M. TECH IN COMPUTER SCIENCE AND ENGINEERING

I SEMESTER (2023– 2025 BATCH)

Sl. No.	Course Code	Course Title	Hours / week				Credits	Tools/Languages	Course Type
			L	T	P	S	C		
1.	UE23CS641A	Computer Systems for Programmers	4	0	0	4	4	C/C++/Python/Java based IDEs and Instruction Set Simulators for Pipelined Multicore processors.	CC-Independent
2.	UE23CS642A	Advanced Data Structures	4	0	0	4	4	C, C++, Java or Python Programming Languages.	CC-Independent
3.	UE23CS643A	Scalable Computing	4	0	0	4	4	AWS, PostMan, Flask, Docker, Hadoop.	CC-Independent
4.	UE23CS644A	Stochastic models and Machine Learning	4	0	0	4	4	Pythorch.	CC-Independent
5.	UE23CS645A	Cyber Security Essentials	4	0	0	4	4	Hashcat, Wireshark, Python(Scapy).	CC-Independent
6.	UE23CS621A	Special Topic - I	2/4/4			2	2		ST
Total			**22**	**0**	**0**	**22**	**22**		

SPECIALIZATION FOR CORE SUBJECTS

Sl. No.	SPECIALIZATION	COURSE CODE	COURSE TITLE
1.	Cloud Computing	UE23CS643A	Scalable Computing
2.	Machine Intelligence	UE23CS644A	Stochastic models and Machine Learning
3.	Cyber Security	UE23CS645A	Cyber Security Essentials

II SEMESTER (BATCH : 2023-2025)

Sl. No.	Course Code	Course Title	Hours / week				Credits	Tools/Languages	Course Type
			L	T	P	S	C		
1.	UE23CS641B	Topics in Advanced Algorithms *	4	0	0	4	4	C /C++ Programming Language.	CC-Independent
2.	UE23CS642B	Distributed Computing	4	0	0	4	4	Wireshark / C compiler with Network Libraries.	CC-Independent
3.	UE23S643BAX	Elective –I	4	0	0	4	4		EC
4.	UE23CS644BBX	Elective –II	4	0	0	4	4		EC
5.	UE23CS645BCX	Elective –III	4	0	0	4	4		EC
6.	UE23CS621B	Special Topics - II	2/4/4			2	2		ST
Total			**22**	**0**	**0**	**22**	**22**		-
Sl. No.	**Elective – I**								
1.	UE23CS643BA1	Million Way Parallelism	4	0	0	4	4	OpenMP, CUDA	EC-Independent
2.	UE23CS643BA2	Speech and Natural Language Processing	4	0	0	4	4	CoreNLP, Natural Language Toolkit (NLTK), TextBlob,Gensim, SpaCy, PyTorch-NLP , OpenNLP	EC-Independent

3.	UE23CS643BA3	Topics in Computer and Network Security	4	0	0	4	4	SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.	EC-Independent
Elective – II									
1.	UE23CS644BB1	Advanced Cloud Computing	4	0	0	4	4	AWS, Kubernetes ,Terraform.	EC-Independent
2.	UE23CS644BB2	Virtual Reality and its applications	4	0	0	4	4	C++, Unity3D, Blender.	EC-Independent
3.	UE23CS644BB3	Software Security[%]	4	0	0	4	4	SEED Labs VM, Scapy, Burp Suite, Metasploit.	EC-Independent
Elective – III									
1.	UE23CS645BC1	Advanced Big Data Analytics	4	0	0	4	4	Apache, Hadoop Hive, Spark, Solr, R, Google Cloud Platform, IBM Watson.	EC-Independent
2.	UE23CS645BC2	Deep Learning Theory and Practices	4	0	0	4	4	TensorFlow 1.15, Keras 2.3.1, Python 3.7	EC-Independent
3	UE23CS645BC3	Cryptography	4	0	0	4	4	Seed virtual machine environment.	EC-Independent
Desirable Knowledge		*UE23CS642A, [%]UE23CS645A							

ELECTIVES TO BE OPTED FOR SPECIALIZATION				
Sl. No.	**SPECIALIZATION**	**ELECTIVE – I**	**ELECTIVE – II**	**ELECTIVE – III**
1.	Cloud Computing	UE23CS643BA1	UE23CS644BB1	UE23CS645BC1
2.	Machine Intelligence	UE23CS643BA2	UE23CS644BB2	UE23CS645BC2
3.	Cyber Security	UE23CS643BA3	UE23CS644BB3	UE23CS645BC3

UE22CS641A: Computer Systems for Programmers (4-0-0-4-4)

This course is an in-depth analysis of the programming interface to the hardware subsystems/mapping of contemporary processor architectures-based computing system. This course provides an end-to-end picture in sufficient advanced detail of Instruction Set Architecture, pipelining and program construct mapping to memory hierarchy.

Course Objectives:

- Introduce concepts of basic processor architecture and its design.
- Understand the concepts of pipeline architecture and hazards.
- Understand the concepts of linking, from traditional static linking, to dynamic linking of shared libraries at load time, to dynamic linking of shared libraries at run time.
- Describes the memory hierarchy, cache memory and its optimization and various benefits of a virtual memory system.

Course Outcomes:

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

- Trace the execution of a program with respect to modern processor architecture fundamentals, Caching.
- Design and implement Instruction Set Simulators for novel processor architectures.
- Write and debug complex programs.
- Utilize modern Processor architectures with complete understanding of contemporary programming language features and gain extensive knowledge on Virtual Memory implementations.

Course Content:

Unit 1: Introduction to computer systems: Compilation system, Processor functioning, Caches, Storage devices, Networks, information storage, Processes, Threads and Concurrency, Parallelism, Number representations. Machine level representation of programs.

14 Hours

Unit 2: Processor architecture: Instruction set architecture logic design, Clocking, Pipelining, Data hazards, Exception handling, Simulators.

14 Hours

Unit 3: Linking: Compiler drivers, Static linking, Object files formats, relocatable object files, Symbol tables, Symbol resolution, Relocation, Dynamic linking, Shared libraries, Loading executable object files, Position independent code.

14 Hours

Unit 4: Memory Hierarchy: Storage technologies, locality of reference, cache memories. Impact of caches on program performance**. Virtual Memory**: Page tables, Locality, Address translation, memory mapping, Dynamic memory allocation, Garbage collection, and Common memory related bugs.

14 Hours

Tools/Languages: C/C++/Python/Java based IDEs and Instruction Set Simulators for Pipelined Multicore processors.

Text Book(s):

1: “Computer Systems: A Programmer’s Perspective”, Randal Bryant and David O’ Halloran, Prentice Hall, 3rd Edition, 2011.

Reference Book(s):

1. Computer Architecture: A Quantitative Approach”, Hennessey and Patterson, MK publishers, 6th Edition, 2011.

UE23CS642A - Advanced Data Structures (4-0-0-4-4)

Data Structures play a central role in modern computer science. Data Structures are the essential building blocks in building efficient algorithms. The course on Advanced Data Structures includes review of basic data structures like stacks, queues and lists. The course also includes study of complex data structures, analysis of complex data structures and their applications.

Course Objectives:

- Appreciate the impact of Data Structures on Algorithms, Program Design and Program Performance.
- Understand and apply Amortized Analysis on Data Structures, including Binary search trees, Mergeable Heaps and Dynamic Tables.
- Analyze the applications of static and dynamic Search Trees and Heaps.
- Understand advanced ADTs with Interface and Implementation separation.
- Understand State Space search and Spatial Data Structures with R-Trees.

Course Outcomes:

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

- Demonstrate the notion of Abstract Data Types (ADT) & Recursive accesses on them.
- Illustrate the relation between Data Structure operations and Amortized Complexity analysis.
- How to Analyze Iterated Lists and variations thereof and demonstrate tree data structures and how to balance them, for specific access needs.
- Implement different Heaps and B-Tree variations.
- Analyze and implement spatial data implementations and their search.

Course Content:

Unit 1: Complexity, Amortized Analysis, Abstract Data Types (ADT), List

Asymptotic Complexity Notations, Amortized Complexity Analysis of Stacks, Binary Counters, and Dynamic Tables, Concept of interface and implementation, Array as an ADT: Different types of Array Implementations. List Interface & List implementations, Concept of Iterator: Operations on Lists and Arrays – traverse, search, replace, reverse, copy, Doubly Linked List.

14 hours

Unit 2: Linked List Variations, Graphs

Skip List: interface and implementation, Multilist: Sparse Matrices, Graphs-Representation, Graph Algorithms: elementary algorithms, Bellman Ford, Johnson's algorithm for sparse graphs, Graph Isomorphism, State space search techniques.

14 hours

Unit 3: Trees and Queue

Tree: Basics, Self-Balancing Binary Search Trees, Trie: Prefix and Suffix trees, Treaps, Double Ended Queue.

14 hours

Unit 4: Priority Queue and Combination of Data Structures

Priority Queues: Leftist Heaps, Skewed Heaps, Binomial Heaps, Fibonacci Heaps, Greedy method, Branch and Bound techniques, Introduction to Spatio-Temporal Data structures and R-Trees.

14 hours

Tools/Languages: C, C++, Java or Python Programming Languages

Text Book(s):

1: “Introduction to Algorithms”, T. H Cormen, C E Leiserson, R L Rivest and C Stein, Prentice-Hall of India, 3rd Edition, 2010.
2: “Spatial Statistics and Spatio-Temporal Data: Covariance Functions and Directional Properties”, Michael Sherman, Wiley, 2010.

Reference Book(s):

1:“Abstract Data Types: Specifications, Implementations, and Applications”, Nell Dale, Henry M. Walker, Jones & Bartlett Learning, 1996.
2: “Data Structures and Algorithm Analysis in C++”, Mark Allen Weiss, Pearson, 4th Edition, 2014.
3: “Data Structures and Algorithms”, Alfred V. Aho, Jeffrey D. Ullman, Pearson, 1983.

UE21CS643A: Scalable Computing (4-0- 0-4-4)

With an increase in the amount of data and need for remote computation due to cloud computing, this course intends to introduce the students to the business needs that have driven the change and then the underlying technologies and principles of computing at scale.

Course Objectives:

The objective(s) of the course is to

- Introduce the basic principles of computing at scale and differentiate between scales up and scale out.
- Introduce the business need and applications for scaling computing.
- Introduce case studies of scalability from Cloud computing and Big Data and how the two relate to each other.
- Understand the theoretical considerations that impact the design of scalable systems.

Course Outcomes:

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

- Motivate and explain trade-offs in computing at scale
- Demonstrate development of Cloud/Hadoop applications.
- Evaluate Service- oriented technologies and their potential for transformation of business.
- Demonstrate use of tools for developing applications at scale.

Course Content:

Unit 1 : Systems Modelling, Clustering and SOA: Operating systems concepts review, Scalable computing over the internet, Technologies for network based systems, system models for distributed and cloud computing, Software environments for distributed systems, performance metrics, Services and SOA, REST,RPC .

14 Hours

Unit 2: Cloud Platform Architectures: Cloud computing and service models – IaaS, PaaS, SaaS, Architectural design, Programming models – IaaS, Master slave, p2p and overlay networks
Case study: AWS/Open stack .

14 Hours

Unit 3: Programming Models: PaaS Models Case study: Azure. Messaging Oriented Middleware, Microservices model - performance, Orchestration, Continuous integration, DevOps.
Case Study: Netflix/Uber

14 Hours

Unit 4: Big Data Programming models: Introduction, Distributed File systems Case Study:HDFS/GFS, Map Reduce Programming Model, Spark
Case Study: Kubernetes

14 Hours

Tools/Languages: AWS, PostMan, Flask, Docker, Hadoop.

Reference Book(s):

1: “Distributed and Cloud Computing: From Parallel Processing to Internet of Things”, Kai Hwang, Jack Dongarra and Geoffrey Fox, 1st Edition, 2013, Morgan Kaufmann.
2: “Hadoop: The Definitive Guide, Tom White”, 4th Edition, O‟Reilly, 2015.
3: Cloud Native DevOps with Kubernetes, Justin Domingus and John Arundel, O‟Reilly, 2019.

UE23CS644A : Stochastic Models and Machine Learning (4-0-0-4-4)

UE23CS644A: Stochastic Modelling and Machine Learning (4-0-0-4-4)

Machine Intelligence(ML) and Machine Learning (ML) surrounds us today: in phones that respond to voice commands, programs that beat humans at Chess and Go, robots that assist surgeries, vehicles that drive in urban traffic, and systems that recommend products to customers on e-commerce platforms. This course aims to familiarise students with the breadth of modern MI, to impart an understanding of the dramatic surge of MI in the last decade, and to foster an appreciation for the distinctive role that MI can play in shaping the future of our society. This course requires the student to have a desirable knowledge of Statistics for Data Science, Linear Algebra and its Applications and Design and Analysis of Algorithms.

Course Objectives:

- Familiarize the concepts of Intelligent Agents and Search Methods.
- Formulate a well - defined Machine Learning problem with clear metrics.
- Understand the notions of Hypotheses Space, Hypotheses Structure and Search.
- Become conversant with types of Machine Learning Algorithms, their applicability and Inductive Bias.
- Familiarize with techniques for Ensemble Learning, Nature based Optimization.

Course Outcomes:

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

- Apply Intelligent Search methods for a variety of problems.
- Distinguish categories of Data Attributes, Dimensions, and Sample Sizes.
- Acquire a thorough understanding of Supervised, Unsupervised Learning,
- Ensemble Methods and Nature based Optimization.
- Apply deep learning methods.

Course Content:

Unit 1: Introduction, Performance Metrics, Classification with Decision trees and KNN

Introduction to AI and ML, Machine Learning and its Models, Concepts of hypotheses, Version space, inductive bias, Performance metrics-accuracy, precision, recall, sensitivity, specificity, AUC, ROC, Bias Variance decomposition. Decision Trees- Basic algorithm (ID3) - for classification, Decision boundary for decision trees(x-y axis), Hypothesis search and Inductive bias, Issues in Decision Tree Learning – Over fitting, Solutions to over fitting, dealing with continuous values. Instance-based learning: k-nearest neighbour learning (Classification & Regression), Decision boundary for KNN,

14 Hours

Unit 2 : Supervised Learning with ANN, Introduction to Deep Learning techniques

Artificial Neural networks: Introduction, Perceptions, Multi-layer networks and back-propagation, Activation functions (Step, Sigmoid, Tanh, ReLU) Various Optimizers(GD, SGD, Momentum-based, Adagrad, Adam,) Introduction to Deep Learning, Introduction to Convolution operation and Convolution Neural Network. (Parameter calculation, Max pooling, Avg pooling). Introduction to Recurrent Neural Network, Vanishing and exploding gradient, Variants of RNN : LSTM, GRU(mention of gates), Introduction to Large Language Models(LLM).

14 Hours

Unit 3 : SVM, Boosting and Stochastic Models

Support Vector Machines – margin and maximization, SVM - The primal problem, the Lagrangian dual, SVM – Solution to the Lagrangian dual,(Hard Margin and Soft Margin, Classification ONLY), Kernel functions: Linear, polynomial (Derivation only for linear function).

Combining weak learners, Improving performance with Gradient Boost, Random Forest, Bayesian Learning – Bayes theorem, Concept learning, Maximum likelihood, Bayes optimal classifier, Naïve Bayes classifier, Expectation maximization and Gaussian Mixture Models.

14 Hours

Unit 4 : HMM, Unsupervised Learning ,Dimensionality Reduction and Genetic Algorithms, PSO:

Hidden Markov Models, Hierarchical vs. non-hierarchical clustering, Agglomerative and divisive clustering, K-means clustering, Bisecting k-means, K-Means as special case of Expectation Maximization, Dimensionality reduction techniques – PCA, SVD applications. Genetic Algorithms – Representing hypothesis, Genetic operators and Fitness function and selection - Application in Decision Trees, Weight Determination and clustering.

14 Hours

Hands-on : 14 Hours

Applications using

1. Decision Trees.

2. ANN (Basics).
3. CNN model.
4. SVM (SVC).
5. Naive Bayes
6. HMM

Tools / Languages : Pytorch.

Text Books(s):

1: “Machine Learning”, Tom Mitchell, McGraw Hill Education (India), 2013.
2: “Neural Networks in Deep Learning” Charu agarwal, Springer International Publishing AG, part of Springer Nature 2018.
3: “Pattern Recognition and Machine Learning”, Christopher Bishop, Springer (2nd Printing), 2011.

Reference Book(s):

1: “Machine Learning: The Art and Science of Algorithms that Make Sense of Data”, Peter Flach, Cambridge University Press (2012).
2. “Hands-on Machine Learning with Scikit-Learn and TensorFlow”, Aurelian Geron, O’REILLY, 1st Edition, 2017.
3: “Artificial Intelligence: A Modern Approach (3rd Edition)”, Stuart Russel and Peter Norvig, Pearson , 2009.

UE23CS645A: Cyber Security Essentials (4-0-0-4-4)

This course will cover the essentials of Cyber security, and students will learn about the characteristics of security principles, technologies, and procedures to secure networks, software and other assets of an organization. They will also gain an insight into Risk Management, Incident Management, Cryptography, Operations Security, Digital Forensics etc.

Course Objectives:

- To provide background information and overall view of Cyber security.
- To understand and implement secure network designs such as Firewalls, IDS, IPS, SIEM, etc.
- To analyze potential indicators associated with web application attacks. To review the basics of cryptographic concepts, use cases and limitations.
- To summarize the importance of operations center, incident response and cyber laws of India.

Course Outcomes:

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

- Compare and contrast threats, attacks, and its indicators.
- Identify the security requirements for a network zone and determine appropriate technology.
- Classify attacks and summarize coding best practices against application attacks. Explain how cryptosystems are used to provide various principles of cyber security.
- Assess the security posture of an enterprise environment and recommend appropriate security solutions.

Course Contents:

Unit 1: Introduction to Cyber Security

CIA principles, Vulnerability, Threat and Risk, Anatomy of an Attack, Attack Landscape, Real-life Cyber-attacks, Security vs. Privacy, Security Cornerstones, Cyber security Framework, Security Mindset, Security Policy and Culture, Security Pillars, Security Principles, Cyber security best practices: Strong password, Social engineering techniques. Introduction to Basics of Network Security : NAT, Socket Programming using python, Packet Construction inside kernel , packet sending tools (netcat, telnet, ssh, /dev/tcp, /dev/udp, ping).

14 Hours

Unit-2 Network Security and Linux Basics

Packets Hop by Hop Transmission, Basics of Sniffing and Spoofing using Wireshark and Scapy, Introduction to Firewalls : writing simple firewall rules using ufw. Introduction to the Domain Name System (DNS), dig command, Intrusion Detection System (IDS) vs Intrusion Prevention System (IPS), Honeypots, Wireless security : Components, Architectural Modes, Threats in Wireless Networks. **Basics of Linux for Security** : Linux file system, Important configuration files, Users & Permissions, Files & Permissions - SUID, SGID, Adding & removing users, locating things, grep, Environment Variables, Logging, Automated Scheduling, Shell Scripting

14 Hours

Unit-3 Software Security and Cryptography

Web Security Basics - Architecture, HTML, CSS, Dynamic Content - Javascript, Browser server communication, Types of HTTP Requests - GET, POST, HTTPS, Cookies, Sandboxing Javascript, Ajax example, Brief overview of various Application attacks: Buffer Overflow vulnerabilities, SQL injection,

Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS). Cryptography : Definition, Terminology, History, Goal and Services, Classical Cryptography, Modern Cryptography. Types of Cryptography : Symmetric Key Cryptography, Asymmetric Key Cryptography, Hash Function. Applications of Cryptography : Steganography, Digital Signatures.

16 Hours

Unit-4 Operations and Incident Response

Security Information and Event Management (SIEM), Security Operations Centre (SOC), Digital forensics & Incident Response, Blockchain, Data Privacy: Personally Identifiable Information (PII), Cyber law.

12 Hours

Tools / Languages: Hashcat, Wireshark, Python(Scapy).

Proposed Labs :

1. Learn the meaning of Plagiarism.
2. Use Hashcat - password recovery tool pre-installed in kali linux to crack password hashes using various hashing algorithms and a variety of attack modes.
3. Learn the basics of network security and perform basic packet sniffing and spoofing using Scapy.
4. Introduction to Wireshark and PCAP analysis.
5. Write appropriate ufw rules to reject traffic to a website. Evade ufw firewall using SSH tunnel.
6. Web Pen Testing - SQL Injection, XSS, CSRF
7. Crypto by hand - classical ciphers
8. Metamask wallet creation and etherscan account verification.

Textbook :

1:“Computer Security: Principles and Practice”, William Stallings and Lawrie Brown, Pearson Education, 3rd Edition, 2010.

Reference Book (s):

1:“Computer & Internet Security: A Hands-on Approach”, Wenliang Du, 2nd Edition, 2019.
2: “The Official CompTIA Security+ Student Guide (Exam SY0-601)”, CompTIA, Course Edition 1.0, 2020.
3: “Cryptography and Network Security”, Behrouz A.Foruzan, 3rd Edition, Tata McGraw Hill, 2017.

UE23CS641B : Topics in Advanced Algorithms (4-0-0-4-4)

Algorithm Design and Analysis is fundamental and important part of Computer Science. The course on Advanced Algorithms introduces the learner to advanced techniques for design and analysis of Algorithms and explores a variety of applications.

Course Objectives:

- Understand in depth, Complexity notions and Computability of functions.
- Understand a few string matching/prediction algorithms and their applications.
- Understand Graph Centric, Max-Flow and Shortest Path algorithms.
- Understand efficient Polynomial Multiplication and hence DFT/FFT algorithm.
- Understand Number Theoretic algorithms and hence RSA cryptography.

Course Outcomes:

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

- Perform Complexity analysis of algorithms.
- Design and implement advanced string matching algorithms.
- Design Maxflow, advanced Graph Centric algorithms and FFT Algorithms.
- Design and implement Number Theoretic algorithms and RSA encryption.
- Design Randomized and Approximate algorithms and estimate complexity.

Desirable Knowledge: UE23CS642A - Advanced Data Structures.

Course Content:

Unit 1 : Review of Analysis Techniques

Growth of Functions, Asymptotic notations, Standard notations and common functions, Recurrences and Solution of Recurrence equations- The Substitution method, the Recurrence tree method, the Master method, Amortized Analysis:Aggregate, Accounting and Potential Methods. NP-Completeness.

14 Hours

Unit 2 : Shortest Path algorithms, Polynomials and FFT

Single source shortest paths in a DAG, Bellman - Ford Algorithm, Johnson's Algorithm for sparse graphs, Flownetworks and Ford-Fulkerson method, Maximum Bi-partite matching. Polynomials and FFT: Representation of polynomials, Efficient Polynomial Multiplication, The DFT and FFT, Efficient implementation of FFT.

14 Hours

Unit3 : Number-Theoretic and String Matching Algorithms

Elementary notions; GCD, Modular Arithmetic, Solving modular linear equations, The Chinese remainder theorem, Powers of an element; RSA cryptosystem; Primality testing; Integer factorization. Naïve string Matching, Rabin - Karp algorithm, String matching with Finite State Automata, Knuth-Morris-Pratt algorithm, Boyer–Moore algorithm.

14 Hours

Unit 4: Probabilistic, Randomized and Approximation Algorithms

Probabilistic analysis, Indicator Random variables, uses of indicator random variables, the Hiring problem, Classes of Randomized algorithms. Approximation Algorithms, Vertex Cover Problem, Travelling Sales man Problem, Randomisation and Linear Programming.

14 Hours

Tools/ Languages: C /C++ Programming Language.

Text Book(s):

1: "Introduction to Algorithms", T. H Cormen, C E Leiserson, R L Rivest and C Stein, Prentice-Hall of India, 3rd Edition, 2010.

Reference Book(s):

1:“The Algorithm Manual”, Steven Skiena, Springer, ISBN: 9788184898651.
2: “Randomized Algorithms”, R Motwani and P Raghavan, Cambridge University Press 2011 edition

UE23CS642B: Distributed Computing (4-0-0-4-4)

Distributed computing deals with all forms of computing, information access, and information exchange across multiple processing platforms connected by computer networks. Design of distributed computing systems is a complex task. It requires a solid understanding of the design issues and an in-depth understanding of the theoretical and practical aspects of their solutions. This course covers the fundamental principles and models underlying the theory, algorithms, and systems aspects of distributed computing.

Course Objectives:

- Expose students to both the abstractions and details of the file system.
- Introduce concepts related to distributed computing systems.
- To focus on performance and flexibility issues related to systems design decisions.
- To prepare students for an industrial programming environment.
- To prepare students to work on Cluster Computing.

Course Outcomes:

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

- Apply knowledge of distributed systems techniques and methodologies.
- Explain the design and development of distributed computing and distributed computing applications.
- Use the application of fundamental Computer Science methods and algorithms in the development of distributed computing and distributed computing applications.
- Discuss the design and testing of a large software system, and to be able to communicate that design to others.
- Design and discuss cluster computing.

Course Content:

Unit 1: Introduction to Distributed Computing and Basics of Cluster Computing

Motivation, Multiprocessor Vs Multicomputer Systems, Distributed Communication models: Remote Procedure Call, Publish/Subscribe model, Message Queues etc. Design issues and Challenges for building distributed computing Systems. Logical time: Scalar time, Vector time, Implementation of Logical and Vector clocks. Cluster computers and MPP architectures, Cluster job and resource management.

14 Hours

Unit 2: Global Snapshot and Overview of Grid Computing

Snapshot algorithms for FIFO/ Non FIFO channels. Terminology and Basic algorithms: Classifications and basic concepts, Elementary graph algorithms, Synchronizers, Maximal Independent set, connected dominating set. Message ordering and group communication: Message ordering Paradigms, Group communication, Application level multicast. Grid architecture and service modelling, Grid resource management.

14 Hours

Unit 3: Distributed Mutual Exclusion and Internet of Things

Assertion based and Token based Mutual exclusion. Consensus and Agreement Protocols: Agreement in failure free and systems with failures, wait-free shared memory consensus in asynchronous systems. IoT for Ubiquitous computing, RFID, Sensors and Zig Bee technologies, Applications of IoT (smart buildings, cyber-physical systems), graph theoretic analysis of social networks, Facebook, and Twitter case studies.

14 Hours

Unit 4: Self-Stabilization and Peer-to-Peer Computing

Designing self-stabilizing systems, self stabilizing distributed spanning tree, probabilistic self stabilizing leader election algorithm, self-stabilization as a solution to fault tolerance. Peer-to-Peer computing and Overlay graphs: Data indexing and overlays, unstructured and structured overlays: Bit Torrent, Tor, Bitcoin, CHORD overlay, Internet graphs, Small world networks, Scale free networks, and Evolving networks.

14 Hours

Tools / Languages: Wireshark / C compiler with Network Libraries.

Text Book(s):

1: "Distributed Computing: Principles, Algorithms, and Systems", Ajay D. Kshemkalyani, MukeshSinghal , Cambridge University Press, 2008 (Reprint 2013).

Reference Book(s):

1: "Distributed and Cloud Computing: From Parallel processing to the Internet of Things", Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra Morgan Kaufmann, 2012 Elsevier Inc.
2: "P2P Networking and Applications", John F. Buford, Heather Yu, and Eng K. Lua, Morgan Kaufmann, 2009 Elsevier Inc.
3: "Grid Computing", Joshy Joseph, and CraifFellenstein, IBM Press, Pearson education, 2011.

UE23CS643BA1: Million Way Parallelism (4-0-0-4-4)

With increasing amount of computation being done due to compute intensive applications like machine learning and audio/video processing, it is imperative to understand various parallel programming models. The student must also become familiar with various different hardware choices and the design choices. The course will also introduce various tools to work.

Course Objectives:

- Introduce the various parallel programming models required to scale.
- Introduce applications that benefit from scalability.
- Introduce different types of hardware such as multi-core CPUs, GPUs and FPGAs.
- Understand and use the tools to analyze these applications.
- Understand the impact of role of various other system components on extracting performance in parallel programs.

Course Outcomes:

- At the end of the course the student will be able to,
- Choose the right programming model for a problem.
- Demonstrate development of applications on different types of hardware.
- Evaluate choice of the right type of hardware to solve the problem.
- Demonstrate use of tools for developing parallel applications and debugging them.
- Analyze systems for issues in performance..

Course Content:

Unit 1: Introduction and Parallel Program Design & Shared Memory Programming

Multicore, Taxonomy, Performance Metrics, Amdahl's Law, Gustafson's law, Decomposition patterns, Program structure patterns. Matching program structure with decomposition, Threads, design concerns, semaphores.

14 Hours

Unit 2: Shared Memory Programming & GPU Programming

Applying semaphores, debugging multithreaded applications, OpenMP- loop level parallelism, and Task level parallelism, Synchronization, Correctness and Optimization. Case Study, GPU architecture, CUDA - programming model, 0execution model. \

14 Hours

Unit 3: GPU Programming & FPGAs

Memory hierarchy, Optimization techniques, Debugging, Case Study, Motivation, FPGA capabilities, Elements – Look up table, switches, logic blocks, interconnect.

14 Hours

Unit 4: FPGAs & Miscellaneous and Trends

I/O blocks, multipliers, memory blocks, CAD software – HDL Simulator, technology mapping, placement, bitstream generation. FPGA application – Machine learning , power considerations, memory systems (3D), Interconnection Networks.

14 Hours

Tools / Languages: OpenMP, CUDA.

Text Book(s):

1:“Multicore and GPU Programming: An Integrated Approach”, GerassimosBarlas, Morgan Kaufmann, 2015.

Reference Book(s):

1: “Programming Massively Parallel Processors: A Hands-on Approach”, David Kirk and Wen- meiHwu, Morgan Kaufmann, 1st edition, 2010.
2: “Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation”, Scott Hauck and Andre Dehon, Morgan Kaufmann 2008.

UE23CS643BA2 : Speech and Natural Language Processing (4-0-0-4-4)

Speech and Natural language processing (SNLP) is one of the most important technologies of the information age. The objective of NLP is to make machines to read, decipher, understand, and make sense of the human languages in a manner that is valuable. Speech and Natural Language Processing today powers many key real-life industry applications, such as Language Translation, Dialog Systems / Chatbots, Sentiment Analysis, Text Summarizers, Speech Recognition, and Autocorrect.

Course Objectives:

The objective(s) of the course is to

- Introduce the student to advances in multimedia technologies relevant to Big Data.
- Understand the business relevance of speech and natural language technologies.
- Introduce the basic models used for processing speech and natural language.
- Work with tools to perform speech/natural language processing.
- Gain a practical insight into solving these problems.

Course Outcomes:

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

- Demonstrate the use of speech/natural language processing in solving real-life problems.
- Demonstrate the use of tools for performing speech/NLP.
- Demonstrate capability to perform analysis and compare various models.
- Work in a team to solve speech/NLP related problems.
- Communicate the solution to the instructor using a report.

Course Content:

Unit 1 : Introduction

Business relevance, Survey of English Morphology (inflectional and derivational morphology), Finite State Morphological parsing, Porter Stemmer, Word and Sentence Tokenization, Detection and Correction of Spelling Errors, Minimum Edit Distance. **Ngrams** – Word Counting in Corpora, Simple (unsmoothed) N-Grams, Training and Test Sets, Evaluating N-Grams, Smoothing. **Vector Semantics and Embeddings -** Lexical Semantics, Vector Semantics, Words and Vectors, Cosine for measuring similarity, TF-IDF, Word2vec**.**

16 Hours

Unit 2 : Part of Speech Tagging

English Word Classes, Tagsets for English, Part-of-Speech Tagging. **Hidden Markov and Maximum Entropy Models** – Markov Chains, Hidden Markov Model, Likelihood Computation, Decoding, HMM Training, Maximum Entropy Models. **Lexical Semantics** – Word Senses, Relation between Senses, Wordnet: Database of Lexical Relations, Event Participants, Primitive Decomposition, Metaphor, Computational Lexical Semantics.

12 Hours

Unit 3 : Word Sense Disambiguation (WSD)

Supervised Word Sense Disambiguation, WSD Evaluation, Minimally Supervised WSD, Word Similarity, Semantic: Role Labeling. **Syntactic Parsing** – Parsing as Search, Ambiguity, Search in the Face of Ambiguity, Dynamic Programming Parsing Methods, Partial Parsing, Statistical Parsing – Probabilistic CFGs, Probabilistic CKY Parsing, Ways to Learn PCFG, Rule Probabilities, Problems with PCFGs, Improving PCFGs, Probabilistic Lexicalized CFGs, Evaluating Parsers. Dependency

18 Hours

Unit 4 : Phonetics

Speech Sounds and Phonetic Transcription, Articulatory Phonetics, Phonological Categories and Pronunciation Variation, Acoustic Phonetics and Signals, **Automatic Speech Recognition** – Speech Recognition Architecture, HMM Applied to Speech, Feature Extraction: MFCC Vectors, Acoustic Likelihood Computation, Lexicon and Language Model, Search and Decoding, Embedded Training, Word Error Rate.

10 Hours

Tools / Languages : CoreNLP, Natural Language Toolkit (NLTK), TextBlob,Gensim, SpaCy, PyTorch-NLP , OpenNLP

Text Book:

1: **"**Speech and Language Processing: An Introduction to Natural Language Processing", Daniel Jurafsky and James H. Martin, Prentice Hall, 2009.(Draft copy ,3rd Edition,2018 can be referred)

Reference Books:

1: "Computational Linguistics and Speech Recognition", Dan Jurafsky, James H. Martin, Prentice Hall, 2nd Edition, 2008.
2: "Foundations of Statistical Natural Language Processing", Christopher D. Manning and Hinrich Schütze, MIT Press, 1999.
3: "Natural Language Understanding",James Allen, Benjamin/Cummings publishing Company, 2ndedition, 1995.
4: "Digital Processing of Speech Signals", Lawrence R. Rabiner, Ronald W. Schafer, Prentice Hall, 1978.

UE22CS643BA3 : Topics in Computer and Network Security (4-0-0-4-4)

This course gives an overview and conceptual understanding of security aspects involved in a network of computers. Students will have opportunities to participate in well-designed hands-on sessions and case study discussions.

Course Objectives:

- To provide an overall view of Network Security.
- To understand the security problems associated in the design and implementation of the TCP, IP/ICMP, ARP protocols by analysing the network packets.
- To learn the vulnerabilities in DNS protocol and to implement and experiment with Firewall rules.
- To provide an overview of network management techniques and the usage of VPN.
- To analyse the risk management and security aspects of wireless networks.

Course Outcomes:

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

- Sniff packets from clients and analyse them to extract important info such as headers, passwords etc.
- Launch DoS and MITM attacks using various protocol vulnerabilities and mitigate them.
- Configure firewalls on Linux machines and exploit vulnerabilities on DNS protocol.
- Design and implement VPN for a secure connection over internet.
- Master in wireless network security systems in depth and perform effective network management.

Course Contents:

Unit 1: Packet Sniffing & Spoofing MAC Layer and Attacks

Packet Sniffing and Spoofing: Introduction, Sending packets: Network Interface Card (NIC), BSD packet filter (BPF). Packet sniffing: Receiving packets using sockets, Packet sniffing using Raw sockets, Packet sniffing using PCAP API, Processing captured packets. Packet spoofing: Sending normal packets using sockets, Constructing spoofed raw ICMP packets and UDP packets. Sniffing and then spoofing, Python vs Scapy, Endianness. MAC layer and attacks: The MAC layer, ARP protocol, ARP cache poisoning attacks, MITM using ARP cache poisoning, Demo, Countermeasure. Network layer: IP, ICMP and attacks: IP protocol, IP fragmentation, Attacks using IP fragmentation: Problem and solution, Routing, and spoofing prevention, ICMP protocol, ICMP redirect attack, Smurf and other ICMP attacks.

14 Hours

Unit 2: TCP Attacks and DNS Attacks

Attacks on the TCP protocols: TCP overview, Send and receive buffers, SYN flood attack: TCP 3-way handshake, the SYN flooding attack, Launching the attack using Netwox and C, Countermeasure. TCP reset attack: TCP reset attack on Telnet, SSH and video streaming connections. TCP session hijacking attack: TCP session and session hijacking, launching the attack, Hijacked TCP connection. Reverse shell: working, redirecting IO to TCP connection, Creating reverse shell. Counter measure. Case Study – 1. DNS Attacks: DNS hierarchy, zones and servers, DNS query process, Constructing DNS request and response using Scapy, DNS attacks: Overview, Local DNS cache poisoning attack, Remote DNS cache poisoning attack (Kaminsky attack), Reply forgery attacks from malicious DNS servers, Countermeasure against DNS spoofing attacks,

14 Hours

Unit 3: Firewalls, IDS, IPS

Firewall: Introduction, Requirements of a firewall, Firewall characteristics and Access policy, Types of firewalls, NG firewall, Shortcomings, Firewall location and configuration: DMZ networks, Firewall topologies. Introduction, Build a simple firewall, Netfilter, iptables firewall in Linux, Stateful firewall and connection tracking, Application/Proxy firewall and Web proxy, Evading firewalls. Intrusion Detection and Prevention: Intruders, Intrusion detection, Analysis approaches, Host-based intrusion detection, Network-based intrusion detection, Distributed or hybrid intrusion detection, Honeypots, Example system: Snort, Intrusion prevention system. SOC, SIEM.

14 Hours

Unit 4: Virtual Private Network and Wireless Network Security

Case Study – 2, Virtual Private Network: Introduction, Why VPN, analogy, and tunnelling. Overview of TLS/SSL VPN: Establishing a tunnel, Forwarding, and releasing IP packets, TLS/SSL VPN details. Building, Setup and Testing VPN. Bypassing Firewall using VPN. Wireless Security: Communications and 802.11 WLAN standards: Wired

Equivalent Privacy (WEP), Wireless Protected Access (WPA), IEEE 802.1x, 802.11i/ WPA2, Wireless Network Threats.

14 hours

Note: Course includes hands-on experience on specific topics in the form of Lab and/or Assignment along with relevant cyber security case study discussions

Tools / Languages: SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.

Text Book(s):

1: "Internet Security: A Hands-on Approach", Wenliang Du, 3rd Edition, 2019.

Reference Book (s):

1: "Computer Security: Principles and Practice", William Stallings and Lawrie Brown, Pearson Education, 3rd Edition, 2010.

UE22CS644BB1: Advanced Cloud Computing (4-0-0-4-4)

Cloud Computing Architecture has been revolutionizing IT development. The benefits which are achieved with Cloud environments are harnessed significantly by using development models and architectures. Cloud native architectures enable the leverage of the cloud benefits; the course will introduce Cloud Native Architectures with specific focus on microservices as the design principle.

Course Objectives:

- Introduce students to various aspects of cloud native architectures and critically evaluate different alternatives.
- Explore and demonstrate micro services and their creation usage etc.
- Understand the principles and techniques for scalability, cost optimization, security as provided by Cloud native platforms.
- Understand the design of efficient cloud application which can be updated regularly for meeting changing requirements from business. Techniques and patterns for automation for cloud applications.

Course Outcomes:

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

- Students will have an ability to consider the different approaches for migration and critically evaluate the different approaches.
- The students will have an ability to critically evaluate various alternatives of cloud native architectures.
- Students will be able to build and deploy a sample microservice.
- Student will be able to build cloud native applications using microservices, dockers, Kubernetes and Terraform. Students will be able to critically evaluate the different aspects of scalability, security etc. and consider potential solutions towards these.

Course Content:

Unit 1: Serverless Computing

Introduction, FaaS, understanding serverless architectures, Serverless pros and cons, Serverless Framework, architectures and patterns, AWS Lambda .

14 Hours

Unit 2: Basics of Cloud Native Architecture

Fundamentals of cloud native applications, Cloud Native vs Traditional architectures, Functions vs. Services, From VMs to Cloud Native, API Design and Versioning, Service communication, Gateways, Service Mesh.

14 Hours

Unit 3: Kubernetes and Terraform

Kubernetes: Cluster architecture, Kubernetes Services, Illustration of managed Kubernetes services, working with Kubernetes Objects – deployments, Pods, scheduler, managing resources, utilities to work with resources.

Terraform: Infrastructure as code, Terraform in practice, Terraform Vs. Kubernetes.

14 Hours

Unit 4: Performance management and Security of cloud applications

Application performance management in cloud environments – public, private, hybrid, IaaS, PaaS, SaaS, KPIs/metrics. User experience management - various industry and government organization standards,
APM approaches - command-and-control, dynamic policy rules.
Guiding Security design principles for Cloud Computing, Identity and access management, Single Sign-on, Identity Federation, Identity providers and service consumers.

14 Hours

Tools/Languages: AWS, Kubernetes, Terraform.

Text Book(s):

1 : Cloud Native Using Containers, Functions, and Data to Build Next-Generation Applications by Boris Scholl, Trent Swanson & Peter Jausovec, O'Reilly, First Edition, 2019.
2: Application Performance Management (APM) in the Digital Enterprise - Managing Applications for Cloud, Mobile, IoT and eBusiness by Rick Sturm Carol Pollard Julie Craig, Morgan Kaufman, 2017
3. Serverless Architectures on AWS by Peter Sbarski, Yan Cui, Ajay Nair, Manning Publications, Second Edition, 2022

Reference Book(s):

1: "Cloud Native Architectures", by Tom Laszewski , Kamal Arora , Erik Farr, Publisher Packt, 2018
2: "Cloud Native DevOps with Kubernetes" by John Arundel, Justin Domingus, O'Reilly,2019.
3: "Securing The Cloud: Cloud Computing Security Techniques and Tactics" by Vic (J.R.) Winkler, Syngress/Elsevier, 2011

UE23CS644BB2: Virtual Reality & its applications (4-0-0-4-4)

The physical entity is simulated into virtual or the imaginary environment which is designed as software or as a program that defies the beliefs of a user compelling him/her to accept it as actual reality. Virtual Reality actually exploits and plays with the sensations & perceptions of our brain by simulating an artificial environment that actually doesn't really exist, but our brains think that it does it's just like make-belief.

Course Objectives:

- Understanding basics of virtual reality through human perception.
- Introduce the use of geometric transformations on graphics objects and their application in composite form and its implementation.
- Modelling 3D world using Unity3D with the knowledge of 3D geometry.
- Exploring applications in the area of Virtual Reality using Motion sensing and Tracking.

Course Outcomes:

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

- Demonstrate the fundamentals of human physiology in the context of virtual reality.
- Apply understanding of graphical rendering to build graphical application using OpenGL
- Apply techniques of 3D geometry for building intuitive graphical applications.
- Apply techniques and tool to design an immersive virtual reality experience.
- Use Unity3D to develop complex graphical applications including 3D interactive games.
- Apply graphics in greater depth to more complex courses like Image Processing, Virtual Reality, etc...

Unit 1: Introduction and Human Perception

Introduction to Virtual reality, Modern VR Experience, History, Bird's Eye View, Hardware, software, Human Physiology and Perception, The Human Eye, cameras, displays, Visual Perception: Perception of Depth, motion and Colour, Combining sources of information; Audio: Auditory perception auditory rendering; Frontiers; Touch and proprioception, smell and taste, robotic Interfaces, Brain Machine Interfaces.

14 Hours

Unit 2: 3D Computer Graphics

Visual rendering: Ray tracing and shading models, Rasterization, Correcting Optical Distortions, Improving Latency and framerates, Immersive Photos and videos, The OpenGL: The OpenGL API, Primitives and Attributes, Colour, Viewing, Control Functions, Polygons.

14 Hours

Unit 3: Geometric Objects and Transformations:

Scalars, Points and Vectors, Three-Dimensional Primitives, Coordinate Systems and Frames, Modelling a Coloured Cube, Overview of 2D Transformations: Rotation, Translation and Scaling, Affine transformations, Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Interfaces to Three Dimensional Applications, Quaternion's.

14 Hours

Unit 4: Tracking and Intelligent VR

Motion in Real and Virtual World: Velocities and Acceleration, Physics in the virtual world, Mismatched Motion and Vection. Tracking: Tracking in 2D orientation, tracking 3D orientation, Tracking Position and Orientation, Tracking Attached Bodies, 3D scanning and Environments. Axis-Angle Representations of Rotation, Reactive AI: Adaptability, Complexity and Universality, Feasibility, More Intelligence in the System: Deliberative AI, Reinforcement learning through interaction, Imitation Learning through human demonstration.

14 Hours

Tools/Languages: C++, Unity3D, Blender

Textbook(s):

1. Virtual Reality, Steven M LavValle, University of oulu, Cambridge University press, 2020 (Available for downloading at http://lavalle.pl/vr/)
2. Creating Augmented and Virtual Realities, by Erin Pangilinan, Steve Lukas, Vasanth Mohan, PUBLISHED BY:O'Reilly Media, Inc.PUBLICATION DATE:March 2019

Reference Book(s):

1. "Interactive Computer Graphics - A top-down approach with shader-based OpenGL", Edward Angel and Dave Shreiner, Pearson Education, Sixth edition, 2012.
2. "Unity Game Development in 24 Hours" , Geig, Mike. Sams Teach Yourself . Pearson Education, 2014.

UE23CS644BB3: Software Security (4-0-0-4-4)

The course presents the challenges and mechanisms to develop safe and secure software. The main focus of the course is to provide an insight of the software vulnerabilities, its consequences during exploitation and the procedure to harden the system against those attacks.

Course Objectives:

- To understand software threats, attacks and design secure software without privilege escalation attacks
- To gain a good comprehension of the landscape of Operating system vulnerabilities.
- To learn web application vulnerabilities
- To understand and practice penetration testing for applications

Course Outcomes:

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

- Design and develop software with security.
- Defend the vulnerabilities of Operating Systems and browsers.
- Analyse various threats and vulnerabilities involved in web application development and apply mitigation approaches to avoid them.
- Inspect various security testing strategies and apply penetration testing to understand the intrusion resiliency.

Desirable Knowledge: UE23CS645A -Cyber Security Essentials.

Course Content:

Unit 1: Introduction and Privilege Escalation attacks

Recent Software threats and Vulnerabilities referenced on Security Standards Organizations. Secure Software Development Life Cycle using use cases and misusecases, Set-UID program: Need for privileged programs, Set-UID mechanism, Superman story, Attack surfaces, Invoking other surfaces, Principle of least privilege. Environment variables and attacks: Environment variables, Attack surface, Attacks via Dynamic linker, External program, and Library**.** Shellshock attack: Shellshock vulnerability, Shellshock attack on Set-UID and CGI programs**.**

14 Hours

Unit 2: Software Vulnerabilities

Buffer overflow attack: Stack and function invocation, Stack buffer-overflow attack, Attacks with Unknown address and Buffer size, Shellcode, Countermeasures & Defeating it. Return-to-libc attack, Format string vulnerability: Introduction to functions and format string, Vulnerable program, Exploiting the vulnerability, Code injection attack, Countermeasures. **Case study : Target case study ,** Threat modelling using STRIDE.

14 Hours

Unit 3: Malware and Web Security

Malware and its Types, Malware analysis: Conficker, Morris, Stuxnet worm, Ransomware attack and protection techniques. Phishing types, tactics and tips to avoid phishing.-Web API security, Security Issues and Challenges in browser and web applications, Cross Site Request Forgery (XSRF/CSRF): Cross-site requests and its problems, CSRF attacks, Attacks on HTTP GET and POST services, Countermeasures, Cross-Site Scripting (XSS/CSS) Attack: CSS attack, CSS attacks in action, Self-propagation, Preventing CSS attacks**.**

14 Hours

Unit 4: Web attacks and Penetration Testing

SQL injection attack: Introduction to SQL, interacting with database in web, Launching SQL injection attacks, Countermeasures **.** Cross-Site Scripting (XSS/CSS) Attack: CSS attack, CSS attacks in action, Self-propagation, Preventing CSS attacks**.** Static analysis, Penetration testing process, Penetrations testing tools review, Penetration testing on live websites. Case study: Apple - Privacy vs Safety. Case study

14 Hours

Hands-on exercises:

1. Set-UID program & Environment variables and attacks
2. Shellshock attack
3. Buffer overflow attack
4. Target Case study
5. Return-to-libc attack.
6. Format string vulnerability.
7. XSS attack
8. CSRF attack.
9. SQL injection.
10. Apple case study
11. Penetration testing on live websites

Tools / Languages: SEED Labs VM, Scapy, Burp Suite, Metasploit.

Text Book(s):

1: “Computer & Internet Security: A Hands-on Approach”, Wenliang Du, 2nd Edition, 2019.

Reference Book(s):

1: “Computer Security: Principles and Practice”, William Stallings and Lawrie Brown, Pearson Education, 3rd Edition, 2014.

2: “Secure Programming with Static Analysis”, Brian Chess and Jacob West, Pearson Education, 2007.

3. “Big Breaches: Cybersecurity Lessons for Everyone”, Neil Daswani, Moudy Elbayadi, Apress, 2021

UE23CS645BC1: Advanced Big Data Analytics (4-0-0-4-4)

The course explores the big data analytics lifecycle: question formulation, data collection and cleaning, exploratory analysis, statistical inference, prediction and decision-making. Focuses on building analytical models using key principles and techniques.

Course Objectives:

- Introduce alternative techniques to perform big data processing.
- Introduce applications of Big Data Processing.
- Use tools and techniques to analyze a large data corpus.
- Technologies for performing processing at large scale.
- Perform a group-based activity to apply tools and techniques learnt to a real world problem.

Course Outcomes:

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

- Motivate and explain trade-offs in big data processing technique design and analysis in written and oral form.
- Demonstrate the usage of tools to design Big Data applications.
- Demonstrate development of analytics applications using alternative technologies to Hadoop.
- Demonstrate ability to work in a small team to solve a real world problem with applications to the society
- Communicate the design through a presentation and build a prototype to showcase the design.

Course Content:

Unit 1: Introduction and MapReduce

Introduction, HDFS overview, formats, MapReduce architecture, YARN, limitations of MapReduce, Algorithms: PageRank, Alternatives to Map Reduce – Iterative, Workflow processing, Workflow model case study.

14 Hours

Unit 2: In Memory processing

Graph Processing, In-memory computation., Apache Spark – RDDs, Scala Performance advantages, Introduction to machine learning. Machine learning with Spark. Clustering, and Collaborative filtering Algorithms applied to Big Data.

14 Hours

Unit 3: Other models and algorithms

Graph model case study – Pregel/Graph. Computation model. Case study – Tensorflow, Watson. Business applications of timeseries data, challenges/tools to process timeseries data. Searching/Matching algorithms

14 Hours

Unit 4: Timeseries analysis

Speech Processing: Variety in Big data, Business cases for Multimedia-Speech processing. Hidden Markov Models, Case study: Sphinx. Video Processing.

14 Hours

Tools/Languages: Apache Hadoop, Hive, Spark, Solr, R, Google Cloud Platform, IBM Watson.

Text Book(s):

1: “Big Data Analytics- A Hands-on Approach”, ArshdeepBahga, Vijay Madisetti,VPT Publication,1st Edition, 2018.

Reference Book(s):

1: “Big Data Analytics Beyond Hadoop”: Real-Time Applications with Storm, Spark, and More HadoopAlternatives,VijaySrinivasaAgneeswaran PhD, 1st Edition, Pearson, 2014.
2: “Mining of Massive Datasets”,AnandRajaraman,JureLeskovec,Jerey D. Ullman, 2nd Edition, Cambridge University Press, 2014.
3: Abu-El-Haija, Sami, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, BalakrishnanVaradarajan, and SudheendraVijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark."arXiv preprint arXiv:1609.08675(2016).

4: Huang, Qi, PetcheanAng, Peter Knowles, Tomasz Nykiel, IaroslavTverdokhlib, AmitYajurvedi, Paul Dapolito IV et al. "SVE: Distributed video processing at Facebook scale." In Proceedings of the 26th Symposium on Operating Systems Principles, pp. 87-103. ACM, 2017.

UE23CS645BC2 : Deep Learning Theory and Practices (4-0-0-4-4)

Deep Learning has received a lot of attention over the past few years and has been employed successfully by companies like Google, Microsoft, IBM, Facebook, Twitter etc. to solve a wide range of problems in Computer Vision and Natural Language Processing. In this course we will learn about the building blocks used in these Deep Learning based solutions. At the end of this course students would have knowledge of deep architectures used for solving various Vision and NLP tasks.

Course Objectives:

- To impart knowledge on Feed Forward Neural Networks.
- Introduce students to Convolutional Neural Networks and Transfer Learning.
- Provide in-depth coverage of Sequence Modelling.
- Introduce students to Auto encoders.
- Introduce students to Generative Adversarial Networks and Graph Neural Networks.

Course Outcomes:

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

- Develop a simple Feed Forward Neural Network using Multilayer Perception.
- Classify images using CNN.
- Solve time-series related problems with RNN.
- Use efficient data representations using Auto encoders.
- Generate data in the form of images using Generative AI models.

Course Contents

Unit 1: Introduction to Deep Learning: CNN Model & Transfer Learning

Introduction**,** Activation functions, Loss functions, Batch Normalization, Regularization and Optimization. **Convolutional Neural Network(CNN):** Introduction, Filters, FeatureMaps, Max-Pool Layers, Other Pooling Types, Back Propogation. Convolution Architectures - Alexnet, ZFNet, VGGNet, GoogleNet, ResNet. **Transfer Learning**: Introduction, Motivation, Variations, TL Architecture of CNNs. Hands-on: Assignment on CNN & TL.

14 hours

Unit 2: Recurrent Neural Networks (RNN)

Introduction-Recurrent Neurons, Memory Cells, Variable-Length Input-Output Sequences, RNN Architecture, Sequence learning problem, BPTT-Back Propagation Through Time, truncated BPTT, Vanishing and Exploding Gradient, Bidirectional RNN, LSTM Cell and GRU Cell, Text Classification with RNN, Encoder/Decoder architecture, Seq2Seq model with Attention, Transformer model and BERT architecture, Transformer Attention.

14 hours

Unit 3: Generative Models and GNNs

Introduction to Autoencoders , Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders, Contrastive autoencoders , Variational Auto Encoders(VAEs). **Generative Adversarial Networks**(GANs)-Architecture and Training Methods, Image Generation, DCGAN, Style GAN, WGAN, Applications.**Graphical Neural Networks(GNN)**: Introduction to GNNs, Graph Convolution Networks, Applications.

14 Hours

Unit 4: Reinforcement Learning, **Diffusion Models, Federated Learning and Overview of Latest DeepLearningModels:**

Introduction, Basic Framework of RL, Learning to Optimize Rewards, Credit Assignment Problem, Temporal Difference, Learning and Q Learning. **Deep RL:** Deep Q Learning, Training and Testing. **Diffusion Model**, Stable diffusion architectures, Introduction to Vision Transformers, GPT Architecture. **Federated Learning:** Horizantal, Vertical and FTL (Federated Transfer Learning **.**

14 Hours

Tools/ Languages: Pytorch.

Text Book(s):

1: “Advanced Deep Learning with Python” - Ivan Vasilev, Packt Publishing, 2019.
2: “Neural Network and Deep learning” by Charu C Agarwal. Springer International Publishing 2018.
Book)
3: “Deep Learning”, Ian Goodfellow, YoshuaBengio, Aaron Courville http://www.deeplearningbook.org/ (E-Book)

Reference Book(s):

1: “Hands-on Machine Learning with Scikit-Learn and TensorFlow”, Aurelian Geron, O’REILLY, 1st Edition, 2017.
2: “Deep Learning with Keras”, Antonio Gulli and Sujit Pal, Packt Publishing, 1st Edition, 2017.
3: “Pattern Recognition and Machine Learning”, Christopher Bishop, Springer, 1st Edition, 2011 (Reprint).
4:Handouts: Transfer Learning / Latest Deep Learning Techniques / Vision Transformers/ GPT / FL

UE23CS645BC3: Cryptography (4-0-0-4-4)

Cryptography is the science of securing data by using mathematical concepts. Cryptography involves the authentication and verification of data in all domains by applying Cryptographic protocols.

Course Objectives:

- Enable to learn the fundamental concepts of cryptography and utilize these techniques in computing systems.
- Discuss about various encryption techniques.
- Understand the concept of public key Cryptography.
- Introduce message authentication and hash function.
- Provide Lab sessions for each unit to help gain deeper insight into Cryptography.

Course Outcomes:

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

- Appreciate the impact of cyber-attacks on the society and the necessity of cryptography.
- Analyse Cryptographic techniques using the mathematical foundations of cryptography.
- Design applications/protocols using cryptographic techniques.
- Apply cryptanalysis to solve real time problems.
- Evaluate the authentication and Hash Algorithms.

Course Content:

Unit 1: Introduction to Cryptography: Why Cryptography? Security trends – legal, ethical and professionalaspects of security, Basic Cryptographic primitives (encryption, decryption, signatures, and authentication), and Classicalencryption techniques: substitution technique, transposition techniques, Steganography, Historical Ciphers andtheir cryptanalysis, Classical vs. Modern cryptography. Principles of Modern cryptography, Perfectly-secret encryption – Vernam's One-time-pad encryption – Limitations, Shannon's theorem.

14 Hours

Unit 2: Modern Cryptography: Stream Ciphers, Block cipher design principles, Block Vs Stream cipher. Mathematical Modular arithmetic- Euclid's algorithm,Congruence and matrices. Structures: Groups, Rings, Fields- Finite fields, Pseudorandom Generators (PRNG). Algebraic Private/Symmetric Key Ciphers: Feistel network, DES, AES, Cryptanalysis: Block cipher mode of operation, Chosen-Ciphertext Attacks, Differential and linear cryptanalysis.

14 Hours

Unit 3: Private Key Cryptography: Public Key Cryptography**:**Mathematics of Public Key Cryptography: Primes, Factorization, Chinese Remainder Theorem, Key Management and the Public Key Revolution: Key distribution and Key Management,Diffie Hellman Protocol,Elgamal encryption, RSA Encryption: Algorithm, Implementation issues and Pitfalls. Rabin Encryption Scheme: Trapdoor, Scheme, Digital Signature: Certificates and Public Infrastructure, Attacks, Scheme, Applications, Signatures from Hash Functions.

14 Hours

Unit 4: MAC and Hash Message Authentication Code (MAC) – Definition, Message Integrity, Cipher Block Chaining (CBC-MAC), Constructing Secure message Authentication codes, Authenticated Encryption, Hash Functions and Applications: MAC using Hash functions HMAC, Generic Attacks on Hash Functions, Random Oracle Model, Applications, Hash functions: MD5, SHA, collision resistant hashing, Merkle-Damgrad and Davies Meyer.

14 Hours

Tools / Languages: Seed virtual machine environment.

Text Book(s):

1: “Introduction to Modern Cryptography”, Jonathan Katz, Yehuda Lindell, CRC Press, 2018.

Reference Book(s):

1: “Cryptography and Network Security”, BehrouzA.Forouzan, Tata McGraw Hill 2007.