals of Quantum Computing: Mathematical foundations: Vectors, Vector space, Inner product; Qubits, Introduction to quantum mechanics and its relevance to Quantum gates, superposition principle, and entanglement Quantum parallelism and interference, No cloning theorem, quantum teleportation.

Unit-II Post-Quantum Security: Deutsch-Jozsa algorithm, Simon's algorithm, Bernstein-Vazirani, RSA algorithm and factorization attack on RSA, Shor's algorithm for integer factorization, Grover's algorithm for unstructured search, Hash preimage attack with Grover’s algorithm, Quantum Fourier transform and its applications, Harrow–Hassidim–Lloyd (HHL) algorithm, Quantum attack resistant Digital Signatures.

Unit-III Quantum Machine Learning and Optimization: Quantum machine learning (QML) models – QSVM, QNN, QCNN, Quantum Linear Regression, Variational Quantum Classifier (VQC), Quantum k-means clustering; kernel methods, Quantum Boltzmann Machines; Quantum optimization techniques: QAOA, quantum annealing.

Unit-IV: Introduction to quantum simulation tools and platforms: Google CIRQ, Amazon Braket, IBM Qiskit, Pennylane, Q#, Tensorflow quantum, Tket/pyket, XACC, Project Q, Quantum Development Kit (QDK).

Readings:

1. Elias F. Combarro, Samuel González-Castillo, and Alberto Di Meglio. A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms. Packt Publishing Ltd, 2023.
2. Noson S. Yanofsky and Mirco A. Mannucci. Quantum computing for computer scientists. Cambridge University Press, 2008.
3. Douglas R. Stinson and Maura B. Paterson. Cryptography, Theory and Practice, CRC Press, 2019.
4. Santanu Pattanayak. Quantum Machine Learning with Python: Using Cirq from Google Research and IBM Qiskit. Apress, 2021.
5. Santanu Ganguly. Quantum Machine Learning: An Applied Approach. Apress, 2021.
6. https://docs.quantum.ibm.com/
7. https://quantumai.google/cirq/experiments/textbook_algorithms

MCSE308: SOFTWARE QUALITY ASSURANCE AND TESTING [3-0-1]

Course Objectives:

Course Learning Outcomes : On completion of this course, the student will be able to:
CO1: understand quality management processes.
CO2: understand the importance of standards in the quality management process and role of SQA function in an organization.
CO3: gain knowledge of statistical methods and process for software quality assurance
CO4: understand the need and purpose of software testing. CO5: model the quantitative quality evaluation of the software products.

Syllabus :
Unit-I Introduction: Concept of Software quality, product and process quality, software quality metrics, quality control and total quality management, quality tools and techniques, quality standards, defect management for quality and improvement.
Unit-II Designing software quality assurance system: Statistical methods in quality assurance, fundamentals of statistical process control, process capability, Six-sigma quality.
Unit-III Testing: Test strategies, test planning, functional testing, stability testing and debugging techniques.
Unit-IV Reliability: Basic concepts, reliability measurements, predictions and management.
Readings:
1. N.S. Godbole, Software Quality Assurance: Principles and Practice for the New Paradigm (2nd Ed.), Narosa Publishing, 2017.

2.G. Gordon Schulmeyer (4th eds.), Handbook of Software Quality Assurance Artech House,Inc, 2008.
3. G. O'Regan, A Practical Approach to Software Quality, Springer Verlag, 2002.
4. Daniel Galin, Quality Assurance: From theory to implementation, Pearson Education Ltd., 2004
5. S.H. Kan, Metrics and Models in Software Quality Engineering (2nd ed.), Pearson Education Inc., 2003.
6. J.D. McGregor and D.A. Sykes, A Practical Guide to Testing, Addison-Wesley, 2001.
7. Glenford J. Myers, The Art of Software Testing (2nd ed.), John Wiley, 2004.
8. D. Graham, E.V. Veenendaal, I. Evans and R. Black, Foundations of Software Testing, Thomson Learning, 2007.

MCAE310 Social Networks

Course Objectives: The course aims to equip students with various SNA approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes: On completing this course, the student will be able to:

CO1: Explain the basic concepts and principles of social network.

CO2: Identify different types of social networks and their characteristics.

CO3: Implement and apply various social network analysis techniques, such as, influence maximization, community detection, link prediction, and information diffusion.

CO4: Apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

Unit-I: Introduction to Social Network Analysis: Introduction to Social Network Analysis, Types of Networks, Nodes Edges, Node Centrality, betweenness, closeness, eigenvector centrality, network centralization, Assortativity, Transitivity, Reciprocity, Similarity, Degeneracy and Network Measure, Networks Structures, Network Visualization, Tie Strength, Trust, Understanding Structure Through User Attributes and Behavior.

Unit-II: Link Analysis and Link Prediction: Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis and Algorithms, Page Rank, Personalized PageRank, DivRank, SimRank, PathSim. Temporal Changes in a Network, Evaluation Link Prediction Algorithms, Heuristic Models,Probabilistic Models, Applications of Link Prediction.

Unit-III: Community Detection: Applications of Community Detection, Types of Communities, Community Detection Algorithms, Disjoint Community Detection, Overlapping Community Detection, Local Community Detection, Evaluation of Community Detection Algorithms.

Unit-IV: Influence Maximization: Applications of Influence Maximization, Diffusion Models, Independent Cascade Model, Linear Threshold Model, Triggering Model, Time-Aware Diffusion Model, Non-Progressive Diffusion Model. Influence

Maximization Algorithms, Simulation-Based Algorithms, Proxy-Based Algorithms, Sketch-Based Algorithms, Community-Based Influence Maximization, and Context-Aware Influence Maximization.

Unit-V: Multilayer Social Network: Multilayer Social Networks, Formation of Multilayer Social Networks, Heuristic-based Approaches, Greedy Approaches, Centrality-based Approaches, Meta-heuristic Approaches, Path-based Approaches, Measuring Multilayer Social Networks.

Readings:

1. Tanmoy Chakraborty, Social Network Analysis, Wiley India, 2021.
2. David Knoke and Song Yang. Social network analysis. SAGE publications, 2019.
3. Mark E. Dickison, Matteo Magnani and Luca Rossi, Multilayer social networks, Cambridge University Press, 2016.
4. Jennifer Golbeck, Analyzing the social web, Morgan Kaufmann, 2013.
5. Stanley Wasserman, and Katherine Faust. Social network analysis: Methods and applications, Cambridge University Press, 2012.
6. M.E.J. Newman, Networks: An introduction. Oxford University Press, 2010.
7. Wei Chen, Carlos Castillo and Laks V.S. Lakshmanan, Information and influence propagation in social networks. Springer Nature, 2014
8. Virinchi Srinivas and Pabitra Mitra, Link prediction in social networks: role of power law distribution. New York: Springer International Publishing, 2016

MCSO301: DATA ANALYSIS AND VISUALIZATION [3-0-1]

Course Objectives: The course develops students' competence in cleaning and analyzing data related to a chosen application. It also aims to develop skills in using various tools for data visualization and choosing the right tool for given data.

Course Learning Outcomes:
On completing the course, the students will be able to :
CO1: use data analysis tools with ease.
CO2: load, clean, transform, merge, and reshape data.
CO3: create informative visualisations and summarise data sets.
CO4: analyse and manipulate time series data.
CO5: solve real world data analysis problems.

Syllabus
Unit 1 Introduction: Introduction to Data Science, Exploratory Data Analysis and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython and Jupyter Notebook. Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels
Unit 2 Getting Started with Pandas: Arrays and vectorized computation, Introductio to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics. Data Loading, Storage and File Formats. Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting with Web APIs, Interacting with Databases Data Cleaning and Preparation. Handling Missing Data, Data Transformation, String Manipulation
Unit 3 Data Wrangling: Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting. Data Visualization matplotlib: Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools
Unit 4 Data Aggregation and Group operations: Data grouping, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

Unit 5 Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.

Readings

1. McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media.
2. O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline, O'Reilly Media.

MCSO302: DATA SCIENCE [3-0-1]

Course Objectives: The objective of this course is to analyze the data statistically and discover valuable insights from it. The course gives hands-on practice on predictive and descriptive modeling of the preprocessed data. In addition, the student also learns to apply mining association rules from the transactional data and mining text from the document will also be covered during the course.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: demonstrate proficiency with statistical analysis of data.

CO2: develop the ability to build and assess data-based models.

CO3: execute statistical analyses and interpret outcomes.

CO4: apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively.

Syllabus:

Unit-I Introduction: Introduction data acquisition, data preprocessing techniques including data cleaning, selection, integration, transformation, and reduction, data mining, interpretation.

Unit-II Statistical data modeling: Review of basic probability theory and distributions, correlation coefficient, linear regression, statistical inference, exploratory data analysis, and visualization.

Unit-III Predictive modeling: Introduction to predictive modeling, decision tree, nearest neighbor classifier, and naïve Bayes classifier, classification performance evaluation, and model selection.

Unit-IV Descriptive Modeling: Introduction to clustering, partitional, hierarchical, and density based clustering (k-means, agglomerative, and DBSCAN), outlier detection, clustering performance evaluation.

Unit-V Association Rule Mining: Introduction to frequent pattern mining and association rule mining, Apriori algorithm, measures for evaluating the association patterns.

Unit-VI Text Mining: Introduction of the vector space model for document representation, term frequency-inverse document frequency (tf-idf) approach for term weighting, proximity measures for document comparison, document clustering, and text classification.

Readings:

1. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython, 2nd Ed., O'Reilly, 2017.
2. P. Tan, M. Steinbach, A Karpatne, and V. Kumar, Introduction to Data Mining, 2nd Edition, Pearson Education, 2018.
3. G. Grolemund, H. Wickham, R for Data Science, 1st Ed., O'Reilly, 2017.