

B. Tech. – M. Tech. (Computer Science and Engineering)

Program Learning Objectives:	Program Learning Outcomes (PLO):
Program Goal 1: Comprehensive Knowledge Acquisition: To provide a strong foundation in both fundamental and advanced concepts of Computer Science and Engineering (CSE), integrating bachelor and master-level education.	Program Learning Outcome 1: PLO-1: Students will acquire and demonstrate a comprehensive understanding of core concepts in computing principles, data structure, algorithms, programming languages. Program Learning Outcome 2: PLO-2: Students will be able to understand problems computationally, design efficient algorithms, and implement software solutions.
Program Goal 2: Training for Research and Industry: To provide quality training for conducting fundamental and advanced research in Computer Science and Engineering and software development.	Program Learning Outcome 3: PLO-3: Students will develop the ability to apply the scientific method to computer science problems, including formulating hypotheses, designing experiments, and analyzing results. Program Learning Outcome 4: PLO-4: Students will demonstrate proficiency in software development, including the use of modern programming environments, operating systems, computer networks, version control, and collaborative development practices.
Program Goal 3: Technical Expertise and Professional Skills: To focus on skill enhancement in system development and security.	Program Learning Outcome 5: PLO-5: Students will be able to design, implement, and manage complex systems, computer architecture, networking, ensuring quality, and security.
Program Goal 4: Communication Skill: To develop various communication skills such as reading, listening, speaking, etc. This will help in expressing ideas and views clearly and effectively.	Program Learning Outcome 6: PLO-6: Students will develop the ability to communicate technical information effectively through written reports, oral presentations, and collaborative projects.
Program Goal 5: Social Awareness: To understand societal issues related to computer science and allied areas and develop methods and means to abate and create awareness in society.	Program Learning Outcome 7: PLO-7: Students will develop an awareness of ethical, social, and environmental issues related to computing, applying responsible practices in their professional activities. Program Learning Outcome 8: PLO-8: Students will learn to work effectively in teams, demonstrating leadership, collaboration, and project management skills.

Sl. No.	Subject Code	SEMESTER VII	L	T	P	C
1.	CS41PQ	B. Tech. Elective – II	3	0	0	3
2.	CS41PQ	B. Tech. Elective – III	3	0	0	3
3.	XX41PQ	IDE-III	3	0	0	3
4.	HS41XX	HSS Elective - II	3	0	0	3
5.	CS4198	Summer Internship/ Summer Project*	0	0	12	3
6.	CS4199	Project - I	0	0	12	6
7.	CS5101	Design and Analysis of Algorithms	3	1	0	4
TOTAL			15	1	24	25

***Note:**

For specific cases of internship after VIth Semester, the performance evaluation would be made on joining the VIIth Semester and graded accordingly in the VIIth Semester:

Note:

a) (i) Summer internship (*) period of at least 60 days' (8 weeks) duration begins in the intervening vacation between Semester VI and VII that may be done in industry / R&D / Academic Institutions including IIT Patna. The evaluation would comprise **combined grading based on host supervisor evaluation, project internship report after plagiarism check and seminar presentation at the Department (DAPC to coordinate)** with equal weightage of each of the three components stated herein.

a) (ii) Further, on return from internship, students will be evaluated for internship work through combined grading based on host supervisor evaluation, project internship report after plagiarism check, and presentation evaluation by the parent department with equal weightage of each component.

b) (i) In the VIIth semester, students can opt for a semester long internship on recommendation of the DAPC and approval of the Competent Authority.

b) (ii) On approval of semester long internship, at the maximum two courses (properly mapped/aligned syllabus) at par with institute electives may be opted from NPTEL and / or SWAYAM and the other two more should be done at the institute through course overloading in any other semester (either before or after the internship) and/or during following summer semester.

b) (iii) The candidates opting two courses from NPTEL and / or SWAYAM would be required to appear in the examination at the Institute as scheduled in the Academic Calendar.

SEMESTER VIII						
Sl. No.	Subject Code	Subject Name	L	T	P	C
1.	CS5201	Advanced Artificial Intelligence	3	0	0	3
2.	CS5202	Theoretical Computer Science	3	0	0	3
3.	CS62XX	M. Tech. Elective – I	3	0	0	3
4.	RM6201	Research Methodology	3	1	0	4
5.	CS4299	Project II	0	0	12	6
TOTAL			12	1	12	19

SEMESTER IX						
Sl. No.	Subject Code	Subject Name	L	T	P	C
1.	CS61XX	M. Tech. Elective – II	3	0	0	3
2.	CS61XX	M. Tech. Elective – III	3	0	0	3
3.	CS61XX	M. Tech. Elective – IV	3	0	0	3
4.	CS6190	Project – III**	0	0	16	8
TOTAL			9	0	16	17

SEMESTER X						
Sl. No.	Subject Code	Subject Name	L	T	P	C
1.	CS6290	Project – IV**	0	0	36	18
TOTAL			0	0	36	18

****Note: M. Tech. Project outside the Institute**

In the IXth Semester, students can opt for a semester long M. Tech. project subject to confirmation from an Institution of repute for research project, on the assigned topic at any external Institution (Industry / R&D lab / Academic Institutions) based on recommendation of the DAPC provided:

(i.) The project topic is well defined in objective, methodology and expected outcome through an abstract and statement of the student pertaining to expertise with the proposed supervisor of the host institution and consent of the faculty member from the concerned department at IIT Patna as joint supervisor.

(ii.) The consent of both the supervisors (external and institutional) on project topic is obtained a priori and forwarded to the academic section through DAPC for approval by the competent authority for office record in the personal file of the candidate.

(iii.) Confidentiality and Non-Disclosure Agreement (NDA) between the two organizations with clarity on intellectual property rights (IPR) must be executed prior to initiating the semester long project assignment and committing the same to external organization and vice versa.

(iv.) The evaluation in each semester at Institute would be mandatory and the report from Industry Supervisor will be given due weightage as defined in the Academic Regulation. Further, the final assessment of the project work on completion will be done with equal weightage for assessment of the host and Institute supervisors, project report after **plagiarism check**. The award of grade would comprise **combined assessment based on host supervisor evaluation, project report quality and seminar presentation at the Department (DAPC to coordinate)** with equal weightage of each of the components stated herein.

(v.) In case of poor progress of work and / or no contribution from external supervisor, the student need to revert back to the Institute essentially to fulfill the completion of M. Tech. project as envisaged at the time of project allotment. However, the recommendation of DAPC based on progress report and presentation would be mandatory for a final decision by the competent authority.

Grand Total (Semester I to X) : 208

ELECTIVE GROUPS

B. Tech. Elective - II						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS4121	Pattern Recognition	3	0	0	3
2.	CS4122	Principles of Programming Languages	3	0	0	3
3.	CS4123	Social Networks	3	0	0	3
4.	CS4124	Multimedia System	3	0	0	3
5.	CS4125	Program Analysis and Verification	3	0	0	3

B. Tech. Elective - III						
1.	CS4117	Time Series Analysis	3	0	0	3
2.	CS4118	Computational Data Analysis	3	0	0	3
3.	CS4126	Graph Machine Learning	3	0	0	3
4.	CS4127	Bioinformatics	3	0	0	3
5.	CS4128	Graph Theory	3	0	0	3
6.	CS4129	Blockchain Technology	3	0	0	3

M. Tech. Elective - I						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6201	Artificial Internet of Things	3	0	0	3
2.	CS6202	Game Theory	3	0	0	3
3.	CS6203	Text Mining & Analytics	3	0	0	3
4.	CS6204	Knowledge Distillation	3	0	0	3
5.	CS6206	Selected Topics in Wireless Networks	3	0	0	3
6.	CS6207	Advanced Big Data Analytics	3	0	0	3
7.	CS6208	Quantum Machine Learning	3	0	0	3
8.	CS6210	Selective Topics in Generative AI	3	0	0	3
9.	CS6215	Quantum Cyber Security	3	0	0	3
10.	CS6211	Selected Topics in Cryptography	3	0	0	3
11.	CS6216	High Performance Computing	3	0	0	3

M. Tech. Elective – II						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6101	Advanced Blockchain Technology	3	0	0	3
2.	CS6102	Advanced Cyber Security	3	0	0	3
3.	CS6104	Formal Methods in Program Analysis and Verification	3	0	0	3
4.	CS6109	Reinforcement Learning	3	0	0	3

M. Tech. Elective - III						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6106	Advanced Cloud Computing	3	0	0	3
2.	CS6107	Advanced Edge Computing	3	0	0	3
3.	CS6113	Cyber Physical Systems	3	0	0	3

M. Tech. Elective - IV						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6103	Advanced Pattern Recognition	3	0	0	3
2.	CS6108	Advanced Computational Data Analysis	3	0	0	3
3.	CS6110	Advanced Graph Machine Learning	3	0	0	3
4.	CS6111	Advanced Time Series Analysis	3	0	0	3

Sl. No.	Subject Code	SEMESTER VII	L	T	P	C
1.	CS41PQ	B. Tech. Elective – II	3	0	0	3
2.	CS41PQ	B. Tech. Elective – III	3	0	0	3
3.	XX41PQ	IDE-III	3	0	0	3
4.	HS41XX	HSS Elective - II	3	0	0	3
5.	CS4198	Summer Internship/ Summer Project*	0	0	12	3
6.	CS4199	Project - I	0	0	12	6
7.	CS5101	Design and Analysis of Algorithms	3	1	0	4
TOTAL			15	1	24	25

B. Tech. Elective - II						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS4121	Pattern Recognition	3	0	0	3
2.	CS4122	Principles of Programming Languages	3	0	0	3
3.	CS4123	Social Networks	3	0	0	3
4.	CS4124	Multimedia System	3	0	0	3
5.	CS4125	Program Analysis and Verification	3	0	0	3

Course Number	CS4121
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Pattern Recognition
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the fundamental principles and techniques of pattern recognition, including classification and clustering methods. (b) To develop basic problem-solving skills by implementing the basic pattern recognition algorithms. (c) To gain proficiency in feature extraction, selection, and dimensionality reduction to enhance pattern recognition performance. (d) Apply pattern recognition algorithms to practical applications in image processing, speech recognition, and data mining.
Course Description	This course on pattern recognition aims to equip students with the theoretical foundations and practical skills necessary to identify and analyze patterns in data. By focusing on essential principles, students will develop the ability to implement and evaluate various pattern recognition algorithms. Students will enhance their understanding of machine learning, statistical methods, and data preprocessing techniques through interactive lectures, exercises, and projects. Upon completion, students will be proficient in designing and applying pattern recognition systems for applications such as image processing, speech recognition, and data mining, thereby enhancing their analytical and problem-solving capabilities in diverse domains.
Course Outline	Introduction to pattern recognition, key concepts, learning types, approaches, decision boundaries, and distance metrics. Pattern extraction and preprocessing, pattern classification and algorithms Different paradigms and representations for pattern clustering techniques and validation. Feature extraction and selection methods, problem statements, and relevant algorithms (branch and bound, sequential selection). Recent advances in pattern recognition, including structural pattern recognition, neuro-fuzzy techniques, and real-life applications.
Learning Outcome	• Mastery of fundamental concepts in pattern recognition. • In-depth understanding of various algorithms across different pattern recognition paradigms. • Comprehensive knowledge of theoretical aspects of feature selection, feature extraction, and projection techniques. • Ability to apply pattern recognition algorithms to real-world projects
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Pattern recognition and machine learning by Christopher Bishop, Springer Verlag, 2006.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman. The elements of Statistical Learning. Springer Verlag (2009).
- Fundamentals of Pattern Recognition and Machine Learning by Ulisses Braga-Neto. Springer Cham (2020)
- Probability, Random Variables and Stochastic processes by Papoulis and Pillai, 4th Edition, Tata McGraw Hill Edition.
- Linear Algebra and Its Applications by Gilbert Strand. Thompson Books.
- Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988

Course Number	CS4122
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Principles of Programming Languages
Learning Mode	Offline
Learning Objectives	To make students understand the existence of different programming language paradigms (i.e., logic, functional, procedural, object-oriented), their specific features, and to choose an appropriate language for a given application. To make students capable to learn new languages easily and to make clear and efficient use of any given language.
Course Description	The objective of this course is to study the design and implementation of programming languages from a foundational perspective.
Course Outline	Introduction: History of Programming Languages; Evolution of the Major Programming Languages; Art of Programming Language Design; Properties and Success of Programming Languages. Programming Language-Paradigms: Imperative (e.g. C, Pascal, Fortran); Functional (e.g. LISP, HASKELL, OCaml); Object Oriented (e.g. JAVA, C++, Scala); Logic-based (e.g. Prolog); Multiparadigm programming languages (e.g. Python, C++11). Programming Language Concepts: Values and Data Types; Block Structure; Scope, Binding and Lifetime of Variables; Static vs. Dynamic Typing; Static vs. Dynamic Scoping; Memory Management; Procedural Abstraction; Data Abstraction; Concurrency; etc. Case Study: Defining Syntax and Semantics of IMP (a simple WHILE-language) and COOL (Classroom Object Oriented Language).
Learning Outcome	• Understand a variety of concepts underpinning modern programming languages. • Understand the concepts and terms used to describe languages that support the imperative, functional, object-oriented, and logic programming paradigms. • Critically evaluate what paradigm and language are best suited for a new problem. • Solve problems using the functional paradigm. • Solve problems using the object-oriented paradigm. • Solve problems using the logic programming paradigm. • Understand how to design and implement your own (domain-specific) language.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Michael L. Scott, “**Programming Language Pragmatics**”, Morgan Kaufmann, 3rd Edition.
- Harold Abelson, Gerald Jay Sussman, Julie Sussman, “**Structure and Interpretation of Computer Programs**”, MIT Press, 2nd Edition.
- Ravi Sethi, K.V. Vishwanatha, “Programming Languages: Concepts and Constructs”, 2/e, Pearson Education, 2007.
- T.W. Pratt and M.V. Zelkowitz, “Programming Languages – Design and Implementation”, Prentice-Hall.
- Robert W. Sebesta, “Concepts of Programming Languages”, Addison-Wesley.
- D. A. Watt, “Programming Language Design Concepts”, John Wiley & Sons.
- Kenneth C. Loudon and Kenneth A. Lambert, “Programming Languages: Principles and Practice”, Cengage Learning.
- Recent Research Papers relevant to the course.

Course Number	CS4123
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Social Networks
Learning Mode	Offline
Learning Objectives	The major objectives of the course would be to make the students understand the basic concepts of social network, understand the fundamental concepts in analyzing the large-scale data that are derived from social networks, implement mining algorithms for social networks, and perform mining on large social networks and illustrate the results.
Course Description	This course delves into the analysis of data within social networks, emphasizing efficient strategies for managing large-scale networks. It presents fundamental theoretical findings in social network mining alongside practical exercises addressing critical topics within the field.
Course Outline	Introduction to social networks. Illustration of various social network mining tasks with real-world examples. Data characteristics unique to these settings and potential biases due to them. Social Networks as Graphs. Random graph models/ graph generators (Erdos-Renyi, power law, preferential attachment, small world, stochastic block models, Kronecker graphs), degree distributions. Models of evolving networks. Node based metrics, ranking algorithms (Pagerank). Graph visualisation. Social network exploration/ processing: Graph kernels, graph classification, clustering of social-network graphs, centrality measures, community detection and mining, degeneracy (outlier detection and centrality), partitioning of graphs. Information Diffusion in Social Networks: Information diffusion in graphs - Cascading behavior, spreading, epidemics, heterogeneous social network mining, influence maximization, outbreak detection; Opinion analysis on social networks - Contagion, opinion formation, coordination and cooperation. Dynamic social networks, Link prediction, Social learning on networks.
Learning Outcome	By completing the course, the students will be able to: • Understand the basic concepts of social networks • Understand the fundamental concepts in analyzing the large-scale data that are derived from social networks • Implement mining algorithms for social networks • Perform mining on large social networks and illustrate the results.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- David Easley and Jon Kleinberg, Networks, crowds, and markets, Cambridge University Press, 2010.
- Jure Leskovec, Anand Rajaraman and Jeffrey David Ullman, Mining of massive datasets, Cambridge University Press, 2014.

Course Number	CS4124
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Multimedia Systems
Learning Mode	Offline
Learning Objectives	The main objective of this course is to provide students with a comprehensive understanding of multimedia systems. Students will learn about the various components and technologies involved in multimedia systems, including audio, video, and image processing. They will explore the principles of multimedia compression, storage, and retrieval, as well as the techniques used for multimedia communication and networking. By the end of the course, students will have a solid theoretical foundation in multimedia systems and will be able to apply this knowledge to solve real-world problems in the field.
Course Description	The course begins with an introduction to multimedia systems, covering the basics of multimedia data representation and the different types of multimedia data. This is followed by a detailed study of multimedia compression techniques, including lossless and lossy compression methods for text, images, audio, and video. The course then explores multimedia storage and retrieval, discussing the different storage media and retrieval techniques used for multimedia data. Next, students will learn about multimedia communication and networking, including the protocols and architectures used for multimedia transmission over networks. The course concludes with a discussion of advanced topics in multimedia systems, such as quality of service, synchronization, and security.
Course Outline	Introduction to Multimedia Systems Understanding multimedia data types: Text, images, audio, and video. Multimedia Data Representation: Pixel-based representation for images, waveform representation for audio, and frame-based representation for video. Compression techniques for multimedia data: Lossy and lossless compression algorithms. Multimedia Storage and Retrieval Multimedia Networking and Streaming Multimedia Synchronization and Interactivity: Timecodes, timestamps, and synchronization protocols, Hypermedia, and interactive multimedia applications. Multimedia applications and trends: virtual and augmented reality
Learning Outcome	By the end of this course, students will be able to: • Understand the fundamental concepts and components of multimedia systems. • Analyse and evaluate different multimedia data types and their representation techniques. • Design and implement multimedia storage, retrieval, and streaming solutions. • Evaluate multimedia networking protocols and techniques for efficient multimedia transmission. • Implement multimedia synchronization and interactivity features in multimedia applications. • Explore real-world applications of multimedia systems and identify future trends in multimedia technology
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- Multimedia Systems: Algorithms, Standards, and Industry Practices" by Parag Havaladar, Gerard Medioni
- "Multimedia Computing: Algorithms, Systems, and Applications" by Ralf Steinmetz, Klara Nahrstedt
- "Introduction to Multimedia Systems" by Sugata Mitra, Tamalika Chaira

Course Number	CS4125
Course Credit	3-0-0-3
Course Title	Program Analysis and Verification
Learning Mode	Offline
Learning Objectives	This course will focus on static and dynamic program analysis techniques that can be used to perform various software engineering tasks. The students will learn the concepts behind the techniques and will apply their learning to develop analyses using the state-of-art tools.
Course Description	This course covers both foundations and practical aspects of the automated analysis of programs. It covers how to represent source codes in appropriate forms, enabling one to apply tools and techniques to extract relevant information about the code and to verify them.
Course Outline	Introduction: Program Analysis and Verification; Abstraction Vs. Approximation; Precision-Efficiency Dilemma. Control and Data Flow: Control-flow analysis; Data-flow analysis; Inter-procedural Analysis. Dependency Analysis: Program dependence graphs; Program slicing; Pointer and Alias analysis. Verification and Validation: Verification and validation strategies; Different testing methodologies; Introduction to formal verification approaches.
Learning Outcome	By taking this course, students will gain an understanding of the concepts and theories that underlie these analysis techniques. Students will also learn to design, implement, and leverage program analysis techniques to solve new verification problems.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Readings:

- Flemming Nielson, Hanne R. Nielson, Chris Hankin. Principles of Program Analysis, Springer, 1999.
- Edsger Wybe Dijkstra. A Discipline of Programming. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.
- David Gries. The Science of Programming. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1987.
- S. S. Muchnick and N. D. Jones, editors. Program Flow Analysis: Theory and Applications. Prentice-Hall: Englewood Cliffs, NJ, 1981.
- Cem Kaner, Hung Q. Nguyen, and Jack L. Falk. Testing Computer Software. John Wiley & Sons, Inc., New York, NY, USA, 1993.
- Recent Research Papers relevant to the course.

B. Tech. Elective - III						
1.	CS4117	Time Series Analysis	3	0	0	3
2.	CS4118	Computational Data Analysis	3	0	0	3
3.	CS4126	Graph Machine Learning	3	0	0	3
4.	CS4127	Bioinformatics	3	0	0	3
5.	CS4128	Graph Theory	3	0	0	3
6.	CS4129	Blockchain Technology	3	0	0	3

Course Number	CS4117
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Time Series Analysis
Learning Mode	Offline
Learning Objectives	• The course is designed to provide basic understanding time series analysis. • Develop Skills in statistical time series Analysis. • To learn variety of modeling techniques that can be used for time series analysis. • Gain proficiency in forecasting and anomaly detection methods • Apply the basic machine learning for time series analysis
Course Description	Using a set of fundamental techniques and broadly explains how time series analysis work at various levels of abstraction. The course introduces time series analysis with focus on applications
Course Outline	Basics of inferential and descriptive statistics: Population vs Sample; Measures of Central tendency, Measures of Variability, probability density functions, properties, mathematical expectation, hypothesis testing, ANOVA. Mathematical models for analysing time series data: Time Series Modelling, autoregressive integrated moving average (ARIMA), Exponential smoothing in time series analysis, process and the Box-Jenkins methodology. Outlier Analysis for Time Series, Multivariate Time Series Models and State-space Models, Forecasting Methods and Application Examples. Transfer Function Model Building. Imputation techniques, Point forecast and confidence intervals. Machine Learning Approaches for Time Series, Probabilistic Neural Networks, Different methods of estimation and inferences of modern dynamic stochastic general equilibrium models: simulated method of moments.
Learning Outcome	The student will be able to: • Appreciate understanding of the time series analysis, key terminology, and current industry trends in time series modeling • Evaluate time series model performance. • Create real-time applications, including anomaly detection and predictive maintenance
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Palit, Ajoy K., and Dobrivoje Popovic. *Computational intelligence in time series forecasting: theory and engineering applications*. Springer Science & Business Media, 2006.
- Box, George EP, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- Brockwell, Peter J., Richard A. Davis, and Matthew V. Calder. *Introduction to time series and forecasting*. Vol. 2. New York: springer, 2002.
- Pollock, David Stephen Geoffrey, Richard C. Green, and Truong Nguyen, eds. *Handbook of time series analysis, signal processing, and dynamics*. Elsevier, 1999.
- Shumway, Robert H., and David S. Stoffer. *Time series analysis and its applications: with R examples*. Springer, 2017.

Course Number	CS4118
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Computational Data Analysis
Learning Mode	Offline
Learning Objective	In this subject, the students will be trained with the knowledge of various computational techniques required for multi-dimensional data analysis such that they are able to apply these techniques in practice through programming, modeling etc.
Course Description	Modern day data is vast and diverse owing to their different acquisition systems and medium. This course aims to give an in-depth view to different data generation/acquisition mechanisms over diverse domains and the challenges incurred. It will discuss the role of computational data analysis techniques to understand and mathematically model data formation process. It will also teach them about the various data processing techniques required to manipulate and operate data to suit various objectives.
Course Outline	Understanding multi-dimensional data formation from physical acquisition devices with example cases in Remote Sensing, Geoscience, Medical sciences. Drawbacks and challenges in data acquisition, Necessity for computational modelling and analysis of data. Mathematical models for data formation and analysis, Probability models, Linear inverse optimization models, L1-L2 Regularizers, Minimizers, Cascade Modelling, Multiscale Modelling, Machine Learning models. Data Interpretation: Handling missing/corrupted data, Handling outliers, Imputation techniques, Interpolation techniques, Curve based approximation, non-convex optimization, sparse regularizers, Non-convex minimizers, Machine learning based. Data compression: Necessity, Applications, Lossless compression techniques, Lossy compression techniques, JPEG compression, Machine learning based. Statistical Models, Data preprocessing techniques in Machine learning, Signal processing techniques for multi-dimensional data, Application in various domains.
Learning Outcome	After completion of course, students will be able to • Understand data formation/generation process and the role of computational techniques in analyzing those data. • Apply the Mathematical principles behind computational techniques for data analysis. • Understand the utilities of statistical models and ML models in data analysis.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Signal Processing: A Mathematical Approach, Charles L. Byrne, Second Edition, Chapman & Hall, 2014.
- Digital Functions and Data Reconstruction: Digital-Discrete Methods, Li M Chen, Springer, 2013.
- Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, Bernhard Mehlig, Cambridge University Press, 2021
- Signal Processing and Machine Learning with Applications, Michael M. Richter, Sheuli Paul, Veton Këpuska, Marius Silaghi, Springer Cham, 2022
- Data Compression: The Complete Reference, David Solomon, 4th Edition, Springer, 2007

Course Number	CS4126
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Graph Machine Learning
Learning Mode	Offline
Learning Objectives	Several real world systems can be represented as a network of entities that are connected to each other through some relations. Often the number of entities is immensely large, thus forming a very large network. Typical examples of such large networks include network of entities in knowledge graphs, co-occurrence graph of the keywords in natural languages, interaction graph of users in social networks, protein-protein interaction graphs and the network of routers in Internet to name a few. Study of these networks is often needed for relational learning tasks, as well as for developing frameworks for representing the intrinsic structure of the data. This course will mainly deal with both the traditional as well as current state of the art machine learning techniques to be applied on Graphs for different downstream tasks.
Course Description	The course will provide knowledge on the representation and statistical descriptions of large networks, along with traditional machine learning and deep learning techniques applied on graphs. Several use cases of Graph Machine Learning across different domains including Natural Language Processing, Social Network Analysis and Computational Biology would be studied.
Course Outline	Introduction and background knowledge of graphs; Network analysis metrics like paths, components, degree distribution, clustering, degree correlations, centrality etc., social network analysis methods; Spectral Analysis of Graphs and its applicability to graph partitioning and community detection; Overview of machine learning applications on graphs; Shallow embedding and deep Learning techniques for generating node and graph representations – Graph Neural Networks, Graph Attention Networks Random Networks; Graph Evolution, Generative models for graphs
Learning Outcome	Course training via lectures & tutorial sessions to • Represent and analyze the structure of graphs • Discover recurring and significant patterns of interconnections in your data with network motifs and community structure. • Gain Knowledge on traditional machine learning techniques applied on graphs • Leverage graph-structured data to make better predictions using graph neural networks • Understand the problems in dealing with large graphs for machine learning tasks and learn how to improvise.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested readings:

- M.E.J. Newman, Networks - An introduction , Oxford Univ Press, 2010.
- Yao Ma and Jilian Tang, Deep Learning on Graphs, Cambridge University Press, 2021
- Goyal, Palash and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey.” *Knowl.-Based Syst.* 151 (2018): 78-94.

Course Number	CS4127
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Bioinformatics
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Gain a thorough understanding of fundamental concepts in bioinformatics. (b) Develop problem-solving skills by implementing basic algorithms tailored for bioinformatics applications. (c) Explore various paradigms and approaches in bioinformatics as applied to biological data, such as sequence alignment, clustering, and classification. (d) Achieve proficiency in designing and implementing real-life bioinformatics projects that integrate deep learning techniques for data analysis and interpretation.
Course Description	This interdisciplinary course on bioinformatics aims to equip students with the knowledge and skills necessary to analyze and interpret biological data using computational tools and techniques. By focusing on fundamental concepts and providing hands-on experiences, students will learn to manage and analyze large-scale biological datasets. Through a combination of lectures, practical lab sessions, and collaborative projects, students will explore topics such as sequence alignment, gene expression analysis, protein structure prediction, and biological databases. Upon completion, students will be proficient in utilizing bioinformatics software and algorithms to address complex biological questions, preparing them for careers in research, biotechnology, and related fields.
Course Outline	Overview of biological databases: Protein Data Bank, SCOP, genome databases, and Cambridge Structural Database. Introduction to protein structures and biophysical methods for structure determination. Protein structure analysis, visualization techniques, and molecular modelling. Mining techniques using protein sequences and structures, including short sequence alignments and multiple sequence alignments. Phylogenetic analysis, genome context-based methods, and RNA/transcriptome analysis techniques. Mass spectrometry applications in proteome and metabolome analysis. Protein docking, dynamics simulation, and algorithms for handling big biological data challenges. Applications of Bioinformatics.
Learning Outcome	• Mastery of fundamental principles and techniques in bioinformatics, including sequence analysis, structural biology, and genomic data interpretation. • Proficiency in applying pattern recognition algorithms to solve biological data problems, such as sequence alignment, clustering, and classification. • Ability to critically analyze and interpret bioinformatics data using computational tools and techniques. • Understanding of the interdisciplinary nature of bioinformatics and its applications in biological research and medicine.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Mount, D.W., Bioinformatics: Sequence and Genome Analysis, Cold. Spring Harbor Laboratory Press, 2001.
- Protein Bioinformatics: From Sequence to Function by M. Michael Gromiha Academic Press, 2010
- Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins 4th Edition, by Andreas D. Baxevanis (Editor), Gary D. Bader (Editor), David S. Wishart (Editor), WILEY
- C. Branden and J. Tooze (eds) Introduction to Protein Structure, Garland, 1991

Course Number	CS4128
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Graph Theory
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) to provide students with a comprehensive understanding of advanced topics in graph theory; (b) to analyze and solve complex problems involving graphs, understand and apply various graph algorithms, and explore the theoretical underpinnings of graph theory; and (c) develop critical thinking and analytical skills, enabling them to approach and solve real-world problems using graph theory concepts.
Course Description	This course delves into the advanced aspects of graph theory, building on foundational knowledge to explore more complex and abstract concepts. Students will study advanced topics such as graph coloring, planarity, network flows, and extremal graph theory. The course will also cover specialized graph classes, advanced algorithms, and their applications in various fields such as computer science, biology, and social sciences. Through a blend of theoretical exploration and practical application, students will gain a deep understanding of how graph theory can be used to model and solve intricate problems.
Course Outline	Introduction and Review of Basic Concepts- Review of fundamental graph theory concepts (graphs, subgraphs, isomorphism, paths, cycles), Types of graphs (simple, multigraphs, weighted graphs, directed and undirected graphs), Graph representation (adjacency matrix, adjacency list, incidence matrix) Graph Connectivity- Connectivity and components, Menger's Theorem, Network flow and cuts, Applications in network design and reliability, Vertex coloring, Edge coloring, Chromatic number and chromatic polynomial, Applications in scheduling and register allocation Euler's formula for planar graphs, Kuratowski's and Wagner's theorems, Graph embedding, Applications in geographic mapping and circuit design, Graph minors and the Minor Theorem, Well-quasi-ordering and its implications, Applications in algorithm design and complexity theory Shortest path algorithms (Dijkstra's, Bellman-Ford, Floyd-Warshall), Maximum flow algorithms (Ford-Fulkerson, Edmonds-Karp, Push-Relabel), Matching algorithms (Bipartite matching, Hungarian algorithm, Stable matching), Basic models of random graphs (Erdős–Rényi model), Properties of random graphs (connectivity, diameter, phase transition), Applications in network theory and epidemiology Eigenvalues and eigenvectors of graphs, Laplacian matrix and its properties, Cheeger's inequality and applications, Applications in clustering and data analysis, Small-world networks, Scale-free networks, Community detection algorithms, Applications in social networks, biological networks, and information networks Graph theory in computational biology (protein-protein interaction networks, metabolic networks), Graph theory in computer networks (routing, fault tolerance), Graph theory in machine learning (graph neural networks, data mining)
Learning Outcome	• Understand and articulate advanced concepts in graph theory. • Analyze and solve complex problems involving graphs. • Apply various graph algorithms to practical problems. • Understand and implement graph coloring techniques and their applications. • Analyze the properties of planar graphs and utilize graph drawing algorithms. • Apply network flow algorithms to solve problems in various fields. • Understand and apply principles of extremal graph theory, including Turán's theorem and Ramsey theory. • Identify and work with specialized graph classes such as bipartite, perfect, and chordal graphs. • Develop and implement advanced graph algorithms, including shortest path, matching, and covering algorithms.

	• Apply graph theory concepts to real-world scenarios in computer science, biology, and social sciences. • Design and conduct practical projects that demonstrate the application of advanced graph theory concepts.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term.

Suggested Reading

- Diestel, R. (2017). *Graph Theory* (5th ed.). Springer.
- Bondy, J. A., & Murty, U. S. R. (2008). *Graph Theory* (1st ed.). Springer.
- West, D. B. (2001). *Introduction to Graph Theory* (2nd ed.). Prentice Hall.
- Gross, J. L., & Yellen, J. (2005). *Graph Theory and Its Applications* (2nd ed.). CRC Press.
- Chartrand, G., & Zhang, P. (2012). *A First Course in Graph Theory* (1st ed.). Dover Publications.

Course Number	CS4129
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Blockchain Technology
Learning Mode	Offline
Learning Objectives	This course will introduce the fundamentals of the blockchain technology. It will highlight the use of blockchain technology in different applications and the learners will be able to develop decentralized applications.
Course Description	This course provides an introductory background of this revolutionary technology, followed by an interesting case study on bitcoin to demonstrate how the technology works. Following this, we would introduce Ethereum and Hyperledger. In addition, the course includes a number of hands-on sessions where we introduce basic blockchain tools and techniques, such as geth, ganache, remix, metamask, truffle, hyperledger, and real case studies.
Course Outline	Introduction and History; Blockchain Foundations; Generic elements of a blockchain; Features of blockchain; Types of blockchain; Applications of blockchain technology; Cryptocurrency and bitcoin basics; Introduction to Ethereum/Hyperledger and Programming; Privacy, Safety and Security Issues in blockchain; Some ongoing research topics.
Learning Outcome	• Gain proficiency in blockchain technology. • Understanding of how bitcoin/ethereum/hyperledger work. • Hands-on experience with various blockchain platforms, tools and techniques.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction, Princeton University Press, 2016.
- Roger Wattenhofer, The Science of the Blockchain, Inverted Forest Publishing, First Edition, 2016.
- Recent Research Papers relevant to the course.

Course number	CS5101
Course Credit (L-T-P-C)	3-1-0-4
Course Title	Design and Analysis of Algorithms
Learning Mode	offline
Learning Objectives	The objective of this course is to equip students with a solid understanding of data structures and algorithms, enabling them to design, analyze, and implement efficient algorithms to solve complex computational problems. The course covers fundamental topics such as data structures, complexity analysis, sorting and searching techniques, problem-solving strategies, graph algorithms. By the end of the course, students will have developed the skills to critically analyze algorithm efficiency and apply advanced algorithms in practical scenarios.
Course Description	This course will provide understanding of advanced methods to solve problems on computers. It will also provide an overview to analyze those theoretically.
Course Outline	Fibonacci heap, unionfind, splay trees. Amortized complexity analysis Randomized algorithms Reducibility between problems and NP-completeness: discussion of different NP-complete problems like satisfiability, clique, vertex cover, independent set, Hamiltonian cycle, TSP, knapsack, set cover, bin packing, etc. Backtracking, branch and bound Approximation algorithms: Constant ratio approximation algorithms. Application areas(i)Geometric algorithms: convex hulls, nearest neighbor, Voronoi diagram, etc.(ii)Algebraic and number-theoretic algorithms: FFT, primality testing, etc.(iii)Graph algorithms: network flows, matching, etc.(iv)Optimization techniques: linear programming
Learning Outcome	By the end of this course, students will be able to solve problems that are computationally intractable
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Mark Allen Weiss, "Data Structures and Algorithms in C++", Addison Wesley, 2003.
- Adam Drozdek, "Data Structures and Algorithms in C++", Brooks and Cole, 2001.
- Aho, Hopcroft and Ullmann, "Data structures and Algorithm", Addison Welsey, 1984.

- Introduction to Algorithms Book by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen
- Sanjoy Dasgupta, Christos H. Papadimitriou and Umesh V. Vazirani, Algorithms, Tata McGraw-Hill, 2008.
- Steven Skiena, The Algorithm Design Manual, Springer
- Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
- Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
- Udi Manber, Algorithms – A Creative Approach, Addison-Wesley, Reading, MA, 1989.
- Tim Roughgarden, Algorithms Illuminated

SEMESTER VIII						
Sl. No.	Subject Code	Subject Name	L	T	P	C
1.	CS5201	Advanced Artificial Intelligence	3	0	0	3
2.	CS5202	Theoretical Computer Science	3	0	0	3
3.	CS62XX	M. Tech. Elective – I	3	0	0	3
4.	RM6201	Research Methodology	3	1	0	4
5.	CS4299	Project II	0	0	12	6
TOTAL			12	1	12	19

Course Number	CS5201
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Artificial Intelligence
Learning Mode	Offline
Learning Objectives	• To understand the principles of Artificial Intelligence and the nature of intelligent agents. • To learn various problem-solving techniques, including informed search and exploration. • To gain proficiency in handling constraint satisfaction problems and adversarial search. • To develop a solid foundation in knowledge representation, first-order logic, and propositional logic. • To learn to plan and act effectively in real-world AI applications. • To grasp the concepts of uncertain knowledge and probabilistic reasoning. • To make informed decisions using simple and complex decision-making models. • To acquire skills in learning from observations and applying statistical learning methods. • To explore advanced AI techniques and their practical applications.
Course Description	This course offers an in-depth exploration of advanced concepts in Artificial Intelligence (AI). Students will delve into the theoretical underpinnings and practical applications of AI, examining intelligent agents, the nature of environments, and advanced problem-solving techniques. The curriculum covers informed search and exploration, constraint satisfaction problems, adversarial search, and knowledge representation. Students will also explore reasoning with first-order and propositional logic, planning and acting in real-world scenarios, and handling uncertainty through probabilistic reasoning. The course concludes with statistical learning methods and advanced AI techniques, providing a comprehensive understanding of AI's capabilities and applications.
Course Outline	Introduction and motivation Artificial Intelligence, intelligent agents, nature of environments, Problem-solving by searching, informed search and exploration, constraint satisfaction problem, adversarial search, Knowledge and reasoning, first order logic, inference and propositional logic, knowledge representation, Planning and acting in real world of AI agent Uncertain knowledge and reasoning, uncertainty, probabilistic reasoning, making simple and complex decisions Learning from observations and knowledge, statistical learning methods, Some advanced techniques of AI and its applications

Learning Outcome	Upon completing this course, students will be able to: • Analyze and implement intelligent agents in various environments. • Apply informed search techniques to solve complex problems. • Formulate and solve constraint satisfaction problems and engage in adversarial search strategies. • Represent and reason with knowledge using first-order and propositional logic. • Develop and execute plans in real-world AI scenarios. • Manage uncertainty and employ probabilistic reasoning to make sound decisions. • Utilize statistical learning methods to derive insights from data. • Implement advanced AI techniques in real-world applications. • Demonstrate a comprehensive understanding of advanced AI concepts and their implications.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Russell, S. J., & Norvig, P. (2016). Artificial intelligence: A modern approach. Pearson.
- Poole, D. L., & Mackworth, A. K. (2010). Artificial Intelligence: foundations of computational agents. Cambridge University Press.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: Springer.

Course Number	CS5202
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Theoretical Computer Science
Learning Mode	Offline
Learning Objectives	• Explore advanced topics in theoretical computer science, including computational complexity, automata theory, and algorithms. • Develop rigorous mathematical reasoning and problem-solving skills applicable to theoretical computer science. • Understand foundational concepts such as Turing machines, formal languages, and computational models. • Engage in independent research and scholarly exploration of theoretical computer science topics. • Apply theoretical insights to practical problems in computer science and related fields.
Course Description	This course offers an advanced study of theoretical computer science, focusing on formal models of computation, computational complexity theory, and algorithmic analysis. Students will delve into abstract concepts and mathematical techniques essential for understanding the limits and capabilities of computing systems. Topics covered include formal languages, automata theory, computability theory, complexity classes, and advanced algorithms. Through lectures, seminars, and research projects, students will develop a deep understanding of theoretical frameworks and their implications for solving real-world computational problems.
Course Outline	Introduction to Theoretical Computer Science , Historical overview and scope of theoretical computer science , Mathematical foundations in computer science , Overview of formal models of computation Automata Theory , Finite automata and regular languages , Context-free grammars and pushdown automata , Turing machines and computability Formal Languages and Computability , Formal language definitions and properties , Decidability and undecidability , Church-Turing thesis and

	implications Computational Complexity , Time and space complexity classes , NP-completeness and beyond , Complexity hierarchies and reductions Advanced Topics in Algorithms , Design and analysis of algorithms , Approximation algorithms and randomized algorithms , Advanced data structures Logic in Computer Science , Propositional and first-order logic , Model theory and logical reasoning , Applications of logic in computing Theory of Computation , Formal models beyond Turing machines , Quantum computing and computational models , Complexity aspects of quantum computation Cryptography and Computational Complexity , Basics of cryptography and cryptanalysis , Complexity-theoretic foundations of cryptography , Applications of cryptography in secure computation
Learning Outcome	• Demonstrate advanced knowledge of theoretical computer science principles and methodologies. • Apply mathematical reasoning and formal methods to analyze computational problems. • Evaluate the computational complexity of algorithms and problems using theoretical frameworks. • Conduct independent research in theoretical computer science and contribute to scholarly discourse. • Translate theoretical insights into practical solutions for complex computational challenges.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- "Introduction to the Theory of Computation" (3rd Edition) by Michael Sipser
- "Computational Complexity: A Modern Approach" (1st Edition) by Sanjeev Arora and Boaz Barak
- "Automata Theory, Languages, and Computation" (3rd Edition) by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman
- "Algorithm Design" (1st Edition) by Jon Kleinberg and Éva Tardos
- "Introduction to Algorithms" (3rd Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

M. Tech. Elective - I						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6201	Artificial Internet of Things	3	0	0	3
2.	CS6202	Game Theory	3	0	0	3
3.	CS6203	Text Mining & Analytics	3	0	0	3
4.	CS6204	Knowledge Distillation	3	0	0	3
5.	CS6206	Selected Topics in Wireless Networks	3	0	0	3
6.	CS6207	Advanced Big Data Analytics	3	0	0	3
7.	CS6208	Quantum Machine Learning	3	0	0	3
8.	CS6210	Selective Topics in Generative AI	3	0	0	3
9.	CS6215	Quantum Cyber Security	3	0	0	3
10.	CS6211	Selected Topics in Cryptography	3	0	0	3
11.	CS6216	High Performance Computing	3	0	0	3

Course Number	CS6201
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Artificial Internet of Things
Learning Mode	Offline
Learning Objectives	• Gain a comprehensive understanding of the convergence of Artificial Intelligence (AI) and Internet of Things (IoT), including basic concepts, architectures, and applications. • Learn various AI techniques and their applications in IoT, including machine learning, deep learning, and data analytics. • Develop skills in designing and implementing IoT systems, integrating sensors, and managing data flow. • Understand the processes for collecting, storing, processing, and analyzing IoT data using AI techniques. • Identify and mitigate security risks and privacy concerns in AIoT systems. • Analyze various real-world applications of AIoT in industries such as healthcare, smart cities, agriculture, and manufacturing. • Understand the regulatory and ethical considerations related to AIoT technologies and their deployment.
Course Description	This course provides an in-depth exploration of the convergence of Artificial Intelligence (AI) and the Internet of Things (IoT), known as AIoT. It covers the fundamental principles and technologies of both AI and IoT, demonstrating how they can be integrated to create intelligent, autonomous systems. Students will learn about IoT architecture, AI algorithms, machine learning, data analytics, and the implementation of AI-driven IoT solutions. Through hands-on projects and real-world case studies, students will gain practical experience in developing smart applications for various domains such as smart cities, healthcare, industrial automation, and smart homes.
Course Outline	Introduction to AIoT, Intersection of AI and IoT, Benefits and challenges of AIoT Fundamentals of IoT, IoT Architecture and Protocols, Layers of IoT architecture, Communication protocols and standards, IoT Devices and Sensors Fundamentals of Artificial Intelligence, Machine Learning and Deep Learning, Overview of AI tools and frameworks AIoT System Architecture, Components and Designing AIoT, Edge Computing in AIoT, Edge vs. cloud computing, AI Models for IoT

	Data Management in AIoT, Data Processing and Analysis, Handling large-scale IoT data, Big data technologies and platforms AIoT Applications and Use Cases: Smart Homes and Buildings, Healthcare and Wearables, Industrial IoT (IIoT), Smart Cities and Transportation AIoT Platforms and Tools: AI Development Tools, Case Studies of AIoT Solutions, AIoT Project Development, Future Trends and Innovations in AIoT
Learning Outcome	At the end of course, students will learn: • Students should grasp the foundational concepts of AI and IoT, including machine learning algorithms, data analytics, sensor technologies, and network protocols. • Ability to integrate AI algorithms with IoT devices and platforms to create intelligent systems capable of data collection, analysis, and decision-making in real-time. • Proficiency in developing AI-driven IoT applications, including sensor data processing, predictive analytics, anomaly detection, and automation. • Awareness of security challenges and solutions in AIoT systems, including data privacy, authentication, encryption, and intrusion detection. • Knowledge of optimization techniques for AIoT systems to enhance performance, scalability, and energy efficiency.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Olivier Hersent, David Boswarthick, and Omar Elloumi, The Internet of Things: Key Applications and Protocols, Wiley
- Maciej Kranz, Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry, Wiley
- John Paul Mueller and Luca Massaron, Machine Learning for the Internet of Things: Practical Guide, Packt

Course Number	CS6202
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Game Theory
Learning Mode	Offline
Learning Objectives	• Learn the principles of decision theory and its relevance to game theory. • Understand and analyze extensive form games, including game trees and backward induction. • Identify and compute pure and mixed strategy Nash equilibria. • Analyze matrix games, specifically two-player zero-sum games. • Understand Bayesian games and apply Bayesian equilibrium concepts to games with incomplete information. • Analyze and compute subgame perfect equilibria in dynamic games. • Explore coalitional games, including the core and the Shapley value. • Explore auction theory and its various models and applications. • Utilize game theory concepts in practical applications such as IoT, wireless networks, and cloud computing.
Course Description	This course aims to establish a solid foundation in both game theory and mechanism design, enabling participants to apply these principles rigorously to solve problems. By the end of the course, students will be equipped to model real-world scenarios using game theory, analyze these scenarios with game-theoretic concepts, and design effective and robust solutions, including mechanisms, algorithms, and protocols suitable for rational and intelligent agents.
Course Outline	Non-cooperative Game Theory: Decision theory, Extensive Form Games, Strategic Form Games, Dominant Strategy Equilibria, Pure Strategy Nash Equilibrium, Mixed Strategy Nash Equilibrium, Computation of Nash Equilibrium, Complexity of Computing Nash Equilibrium, Matrix Games (Two Player Zero-sum Games), Bayesian Games, Subgame Perfect Equilibrium. Cooperative Game: Correlated Strategies and Correlated Equilibrium, Two Person Bargaining Problem, Coalitional Games, Core, Shapley Value. Mechanism Design: Introduction to Mechanism Design, Social Choice Functions and their properties, Incentive Compatibility, Auction theory and its variants. Applications: IoT, Wireless Networks, Cloud Computing

Learning Outcome	By the end of this course, students will be able to: • Describe the principles of decision theory and its importance in game theory. • Formulate and solve strategic form games, identifying dominant strategy equilibria and Nash equilibria. • Analyze and solve matrix games, particularly two-player zero-sum games. • Formulate Bayesian games and determine Bayesian equilibria for games with incomplete information. • Compute subgame perfect equilibria for dynamic games using appropriate techniques. • Apply the concepts of correlated strategies and correlated equilibria in cooperative settings. • Analyze and solve two-person bargaining problems. • Analyze social choice functions and their properties, focusing on incentive compatibility. • Utilize game theory concepts to address practical problems in IoT, wireless networks, and cloud computing.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbook:

- M. Osborne, An Introduction to Game Theory, Oxford University Press.
- Y. Narahari. Game Theory and Mechanism Design. IISc Press and the World Scientific.

Reference Book:

- M. Maschler, E. Solan, and S. Zamir, Game Theory. Cambridge University Press
- D. Niyato, & W. Saad. Game theory in wireless and communication networks. Cambridge University Press.

Course Number	CS6203
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Text Mining and Analytics
Learning Mode	Offline
Learning Objectives	• To understand the fundamental principles and scope of text mining and analytics. • To acquire skills in data collection, cleaning, and integration for text data. • To learn text preprocessing techniques including tokenization, stemming, stopword removal, and normalization. • To construct knowledge graphs by linking entities and extracting relationships. • To identify and mine frequent patterns and apply advanced pattern mining techniques. • To extract features from text data and apply clustering and classification methods. • To implement practical applications such as sentiment analysis and text summarization. • To utilize advanced techniques for enhanced text data analysis and mining.
Course Description	This course provides a comprehensive understanding of the principles and techniques of text mining and analytics. Students will learn about data collection, cleaning, integration, and preprocessing methods essential for handling text data. The course covers knowledge graph construction, pattern mining, feature extraction, and advanced text clustering and classification techniques. Practical applications such as sentiment analysis and text summarization are also explored. By the end of the course, students will be prepared to tackle real-world challenges in data mining and text analytics.
Course Outline	Text mining and analytics introduction: Overview, motivation, scope, Data Collection and Pre-processing: Techniques for collecting data from various sources, Text data cleaning and integration, descriptive analytics Text preprocessing: tokenization, stemming, stopword removal, and normalization Knowledge graph construction: Basics of graphs, entity linking, relationship extraction Concepts of frequent patterns, closed patterns, max-patterns, and association rules, mining frequent patterns: apriori algorithm, pattern-growth approach.

	Advanced: mining sequential patterns Feature extraction, Bag-of-Words, TF-IDF, word embeddings Clustering and classifying text data, Expectation-maximization (EM) algorithm for text data, Latent Dirichlet Allocation (LDA) for topic modeling, and some advanced techniques Some applications: sentiment analysis, text summarization, etc. Some advanced topics and project
Learning Outcome	• By the end of this course, students will be able to: • Understand the core principles and scope of text mining and analytics. • Collect, clean, and integrate text data from various sources. • Apply text preprocessing techniques such as tokenization, stemming, and normalization. • Construct and utilize knowledge graphs for entity linking and relationship extraction. • Identify and mine various patterns in text data, including frequent, closed, and sequential patterns. • Extract features from text data using methods like Bag-of-Words, TF-IDF, and word embeddings. • Perform text clustering and classification using algorithms such as EM and LDA. • Implement practical text analytics applications such as sentiment analysis and text summarization. • Utilize advanced techniques for enhanced text data analysis and mining.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading

- Srivastava, A. N., & Sahami, M. (Eds.). (2009). Text mining: Classification, clustering, and applications. CRC press.
- Chakraborty, G., Pagolu, M., & Garla, S. (2014). Text mining and analysis: practical methods, examples, and case studies using SAS. SAS Institute.
- Sarkar, D. (2016). Text analytics with python (Vol. 2). New York, NY, USA:: Apress.
- Witten, I. H., Frank, E., Hall, M. A., Pal, C. J., & Data, M. (2005, June). Practical machine learning tools and techniques. In Data mining (Vol. 2, No. 4, pp. 403-413). Amsterdam, The Netherlands: Elsevier.

Course Number	CS6204
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Knowledge Distillation
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) understand and apply knowledge distillation techniques; (b) master deep neural network compression methods; (c) deploy ML/DNN models on edge devices like Raspberry Pi and others; (d) analyze and optimize model performance in resource-constrained environments; (e) identify the research opportunity in the domain of knowledge distillation and DNN compression on resource-constrained devices.
Course Description	This course delves into advanced techniques for enabling machine learning on resource-constrained devices. Beginning with an introduction to on-device training, students will explore the principles and methods of knowledge distillation and deep neural network (DNN) compression. The course covers practical strategies for implementing machine learning and deep neural networks on devices with limited computational resources. Additionally, students will learn to combine knowledge distillation and compression techniques to optimise performance, making sophisticated machine-learning models viable on edge devices.
Course Outline	• Introduction to on-device training: Overview of resource-constrained edge devices and their significance, possibilities of enabling machine learning (ML) and deep neural networks (DNN) models on resource-constrained devices, applications and use cases of ML/DNN on edge devices. • Knowledge Distillation: Concept and principles of knowledge distillation, Teacher-student model framework, Applications and benefits of knowledge distillation. Advanced techniques in knowledge distillation, Implementation of knowledge distillation in various frameworks, and Practical exercises on distilling models. • Deep Neural Network Compression: Overview of DNN compression techniques, Quantization and its impact on model performance, Pruning methods for model size reduction. Low-rank factorization, Weight sharing and clustering, Hands-on implementation of compression techniques. • ML/DNN on resource-constrained devices: Introduction to edge devices: Raspberry Pi, NVIDIA Jetson, etc, Setting up an AI development environment on Raspberry Pi, Case study: Running a pre-trained model on Raspberry Pi. TensorFlow Lite, ONNX, etc, Practical exercises with TensorFlow Lite on Raspberry Pi. • Combining Knowledge Distillation and Compression: Integrating knowledge distillation and compression for optimal performance, Strategies for balancing accuracy and efficiency, Real-world examples and case studies.

Learning Outcome	• Explain and implement knowledge distillation techniques. • Apply DNN compression methods such as quantisation and pruning. • Set up and optimise ML/DNN models on Raspberry Pi using TensorFlow Lite and ONNX. • Evaluate and enhance ML/DNN model performance on edge devices. • Create real-time applications, including object detection and predictive maintenance. • Plan, develop and present comprehensive projects that may lead to publication.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Deep Learning for Edge AI” by John Doe
- Knowledge Distillation: Principles, Methods and Applications” by Jane Smith
- Official documentation and tutorials for TensorFlow Lite, ONNX, and edge devices
- “Knowledge Distillation: A Survey” Jianping Gou, Baosheng Yu, Stephen John Maybank, Dacheng Tao
- K. Nan, S. Liu, J. Du and H. Liu, "Deep model compression for mobile platforms: A survey," in Tsinghua Science and Technology, vol. 24, no. 6, pp. 677-693, Dec. 2019, doi: 10.26599/TST.2018.9010103.
- Mishra et al.. "A survey on deep neural network compression: Challenges, overview, and solutions."

Course Number	CS6206
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Selected Topics in Wireless Networks
Learning Mode	Offline
Learning Objectives	In this subject, the students will be trained with the knowledge of 802.11 wireless networks, including protocol knowledge and the associated security vulnerabilities.
Course Description	In the consumer, industrial, and military sectors, 802.11-based wireless access networks have been widely used due to their convenience. This application, however, is reliant on the unstated assumptions of availability and anonymity. The management and media access protocols of 802.11 may be particularly vulnerable to malicious denial-of-service (DoS) and various security attacks. This course analyzes these 802.11-specific attacks, including their applicability, effectiveness, and proposed low-cost implementation improvements to mitigate the underlying vulnerabilities.
Course Outline	Introduction to Wireless Networks: Basic principles, types of wireless networks (Wi-Fi, Bluetooth, cellular), and network topologies. Wireless Communication Fundamentals: Radio frequency, signal propagation, modulation techniques, and interference management. Network Protocols and Standards: IEEE 802.11 (Wi-Fi), IEEE 802.15 (Bluetooth), and cellular standards (2G, 3G, 4G, 5G). Network Design and Architecture: System design, frequency reuse, and resource allocation. Mobility and Handoff: Techniques for managing mobility, handoff processes, and roaming. Security in Wireless Networks: Security protocols, encryption, and threat mitigation. Emerging Technologies: Overview of 6G, IoT, in-network caching
Learning Outcome	On successful completion of the course, students should be able to: • Understand the fundamentals of 802.11 wireless networks • Describe the WLAN services-association, disassociation, re-association, distribution, integration, authentication, de authentication and data delivery services • Comprehend the vulnerabilities associated with 802.11 protocol.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Text Books and References:

- "Wireless Communications: Principles and Practice" by Theodore S. Rappaport (2nd Edition)
- "802.11 Wireless Networks: The Definitive Guide" by Matthew S. Gast (2nd Edition)
- "Wireless Communications & Networks" by William Stallings (2nd Edition)
- "Wireless Communications: Principles and Practice" by Andreas F. Molisch (2nd Edition)

- "Fundamentals of Wireless Communication" by David Tse and Pramod Viswanath (1st Edition)
- "Next Generation Wireless LANs: 802.11n and 802.11ac" by Eldad Perahia and Robert Stacey (2nd Edition)
- "Wireless Networking: Understanding Internetworking Challenges" by Anurag Kumar, D. Manjunath, and Joy Kuri (1st Edition)
- "Wireless Communications: Principles and Practice" by Kaveh Pahlavan and Prashant Krishnamurthy (1st Edition)

Course Number	CS6207
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Big Data Analytics
Learning Mode	Offline
Learning Objectives	The primary objective of this course is to equip students with advanced knowledge and skills in big data analytics. By the end of the course, students will be able to understand and apply advanced data analysis techniques, develop and implement big data solutions, and leverage big data technologies for strategic decision-making. Additionally, students will gain proficiency in using big data tools and platforms, enhance their ability to handle and analyze large datasets, and develop critical thinking skills for solving complex data-driven problems.
Course Description	This course provides an in-depth exploration of advanced big data analytics, focusing on the theoretical foundations, practical techniques, and cutting-edge technologies in the field. Students will learn about various aspects of big data, including data acquisition, storage, processing, and analysis. The course covers advanced topics such as machine learning algorithms for big data, real-time data processing, and big data visualization. Emphasis will be placed on the use of big data tools and platforms such as Hadoop, Spark, and NoSQL databases. Through hands-on projects and case studies, students will develop the skills needed to design and implement big data solutions for a variety of applications.
Course Outline	Introduction to Big Data Analytics: Definition and characteristics (Volume, Velocity, Variety, Veracity, and Value), Importance and challenges of Big Data. Big Data Ecosystem: Components and architecture, Key players and technologies in Big Data (e.g., Hadoop, Spark). Big Data vs. Traditional Data: Differences in processing and analysis, Applications of Big Data Analytics- Industry-specific applications and Case studies Data Acquisition and Storage: Structured, semi-structured, and unstructured data and Data generation and collection methods. Distributed file systems (e.g., HDFS), NoSQL databases (e.g., MongoDB, Cassandra), and Cloud storage options, ETL (Extract, Transform, Load) processes, Data pipelines and workflow automation, Insuring data integrity and accuracy, Data privacy and security considerations Data Processing Frameworks: Hadoop MapReduce architecture and workflow, Advantages and limitations, Apache Storm, Apache Flink, and Kafka Streams, Real-time data processing and its significance, Apache Spark architecture and RDDs (Resilient Distributed Datasets), Spark SQL, Spark Streaming, and MLlib

	Machine Learning for Big Data: Introduction to Machine Learning, Machine Learning Algorithms, Machine Learning Tools and Libraries, Training and evaluating models on large datasets, Scalability and performance optimization Real-Time Data Processing: Importance and applications of real-time analytics, Apache Kafka, Apache Flink, and Apache Storm, Designing and implementing real-time data workflows, Industry examples and best practices Big Data Visualization: Making data comprehensible and actionable, Visualization Tools and Techniques, Building user-friendly and interactive data dashboards, Intersection of data science and big data analytics, Integrating AI techniques with big data analytics, Processing and analyzing IoT-generated data, Distributed computing at the edge of networks, and Industry-Specific Case Studies- Healthcare, finance, retail, and other industries.
Learning Outcome	• Understand the key concepts and significance of big data analytics. • Acquire, store, and manage large datasets using appropriate big data technologies. • Apply advanced data processing techniques using Hadoop and Spark. • Implement machine learning algorithms for big data applications. • Perform real-time data processing and analysis. • Utilize big data visualization tools to interpret and present data insights. • Develop and implement comprehensive big data solutions for various industry applications. • Critically evaluate and solve complex data-driven problems using advanced analytics techniques.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Marz, N., & Warren, J. (2015). Big Data: Principles and Best Practices of Scalable Real-Time Data Systems (1st ed.). Manning Publications.
- White, T. (2015). Hadoop: The Definitive Guide (4th ed.). O'Reilly Media.
- Karau, H., & Warren, R. (2017). High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark (1st ed.). O'Reilly Media.
- Gulla, U., Gupta, S., & Kumar, V. (2020). Practical Big Data Analytics: Hands-on Techniques to Implement Enterprise Analytics and Machine Learning Using Hadoop, Spark, NoSQL and R (2nd ed.). Packt Publishing.
- Zikopoulos, P. C., Eaton, C., & deRoos, D. (2012). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data (1st ed.). McGraw-Hill Education.

Course Number	CS6208
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Quantum Machine Learning
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) proficiency in implementing and applying classical machine learning algorithms, including classification, regression, gradient descent, and neural networks. (b) grasp the foundational principles of quantum computing, quantum states, qubits, and basic quantum operations.(c) advanced quantum algorithms and their applications in machine learning and computational tasks. (d) gain practical experience in implementing quantum algorithms and simulating quantum processes.
Course Description	This course offers a comprehensive exploration of machine learning (ML) and quantum computing (QC) principles, preparing students to navigate the intersection of classical and quantum computational paradigms. Students will master classical ML techniques including classification, regression, neural networks, and optimization methods like gradient descent. In the quantum computing segment, foundational concepts such as quantum states, qubits, and basic quantum operations (e.g., Hadamard gates) will be covered, alongside encoding classical data on quantum systems and implementing basic quantum algorithms. Advanced topics include variational quantum algorithms, quantum support vector machines, the HHL algorithm for linear systems, and quantum neural networks. Through lectures, practical exercises using quantum programming frameworks, and real-world applications, students will develop a dual proficiency in classical ML and quantum computing, equipping them for roles in research, development, or applications across industries leveraging emerging quantum technologies.
Course Outline	Overview of Machine Learning, Quantum Circuit, Variational quantum algorithm, Quantum Neural Network
Learning Outcome	• Understanding of Machine Learning and Quantum Computing Fundamentals. • Apply the concept of feature vectors, encode data in Quantum computing. • Analysis of Variational quantum algorithms to solve complex problems. • Implementation and analysis of advanced quantum machine learning algorithms.

Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term
--------------------------	---

Textbooks:

- Nielsen, M.A. and Chuang, I.L., 2010. *Quantum computation and quantum information*.
- Schuld, M. and Petruccione, F., 2021. *Machine learning with quantum computers* (Vol. 676). Berlin.
- Relevant research articles.

Reference books:

- Kasirajan, V., 2021. *Fundamentals of quantum computing*.
Quantum Machine Learning, Link: <http://sites.iiserpune.ac.in/~santh/course/QML/qml.html>

Course Number	CS6210
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Selective Topics in Generative AI
Learning Mode	Offline
Learning Objectives	To gain a comprehensive understanding of advanced AI architectures, particularly in the context of Generative AI. To develop proficiency in implementing and evaluating a variety of Generative AI techniques and models. To understand the principles and applications of Generative Pre-trained Transformers and other application-specific architectures. To explore and address ethical considerations and biases in Generative AI, emphasizing the importance of explainability. To engage with advanced topics and apply knowledge through hands-on projects.
Course Description	This course provides an in-depth exploration of Generative AI (GenAI), focusing on advanced AI architectures such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Generative Pre-trained Transformers (GPT). Students will learn about hybrid and emerging models, application-specific architectures, and the ethical considerations and biases in Generative AI. The course includes hands-on projects to design, implement, and evaluate sophisticated generative AI models, emphasizing innovation and practical problem-solving skills.
Course Outline	Introduction to advanced AI, overview of advanced AI architectures and Generative AI (GenAI) Generative Adversarial Network (GAN): various GAN architectures, DCGAN Advanced Variational AutoEncoder (VAE): hierarchical VAEs, Semi-supervised VAE Hybrid and emerging models: Energy-based models, diffusion models, autoregressive and flow-based models, attention mechanism in generative models Generative Pre-trained Transformers (GPT): architectural details and variations Advanced application-specific architecture: Models for Image-to-Text generation, Text-to-Image generation, Prompt engineering, Multimodality Ethical consideration and bias in Generative AI, Explainability Some advanced topics and project.

Learning Outcome	• Master various Generative AI architectures, including GANs, VAEs, and emerging models. • Demonstrate proficiency in implementing and evaluating advanced Generative AI techniques, such as hierarchical VAEs and energy-based models. • Understand the design principles and applications of Generative Pre-trained Transformers (GPT) and application-specific architectures. • Analyze and address ethical considerations and biases in Generative AI, emphasizing the importance of explainability. • Explore advanced topics in Generative AI and apply acquired knowledge through hands-on projects, fostering innovation and practical problem-solving skills.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Foster, D. (2022). Generative deep learning: : Teaching Machines to Paint, Write, Compose, and Play. " O'Reilly Media, Inc."
- Valle, R. (2019). Hands-On Generative Adversarial Networks with Keras: Your guide to implementing next-generation generative adversarial networks. Packt Publishing Ltd.

Research Papers and Articles from Journals such as JMLR, IEEE Transactions on Neural Networks and Learning Systems, etc., and Conference Proceedings from NeurIPS, ICML, and CVPR,etc.

Course Number	CS6215
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Quantum Cyber Security
Learning Mode	Physical
Learning Objectives	To have a clear understanding of Quantum technology and that brings on the security and privacy of communication and computation
Course Description	The course covers various effects that developing quantum technologies will have on cyber security.
Course Outline	Quantum information concepts: qubits, mixed states, operations, distance measures, quantum circuits, quantum algorithms (factoring, discrete logarithms, search). Classical Cryptography, encryption, authentication and key distribution protocols, Security analysis, quantum cryptography, quantum-key-distribution protocols, Security and implementation aspects. classical protocols and their security under quantum attacks, general quantum attacks (superposition attacks)
Learning Outcome	After completion of this course a student will have • Understanding of the quantum technologies. • Demonstrate their understanding of the power of quantum algorithms and be able to use the basic mathematical formalism for quantum information and quantum cryptography • Test whether a classical cryptosystem is secure against a range of quantum attacks • Use security notions for quantum information, such as encryption and authentication, in quantum cryptographic protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Quantum Computation and Quantum Information by Nielsen and Chuang
- Cryptography: Theory and Practice by D.R. Stinson

Course Number	CS6211
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Selected topics on Cryptography
Learning Mode	physical
Learning Objectives	To have a clear understanding of design and analysis of different advanced cryptographic protocols
Course Description	The course covers design and analysis of advanced cryptographic protocols
Course Outline	Mathematical Background: Modular Arithmetic, Finite Fields, Elliptic Curves over Finite Fields Crypto basics: SKE, PKE Zero-knowledge proofs, and protocols, Secret sharing, Commitment, Oblivious transfer, Secure multiparty computation, Homomorphic encryption, obfuscation, Post quantum cryptography
Learning Outcome	After completion of this course a student will have • Understanding and analysis of symmetric key cryptography • Understanding and analysis of asymmetric key cryptography • Designing and analysis of advanced cryptographic protocols • Understanding and analysis of post quantum cryptographic protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Doug Stinson, Cryptography: Theory and Practice, Chapman and Hall/CRC

Course Number	CS6216
Course Credit (L-T-P-C)	3-0-0-3
Course Title	High Performance Computing
Learning Mode	Offline
Learning Objectives	The course is designed to provide basic understanding of structure, and function of various building blocks of high performance Computing System. Students will be able to design various functional units and components and to identify the elements of modern GPUs and their impact on processor/GPU/TPU and parallel architecture design including memory
Course Description	Using a set of fundamental techniques and technologies, the high performance systems theme broadly explains how computing platforms work at various levels of abstraction, including both software and hardware. The course introduces HPS architecture with focus on parallel architectures
Course Outline	Computer types, Structure with basic computer components - instruction sets of some common CPUs/GPUs; Parallel Processing Concepts: a) Levels of parallelism (instruction, transaction, task, thread, memory, function) Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc) c) Architectures: N-wide superscalar architectures, multi-core, multi-threaded Parallel Programming with CUDA: a) Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: Fundamental Design Issues in Parallel Computing: a) Synchronization b) Scheduling, c) Job Allocation d) Job Partitioning, e) Dependency Analysis, f) Mapping Parallel Algorithms onto Parallel Architectures g) Performance Analysis of Parallel Algorithms Power-Aware Computing and Communication: a) Power-aware Processing Techniques Advanced Topics: (a) Petascale Computing, (b) Optics in Parallel Computing, (c) Quantum Computers, (d) Recent developments in Nanotechnology and its impact on HPC
Learning Outcome	The student will be able to : • Appreciate understanding of the HPC blocks, key terminology, and current industry trends in high performance computer architecture. • Understand parallel programming and evaluate and compare the architectural features of the state of the art high performance commodity hardware platforms. • Understand the processor (CPU/GPU/TPU) subsystem. • Employ concepts of the HPS memory subsystem and hierarchy

Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term
--------------------------	---

Text Books:

- "Highly Parallel Computing", by George S. Almasi and Alan Gottlieb
- "Advanced Computer Architecture: Parallelism, Scalability, Programmability", by Kai Hwang, McGraw Hill 1993
- "Parallel Computer Architecture: A hardware/Software Approach", by David Culler Jaswinder Pal Singh, Morgan Kaufmann, 1999.
- "Scalable Parallel Computing", by Kai Hwang, McGraw Hill 1998.
- "Principles and Practices on Interconnection Networks", by William James Dally and Brian Towles, Morgan Kauffman 2004.
- GPU Gems 3 --- by Hubert Nguyen (Chapter 29 to Chapter 41)
- Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, © 2003.
- Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, © 2007.

Course Number	RM6201
Course Credit (L-T-P-C)	3-1-0-4
Course Title	Research Methodology
Learning Mode	Lectures
Learning Objectives	The objective of the course is to train student about the modelling of scalar and multi-objective nonlinear programming problems and various classical and numerical optimization techniques and algorithms to solve these problems
Course Description	Advanced Optimization Techniques, as a subject for postgraduate and PhD students, provides the knowledge of various models of nonlinear optimization problems and different algorithms to solve such problems with its applications in various problems arising in economics, science and engineering.
Course Content	Module I (6 lecture hours) – Research method fundamentals: Definition, characteristics and types, basic research terminology, an overview of research method concepts, research methods vs. method methodology, role of information and communication technology (ICT) in research, Nature and scope of research, information based decision making and source of knowledge. The research process; basic approaches and terminologies used in research. Defining research problem and hypotheses framing to prepare a research plan. Module II (5 lecture hours) - Research problem visualization and conceptualization: Significance of literature survey in identification of a research problem from reliable sources and critical review, identifying technical gaps and contemporary challenges from literature review and research databases, development of working hypothesis, defining and formulating the research problems, problem selection, necessity of defining the problem and conceiving the solution approach and methods. Module III (5 lecture hours) - Research design and data analysis: Research design – basic principles, need of research design and data classification – primary and secondary, features of good design, important concepts relating to research design, observation and facts, validation methods, observation and collection of data, methods of data collection, sampling methods, data processing and analysis, hypothesis testing, generalization, analysis, reliability, interpretation and presentation. Module IV (16 lecture hours) - Qualitative and quantitative analysis: Qualitative Research Plan and designs, Meaning and types of Sampling, Tools of qualitative data Collection; observation depth Interview, focus group discussion, Data editing, processing & categorization, qualitative data analysis, Fundamentals of statistical methods, parametric and nonparametric techniques, test of significance, variables, conjecture, hypothesis, measurement, types of data and scales, sample and sampling techniques, probability and distributions, hypothesis testing, level of significance and confidence interval, t-test, ANOVA, correlation, regression analysis, error analysis, research data analysis and evaluation using software tools (e.g.: MS Excel, SPSS, Statistical, R, etc.). Module V (10 lecture hours) – Principled research: Ethics in research and Ethical dilemma, affiliation and conflict of interest; Publishing and sharing research, Plagiarism and its fallout (case studies), Internet

	research ethics, data protection and intellectual property rights (IPR) – patent survey, patentability, patent laws and IPR filing process.
Learning Outcome	On successful completion of the course, students should be able to: 1. Understand the terminology and basic concepts of various kinds of nonlinear optimization problems. 2. Develop the understanding about different solution methods to solve nonlinear Programming problems. 3. Apply and differentiate the need and importance of various algorithms to solve scalar and multi-objective optimization problems. 4. Employ programming languages like MATLAB/Python to solve nonlinear programming problems. 5. Model and solve several problems arising in science and engineering as a nonlinear optimization problem.
Assessment Method	Quiz /Assignment/ Project / MSE / ESE

Textbooks & Reference Books:

1. C. R. Kothari, Research methodology: Methods and Techniques, 3rd Edn., New age International 2014.
2. Mark N K. Saunders, Adrian Thornhill, Phkip Lewis, “Research Methods for Studies, 3/c Pearson Education, 2010.
3. K.N. Krishnaswamy, apa iyer, siva kumar, m. Mathirajan, “Management Research Methodology”, Pearson Education, 2010.
4. Ranjit Kumar; “Research Methodology: A Step by Step Guide for Beginners; 2/e; Pearson Education, 2010.
5. Suresh C. Sinha, Anil K. Dhiman, ess ess, 2006 “Research Methodology” Panner Selvam.R. “Research Methodology”, Prentice Hall of India, New Delhi, 2004.
6. C.G. Thomas, Research methodology and scientific writing, Ane books, Delhi, 2015.
7. H. J. Ader and G. J. Mellenbergh, Research Methodology in the Social, Behavioural and Life Sciences Designs, Models and Methods, 3rd Edn., Sage Publications, London, 2000.

SEMESTER IX						
Sl. No.	Subject Code	Subject Name	L	T	P	C
1.	CS61XX	M. Tech. Elective – II	3	0	0	3
2.	CS61XX	M. Tech. Elective – III	3	0	0	3
3.	CS61XX	M. Tech. Elective – IV	3	0	0	3
4.	CS6190	Project – III**	0	0	16	8
TOTAL			9	0	16	17

M. Tech. Elective – II						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6101	Advanced Blockchain Technology	3	0	0	3
2.	CS6102	Advanced Cyber Security	3	0	0	3
3.	CS6104	Formal Methods in Program Analysis and Verification	3	0	0	3
4.	CS6109	Reinforcement Learning	3	0	0	3

Course Number	CS6101
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Blockchain Technology
Learning Mode	Offline
Learning Objectives	The objective of this course is to cover a number of popular blockchain platforms and smart contract language paradigms. This course makes the learners familiar with various (a) research challenges, such as interoperability, scalability, security vulnerabilities, functional/non- functional correctness proof, etc., and their possible solutions, (b) synergizing machine Learning and blockchain, and (c) development of secure blockchain-based decentralized applications using Ethereum and Hyperledger.
Course Description	This course will start with a quick introductory background of blockchain technology and its working principle. The primarily focus of this course is to provide a detailed information about the state-of-the-art blockchain platforms and their supported smart contract languages. In particular, syntax, semantics, and paradigms of various smart contract languages will be discussed. In this perspective, blockchain-oriented software development life cycle and decentralized application development will be discussed. Following this, the course will cover two important directions: addressing various research challenges in blockchain and AI/machine learning for blockchain (and vice-versa).
Course Outline	Introduction to Blockchain Technology: A Quick Tour Different Blockchain Platforms and Smart Contract Languages: Bitcoin, Ethereum, Hyperledger, Solidity, GoLang. Consensus Mechanisms: PoW Vs. PoS, Alternative Consensus Synergizing Machine Learning and Blockchain: Transaction Analysis, Smart Contract Code Analysis, AI-driven Blockchain Applications, Blockchain for AI, Decentralized Learning. Research Challenges in Blockchain: Scalability, Interoperability, Security, Privacy, Decentralized Identity, Smart Contract Vulnerabilities and Detection, Real case studies on developing DApps, Metaverse, Some ongoing relevant research topics.
Learning Outcome	• Gain proficiency in blockchain technology and software engineering of developing decentralized applications. • An overview of the state-of-the-art blockchain platforms and their supported smart contract languages. • Know about the paradigms of various smart contract languages. • Understand how AI/machine learning brings benefits to blockchain technology and vice-versa. • Identify various research challenges and opportunities, such as scalability, interoperability

Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term
--------------------------	---

Suggested Readings:

- Don Tapscott and Alex Tapscott: Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World, Portfolio (May 2016).
- Andreas M. Antonopoulos, Gavin Wood: Mastering Ethereum: Building Smart Contracts and Dapps, O'Reilly, first edition (Dec 2018).
- Nitin Gaur, Luc Desrosiers, Venkatraman Ramakrishna, Petr Novotny, Salman A. Baset, Anthony O'Dowd: Hands-On Blockchain with Hyperledger, Packt Publishing, first edition (June 2018).
- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction, Princeton University Press (2016).
- Stijn Van Hiejte: Blockchain Platforms: A Look at the Underbelly of Distributed Platforms, Morgan & Claypool Publishers, first edition (July 2020)
- Relevant Research Papers and Study Materials available online.

Course Number	CS6102
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Cyber Security
Learning Mode	Offline
Learning Objectives	To have a clear understanding of security and privacy issues in various aspects of computing, including: Programs, Operating systems, Networks, Web Applications
Course Description	The course covers. security and privacy issues in various aspects of computing, including: Programs, Operating systems, Networks, Web Applications
Course Outline	Introduction to Computer Security and Privacy: security and privacy; types of threats and attacks; methods of defense Basics of cryptography, Authentication & key agreement, Authorization and access control Program Security: nonmalicious program errors; vulnerabilities in code, Secure programs; malicious code; Malware detection Internet security: IPSEC, TLS, SSh, Email security Wireless security: WEP, WPA, Bluetooth security, Web Security: XSS attack, CSRF attack, SQL Injection, DoS attack & defense
Learning Outcome	After completion of this course a student will have • Understanding of security issues in computer and networks, • Understanding and analysis of internet security protocols • Understanding and analysis of web security protocols
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Computer Security: Principles and Practice: Dr. William Stallings and Lawrie Brown, Pearson
- O'Reilly Web Application Security by Andrew Hoffman

Course Number	CS6104
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Formal Methods in Program Analysis and Verification
Learning Mode	Offline
Learning Objectives	Formal methods are mathematically rigorous techniques to facilitate in building high-confidence critical systems with stringent quality requirements, such as safety and security. It provides a systematic guidance for specification, development, and verification of software and hardware systems. Examples for such systems are banking software, avionics software, medical device software, software used to control industrial plants/cars, etc. This course will provide necessary background on formal methods and their role in software engineering practice. A range of formal methods will be introduced along with practical case studies of their use. Students will learn how these methods can be used to build reliable software, hardware, and security protocols.
Course Description	This course will start with the fundamentals of set theory, relation and function, lattice theory, propositional and predicate logic, and proof techniques. In order to demonstrate how to analyze and verify a software system, this course will discuss the following three formal approaches using suitable examples: (a) Abstract Interpretation Theory, (b) Temporal Logic and Model Checking, and (c) Deductive Reasoning. In this context, formalism of syntax and semantics of programming languages will be explained considering a simple imperative language WHILE. All these approaches will be illustrated using real-life examples, such as microwave oven, mutual exclusion problem, etc.
Course Outline	Introduction: Introduction to critical systems, Introduction to formal methods and its role, Dependability, Testing Vs. Verification Formal Syntax and Semantics: the WHILE Language, Syntax Vs. Semantics, Formal Program Semantics - Operational, Denotational, Axiomatic Formal Program Analysis: Program Slicing, Dataflow Analysis, Fixpoint Algorithm, Abstract Interpretation Framework Formal Program Verification: Deductive Reasoning; Predicate Abstraction and CEGAR, Temporal Logic and Model Checking, Role of some other formal methods in software engineering New Research Directions: Recent trends on the application of formal methods in Machine Learning and Blockchain Tools: Introduction to various state-of-the-art Analyzers and Verifiers (e.g., NuSMV, UPPAAL, SPIN, ASTREE, CBMC, etc.)
Learning Outcome	• Gain proficiency in formal methods and their role in critical systems. • Understanding formal tools and techniques for analysis and verification of software source codes. • Learning how to define semantics of a software formally, and how to abstract its semantics at different levels of precision in order to capture its run-time behavioral properties of interests. • Learning temporal logic to express system's time-varying behaviors. • Applying automatic software verification tools based on model checking and deductive reasoning. • Hands-on experience with NuSMV, Uppaal, Z3 SMT solver, etc.

	• Application of formal methods in cutting edge research domains including Robotics, IoT, Blockchain Smart Contracts.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Flemming Nielson, Hanne R. Nielson, Chris Hankin. Principles of Program Analysis, Springer, 1999.
- Edmund M. Clarke, Orna Grumberg, Doron A. Peled. Model Checking, The MIT Press, 1999.
- Glynn Winskel. The formal semantics of programming languages: an introduction, The MIT Press, 1993.
- José Bacelar Almeida, Maria João Frade, Jorge Sousa Pinto, Simão Melo de Sousa. Rigorous Software Development: An Introduction to Program Verification. Springer- Verlag London, 2011
- Recent Research Papers relevant to the course.

Course Number	CS6109
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Reinforcement Learning
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the foundational concepts and mathematical frameworks of reinforcement learning. (b) Gain proficiency in key reinforcement learning algorithms, including dynamic programming, Monte Carlo methods, and temporal-difference learning (c) Apply deep reinforcement learning techniques to solve complex problems using methods such as deep Q-networks and policy gradient algorithms. (d) Explore recent advancements and applications of reinforcement learning, including multi-agent systems and ethical considerations.
Course Description	This specialized course on reinforcement learning aims to give students a deep understanding of the algorithms and methodologies used to train agents to make decisions through trial and error. Students will learn to develop and implement reinforcement learning models by focusing on foundational theories and practical applications. Students will explore key concepts such as Markov decision processes, policy gradients, Q-learning, and deep reinforcement learning through a mix of theoretical lectures, coding exercises, and project-based learning. Upon completion, students will be equipped to design and apply reinforcement learning solutions to complex problems in fields such as robotics, game development, and autonomous systems, enhancing their expertise in this dynamic area of artificial intelligence.
Course Outline	Foundations: Basics of machine learning and reinforcement learning (RL) terminology. Probability Concepts: Axioms of probability, random variables, distributions, and correlation. Markov Decision Process: Introduction to MDPs, Markov property, and Bellman equations. State and Action Value Functions: Concepts of MDP, state, and action value functions. Tabular Methods and Q-networks: Dynamic programming, Monte Carlo, TD learning, and deep Q-networks. Policy Optimization: Policy-based methods, REINFORCE algorithm, and actor-critic methods. Recent Advances and Applications: Meta-learning, multi-agent RL, ethics in RL, and real-world applications.
Learning Outcome	• Mastery of fundamental principles and mathematical frameworks of reinforcement learning. • Proficiency in implementing key reinforcement learning algorithms and techniques. • Ability to apply deep reinforcement learning methods to complex, real-world problems. • Understanding of recent advancements in reinforcement learning and their ethical implications.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto, The MIT Press (1 January 1998).
- Deep Reinforcement Learning Hands-On by Maxim Lapan, Packt Publishing Limited (21 June 2018).
- Algorithms for Reinforcement Learning by Csaba Szepesvari, Morgan and Claypool Publishers (2010)
- Deep Reinforcement Learning: Fundamentals, Research and Applications by Hao Dong, Springer Verlag (2020)

M. Tech. Elective - III						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6106	Advanced Cloud Computing	3	0	0	3
2.	CS6107	Advanced Edge Computing	3	0	0	3
3.	CS6113	Cyber Physical Systems	3	0	0	3

Course Number	CS6106
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Cloud Computing
Learning Mode	Offline
Learning Objectives	This course aims to help the students understand (a) how and why cloud systems work and the cloud technologies that manifest these concepts, such as those from Amazon AWS and Microsoft Azure; (b) distributed systems concepts like virtualisation, data parallelism, CAP theorem, and performance analysis at scale; (c) Big Data programming patterns such as Map-Reduce (Hadoop), Vertex-centric graphs (Giraph), Continuous Dataflows (Storm), and NoSQL storage systems to build Cloud applications; (d) Cloud native computing and micro-services.
Course Description	This course provides an in-depth understanding of cloud computing, virtualisation, and distributed systems. It covers foundational concepts, advanced techniques, and real-world applications. Students will explore various aspects of cloud infrastructure, virtualisation technologies, distributed algorithms, and cloud-native computing. By the end of the course, students will be equipped with the knowledge and skills to design, implement, and manage cloud-based solutions and distributed systems effectively.
Course Outline	Cloud computing features and categories. Virtualization: Virtualization Models, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para virtualization, Device virtualization. Virtual Machine: Live VM Migration Stages, Virtual Machine Migration for Enterprise Data Centers, Data Center Workloads, Provisioning methods, Resource provisioning. Geo-distributed Clouds: Server Virtualization, Network Virtualization, Approaches for Networking of VMs: Hardware approach: Single-root I/O virtualization (SR-IOV), Software approach: Open vSwitch, Mininet and its applications. Software Defined Network for Multi-tenant Data Centers: Network virtualization, Case Study: VL2, NVP Geo-distributed Cloud Data Centers: Inter-Data Center Networking, Data center interconnection techniques: MPLS, Google's B4 and Microsoft's

	Swan. Leader Election algorithms in Cloud. Google's Chubby and Apache Zookeeper. Time and Clock Synchronization in Cloud Data Centers, Datacenter time protocol (DTP. Consensus, Paxos and Recovery in Clouds. Cloud Storage: Key-value stores/NoSQL, Design of Apache Cassandra, HBase. Peer to Peer Systems in Cloud Computing. Cloud application: MapReduce Examples. Advances in Cloud Computing with decentralization and Edge Computing.
Learning Outcome	• Cloud Computing as a Distributed Systems: Explain and contrast the role of Cloud computing within this space. • Cloud Virtualization, Abstractions and Enabling Technologies: Explain virtualisation and their role in elastic computing. Characterise the distinctions between Infrastructure, Platform and Software as a Service (IaaS, PaaS, SaaS) abstractions, and Public and Private Clouds, and analyse their advantages and disadvantages. • Programming Patterns for "Big Data" Applications on Cloud: Demonstrate using Map-Reduce, Vertex-Centric and Continuous Dataflow programming models. • Application Execution Models on Clouds: Compare synchronous and asynchronous execution patterns. Design and implement Cloud applications that can scale up on a VM and out across multiple VMs. Illustrate the use of NoSQL Cloud storage for information storage. • Performance, scalability and consistency on Clouds: Explain the distinctions between Consistency, Availability and Partitioning (CAP theorem), and discuss the types of Cloud applications that exhibit these features.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Distributed and Cloud Computing From Parallel Processing to the Internet of Things; Kai Hwang, Jack Dongarra, Geoffrey Fox Publisher: Morgan Kaufmann, Elsevier, 2013.
- Cloud Computing: Principles and Paradigms; Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski Publisher: Wiley, 2011.
- Distributed Algorithms Nancy Lynch Publisher: Morgan Kaufmann, Elsevier, 1996.
- Cloud Computing Bible Barrie Sosinsky Publisher: Wiley, 2011.
- Cloud Computing: Principles, Systems and Applications, Nikos Antonopoulos, Lee Gillam Publisher: Springer, 2012.

Course Number	CS6107
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Edge Computing
Learning Mode	Offline
Learning Objectives	Upon successful completion of this course, students will be able to: (a) understand the fundamental concepts and limitations of cloud computing and identify the advantages of edge computing; (b) describe various edge computing architectures and differentiate them from traditional cloud models; (c) comprehend the principles of distributed systems as they apply to edge computing environments; (d) explore the functionalities of edge data centers and lightweight edge clouds; (e) deploy and manage containerized applications using Docker and Kubernetes in edge computing contexts; and (f) implement and evaluate edge storage systems and end-to-end edge pipelines utilising MQTT and Kafka, as well as investigate advanced edge computing technologies for real-world applications.
Course Description	This course delves into the emerging field of edge computing, providing a comprehensive understanding of its architectures, systems, and technologies. Students will explore the limitations of traditional cloud computing and learn about the advantages and applications of edge computing. The course covers key concepts in distributed systems, edge data centers, and lightweight edge clouds and includes hands-on experience with Docker, Kubernetes, and edge storage systems. Additionally, students will gain insights into end-to-end edge pipelines using MQTT and Kafka and examine advanced edge computing technologies. By the end of the course, students will be equipped with the knowledge and skills to design, implement, and manage edge computing solutions.
Course Outline	Cloud Computing Basics.Edge Computing basics. Edge Computing Use-Cases, Benefits. Different Types of Edge. Edge Deployment Modes. Edge Computing in 5G, Multi-access Edge Computing (MEC) and Mobile Edge Computing.
Learning Outcome	• Critically evaluate advanced edge computing architectures, such as hierarchical, mesh, and hybrid models, considering their suitability for specific use cases and environments. • Analyses emerging technologies and trends in advanced edge computing, such as edge AI, blockchain, and serverless computing, and assess their potential impact. • Design and implement innovative edge computing solutions that leverage advanced techniques, such as federated learning, edge caching, and dynamic resource allocation. • Evaluate the performance and scalability of advanced edge computing systems using benchmarking, simulation, and experimentation.

	• Investigate advanced techniques for ensuring security, privacy, and data integrity in edge computing ecosystems, such as secure enclaves, encryption, and access control mechanisms. • Explore specialised applications of advanced edge computing in domains such as healthcare, smart cities, and autonomous systems, analysing their requirements and challenges.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya (Editor), Satish Narayana Srirama (Editor), Wiley, 2019
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
- Cloud and Distributed Computing: Algorithms and Systems, Rajiv Misra, Yashwant Patel, Wiley 2020.
- Besides these books, we will provide Journal papers as references.

Course number	CS6113
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Cyber Physical Systems
Learning Mode	offline
Learning Objectives	To learn how to model and design the joint dynamics of software, networks, and physical processes., To develop the skills to realize embedded systems that are safe, reliable, and efficient in their use of resources., To learn to think critically about technologies that are available for achieving such joint dynamics.
Course Description	This course will provide an overview of modeling, building, analyzing methods for cyber physical systems.
Course Outline	Models of computation: finite state machines, threads, ordinary differential equations, hybrid systems, actors, discrete-events, data flow Basic analysis, control, and systems simulation: Bisimulations, reachability analysis, controller synthesis, approximating continuous-time systems. Interfacing with the physical world: sensor/actuator modeling and calibration, concurrency in dealing with multiple real-time streams, handling numerical imprecision in software Mapping to embedded platforms: real-time operating systems, execution time analysis, scheduling, concurrency Distributed embedded systems: Protocol design, predictable networking, security
Learning Outcome	• Basic understanding of cyber physical systems • To develop the skills to realize embedded systems that are safe, reliable, and efficient in their use of resources., • To learn to think critically about technologies that are available for achieving such joint dynamics.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- Introduction to Embedded Systems - A Cyber-Physical Systems Approach, Second Edition, by E. A. Lee and S. A. Seshia, 2015
- Vahid, F. and T. Givargis (2010). Programming Embedded Systems - An Introduction to Time-Oriented Programming, UniWorld Publishing.
- Schaumont, P. R. (2010). A Practical Introduction to Hardware/Software Codesign, Springer.
- E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*, Addison-Wesley, 2003.

M. Tech. Elective - IV						
Sl. No.	Subject Code	Subject	L	T	P	C
1.	CS6103	Advanced Pattern Recognition	3	0	0	3
2.	CS6108	Advanced Computational Data Analysis	3	0	0	3
3.	CS6110	Advanced Graph Machine Learning	3	0	0	3
4.	CS6111	Advanced Time Series Analysis	3	0	0	3

Course Number	CS6103
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Pattern Recognition
Learning Mode	Offline
Learning Objectives	This course aims to help the students (a) Understand the advanced topics of pattern recognition, including classification and clustering methods. (b) To understand the advanced topics of feature selection, multi-label classification. (c) Apply advanced pattern recognition algorithms to practical applications in image processing, speech recognition, and data mining.
Course Description	This course on advanced pattern recognition aims to equip students with the advanced topics of classification, clustering, and feature selection. By focusing on advanced topics, students will develop the ability to implement and evaluate various pattern recognition algorithms. Students will enhance their understanding of advanced topics of classification, clustering, statistical methods, and data preprocessing techniques through interactive lectures, exercises, and projects. Upon completion, students will be proficient in designing and applying advanced pattern recognition systems for applications such as image processing, text mining, speech recognition, and data mining, thereby enhancing their analytical and problem-solving capabilities in diverse domains.
Course Outline	Introduction and motivation of advanced pattern recognition Modern Classification Methods, Random fields, Pattern recognition based on multidimensional models Contextual classification, Hidden Markov models, Multi-classifier systems Advanced parameter estimation methods, Advanced Unsupervised classification, Modern methods of feature selection. Data normalization and invariants, Benchmarking. Analysis and synthesis of image information. Applications of pattern recognition in Text Processing and Healthcare.
Learning Outcome	• Mastery of advanced concepts in pattern recognition. • In-depth understanding of various advanced algorithms across different pattern recognition paradigms. • Comprehensive knowledge of advanced aspects of classification, clustering, feature selection, feature extraction, and projection techniques. • Ability to apply advanced pattern recognition algorithms to real-world projects
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Reading:

- P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, 1982.
- R. Duda and P. Hart and D.G. Stork, Pattern Classification, J. Wiley, 2001.
- Webb, Statistical Pattern Recognition, J. Wiley, 2002.
- S.Theodoridis, K.Koutroumbas, Pattern Recognition, Elsevier, 2003.
- S. Z. Li, Markov Random Field Modeling in Image Analysis, Springer, 2009.

Research papers will be provided on various topics

Course Number	CS6108
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Computational Data Analysis
Learning Mode	Offline
Learning Objective	In this subject, the students will be trained with the knowledge of various advanced data analytics techniques encountered in real life.
Course Description	Current Physical systems/devices are highly complex and fast and operate with high data acquisition and generation capabilities. Data generated from such systems require advanced level of analytics for apprehension and further usage. This course aims to give a broad understanding what is advanced level data analytics techniques and how they play a critical role in analysing modern day physical systems acquired data.
Course Outline	Introduction, Operation of physical systems and data generation, Complexity, Drawbacks and Challenges in data generation from physical devices. Requirement of advanced data analytics. Foundations of advanced data analytics principles, mathematical models, probabilistic models, optimization models, deep learning and machine learning models. Role of advanced data analytics in data apprehension and compression, curve-based approximation techniques, interpolation techniques, machine learning models for data interpretation. Statistical models to advanced data analytics, data analytics for 2D and 3D data processing and data manipulation, application of advanced data analytics to real life cases, problem solving.
Learning Outcome	• Gain understanding on data generation systems and the role of advanced data analytics. • Apply the Mathematical models of advanced data analytics to real time • Understand the utilities of statistical models and ML models for advanced data analytics.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- Signal Processing: A Mathematical Approach, Charles L. Byrne, Second Edition, Chapman & Hall, 2014.
- Digital Functions and Data Reconstruction: Digital-Discrete Methods, Li M Chen, Springer, 2013.

- Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, Bernhard Mehlig, Cambridge University Press, 2021
- Signal Processing and Machine Learning with Applications, Michael M. Richter, Sheuli Paul, Veton Këpuska, Marius Silaghi, Springer Cham, 2022
- Data Compression: The Complete Reference, David Solomon, 4th Edition, Springer, 2007

Course Number	CS6110
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Graph Machine Learning
Learning Mode	Offline
Learning Objectives	Several real world systems can be represented as a network of entities that are connected to each other through some relations. Often the number of entities is immensely large, thus forming a very large network. Typical examples of such large networks include network of entities in knowledge graphs, co-occurrence graph of the keywords in natural languages, interaction graph of users in social networks, protein-protein interaction graphs and the network of routers in Internet to name a few. Study of these networks is often needed for relational learning tasks, as well as for developing frameworks for representing the intrinsic structure of the data. This course will mainly deal with both the traditional as well as current state of the art machine learning techniques to be applied on Graphs for different downstream tasks.
Course Description	The course will provide knowledge on the representation and statistical descriptions of large networks, along with traditional machine learning and deep learning techniques applied on graphs. Several use cases of Graph Machine Learning across different domains including Natural Language Processing, Social Network Analysis and Computational Biology would be studied.
Course Outline	Introduction and background knowledge of graphs; Network Measures and Metrics; Spectral Analysis of Graphs and its applications; Random Networks; Properties of Random Networks; Overview of machine learning applications on graphs; Feature based learning on graphs, Shallow embedding and deep Learning techniques for generating node and graph representations – Graph Neural Networks, Graph Attention Networks, Graph Transformers; Graph Neural Networks Pretraining techniques; Generative models for graphs; Models for scale-free and small-world networks; Temporal networks, Modeling temporal networks
Learning Outcome	Course training via lectures & tutorial sessions to • Represent and analyze the structure of graphs • Discover recurring and significant patterns of interconnections in your data with network motifs and community structure. • Gain Knowledge on traditional machine learning techniques applied on graphs • Leverage graph-structured data to make better predictions using graph neural networks

	• Understand the problems in dealing with large graphs for machine learning tasks and learn how to improvise • Analyze temporal and dynamic graphs Scaling neural networks with generative models for graphs.
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Suggested Readings:

- M.E.J. Newman, Networks - An introduction , Oxford Univ Press, 2010.
- Yao Ma and Jilian Tang, Deep Learning on Graphs, Cambridge University Press, 2021
- Goyal, Palash and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey.” *Knowl.-Based Syst.* 151 (2018): 78-94.

Course Number	CS6111
Course Credit (L-T-P-C)	3-0-0-3
Course Title	Advanced Time Series Analysis
Learning Mode	Offline
Learning Objectives	• This course on advance time series will teach both the fundamental concepts time series analysis, as well as recent trends in time series analysis. • Students will learn to design successful time series data applications with sequential Neural Networks. • Deploy Nonlinear Auto-regressive Network with Exogenous Inputs • Adapt Deep Neural Networks for Time Series Forecasting and classification
Course Description	This course provides advanced concepts in time series analysis including some fundamentals of time series, data pre-processing, feature selection, Variety of modeling techniques, Anomaly Detection in Time Series and forecasting financial series using statistical, econometric, machine learning, and deep learning approaches and Practical Applications and Deployment of models.
Course Outline	Introduction to classical time series methods, time series Virtualization Univariate Stationary Processes; Granger Causality; Vector Autoregressive Processes Nonstationary Processes; Cointegration; Cointegration in Single Equation Models: Representation, Estimation and Testing. Applied Predictive Modeling Techniques; Autoregressive Conditional Heteroskedasticity. Finance and Algorithmic trading: Machine Learning and Deep Learning in Stock Price Prediction Machine Learning, Deep Learned Time series Analysis, Risk and Portfolio Management Practical Applications and Deployment of models; applications of convolutional neural network (CNN) and long-and-short-term memory (LSTM) network architectures; designing predictive models for financial time series data Stock Price Prediction using Deep Learning and Natural Language Processing

Learning Outcome	At the end of the course, students will have achieved the following learning objectives. • problems relating to obtaining, cleaning, simulating, and storing time series data. • Variety of modeling techniques that can be used for recent time series analysis • techniques of financial time series analysis and forecasting financial series using statistical, econometric, machine learning, and deep learning approaches. • Apply more recently developed methods, such as machine learning and neural network, to time series data, highlighting the challenges of data processing and data layout when time series data is used for fitting models
Assessment Method	Internal(Quiz/Assignment/Project), Mid-Term, End-Term

Textbooks:

- Kirchgässner, Gebhard, Jürgen Wolters, and Uwe Hassler. *Introduction to modern time series analysis*. Springer Science & Business Media, 2012.
- Lazzeri, F. (2020). *Machine learning for time series forecasting with Python*. John Wiley & Sons.
- Jaydip, Sen, and Mehtab Sidra. *Machine Learning in the Analysis and Forecasting of Financial Time Series*. 2022.