

Jharkhand Raksha Shakti University

Meur's Road SKIPA Premises, Ranchi



Detailed Syllabi (First Year)

of

M.Sc. Programme

in

Computer Application and Cyber Security (Two Year)

(W.e.f. from Academic Year 2026-27)

DEPARTMENT OF CYBER DEFENCE

SSK David Shuchi Beiga Kiyann Jm
Laxman

Course Title	Mathematical Foundations for Machine Learning
Course Code	MCACS - 101
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. Build a strong mathematical foundation to understand machine learning algorithms.
2. Apply linear algebra, probability, and statistics to real-world data problems.
3. Understand and use calculus and optimization in learning algorithms.
4. Connect mathematical theories to algorithmic and model design.
5. Prepare students for advanced topics like deep learning and probabilistic models.

Course Outcomes:

After successful completion of this course, students will be able to:

CO1: Demonstrate proficiency in linear algebra techniques used in ML models.

CO2: Use probabilistic reasoning and statistical tools in data modeling.

CO3: Perform calculus-based optimizations for ML algorithms.

CO4: Analyze ML systems using convex optimization principles.

CO5: Apply mathematical tools to implement ML algorithms.

Syllabus

Module I:

(9 Hours)

Vectors and vector spaces, Matrix operations, Orthogonality, Linear independence, Projection, Eigenvalues & eigenvectors, Diagonalization, Matrix norms, SVD, PCA, Matrix calculus.

Module II:

(9 Hours)

Probability axioms, Conditional probability, Bayes' theorem, Random variables, Distributions, Expectation & variance, Gaussian & multivariate Gaussian, LLN, CLT.

Module III:

(9 Hours)

Limits, Partial derivatives, Gradients, Chain rule, Jacobian & Hessian, Taylor expansion, Optimization, Gradient descent (SGD, RMSProp, Adam).

Module IV:

(9 Hours)

Convex sets/functions, Optimization, Lagrange multipliers, Duality, KKT conditions, Regularization (L1, L2), Loss functions.

SSG *Plaw* *Shuchi Praga* *Arjun* *Sanjay*

Module V:

(9 Hours)

Descriptive stats, Hypothesis testing, Confidence intervals, MLE, MAP, Bias-variance, Entropy, KL divergence, Mutual information, Bayesian networks.

Practical's

Implement the following programs using NumPy/Python

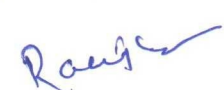
1. Solve a system of linear equations $Ax=b$ using Gaussian Elimination and compare it with the Matrix Inversion method.
2. Develop a program to project a 3D vector onto a 2D plane and visualize the projection error.
3. Write a program to perform image compression using Singular Value Decomposition (SVD) by retaining the top kkk singular values.
4. Implement Principal Component Analysis (PCA) using eigendecomposition of the covariance matrix.
5. Design a simple automatic differentiation (Autograd) system for a polynomial function using NumPy/Python.
6. Optimize a non-convex function (e.g., Rosenbrock function) using gradient-based optimization with different learning rates and analyze convergence.
7. Implement and compare the closed-form solution and Stochastic Gradient Descent (SGD) for linear regression.
8. Develop a program to update the parameters of a Beta distribution using sequential coin flip data (Bayesian updating).
9. Implement clustering of a synthetic dataset using the Expectation-Maximization (EM) algorithm for Gaussian Mixture Models.
10. Transform non-linearly separable data into a higher-dimensional feature space to achieve linear separability.
11. Formulate and solve the dual optimization problem for a hard-margin Support Vector Machine (SVM).
12. Build a two-layer neural network using only NumPy and implement forward and backward propagation by manually computing gradients.

Text Book:

1. Mathematics for Machine Learning by Deisenroth, Faisal, and Ong.
2. Deep Learning (Part I: Applied Math) by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

Reference Book:

1. Introduction to Linear Algebra by Gilbert Strang.
2. An Introduction to Statistical Learning by James, Witten, Hastie, and Tibshirani.

889  Shuchi Priya   

Course Title	Advanced Artificial Intelligence
Course Code	MCACS - 102
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To understand the fundamental concepts of probabilistic models.
2. To explore the scope and significance of generative networks in Artificial Intelligence.
3. To understand the architecture and training mechanisms of autoencoders.
4. To learn the principles and applications of transfer learning.
5. To develop a comprehensive understanding of ensemble learning methods and their applications.
6. To explore emerging and nascent technologies in Artificial Intelligence.

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1 Acquire foundational knowledge of probabilistic models.
- CO2 Analyze the working principles and architectures of generative networks.
- CO3 Interpret different types and components of autoencoders.
- CO4 Apply transfer learning techniques in practical scenarios.
- CO5 Utilize ensemble learning methods to improve predictive performance.
- CO6 Understand emerging trends and technologies in Artificial Intelligence.

Syllabus

Module I:

(9 Hours)

Basic terminology, pre-trained models and datasets, feature extraction, fine-tuning techniques. Recent advancements: self-supervised learning and meta-learning. Introduction to ensemble learning. Bagging and Random Forests. Boosting algorithms: AdaBoost, Gradient Boosting, XGBoost. Stacking and blending techniques. Applications in regression and classification tasks.

Module II:

(9 Hours)

Overview of generative models and their significance in AI. Fundamentals of probability theory and generative modelling. Introduction to GANs, VAEs, and other generative models. Importance and challenges of generative modelling. Gaussian Mixture Models (GMMs),

[Handwritten signatures]

Hidden Markov Models (HMMs), Bayesian Networks, Markov Random Fields (MRFs), and Probabilistic Graphical Models.

Module III:

(9 Hours)

Architecture of Generative Adversarial Networks, Generator and Discriminator models, Training procedure, Vanilla GAN. GAN variants and improvements: DCGAN, WGAN, Conditional GAN, CycleGAN. Challenges in GAN training: instability, mode collapse.

Module IV:

(9 Hours)

Basic components of Variational Autoencoders (VAEs), architecture and training, loss function, latent space representation, inference, applications in image generation. Undercomplete Autoencoders, Sparse Autoencoders, Contractive Autoencoders, Denoising Autoencoders, Variational Autoencoders (for generative modelling).

Module V:

(9 Hours)

Limitations of traditional 2D learning environments. Evolution of virtual worlds and immersive technologies. Concepts of Augmented Reality (AR) and Virtual Reality (VR). Introduction to the Metaverse: definition, components, and characteristics. Challenges and opportunities in the Metaverse ecosystem. Overview of AI in emerging domains such as quantum computing.

Practical's

1. Implement Gaussian distribution and visualize univariate and multivariate cases.
2. Implement Gaussian Mixture Model (GMM) using the Expectation-Maximization (EM) algorithm for clustering.
3. Develop a basic Generative Adversarial Network (GAN) for image generation.
4. Implement a Deep Convolutional GAN (DCGAN) and compare its performance with a basic GAN.
5. Implement a Variational Autoencoder (VAE) and visualize latent space representations.
6. Design and implement a denoising autoencoder and analyze reconstruction performance.
7. Apply transfer learning using a pre-trained CNN for feature extraction.
8. Fine-tune a pre-trained convolutional neural network on a custom dataset.
9. Implement Random Forest and evaluate its performance on a classification dataset.
10. Implement a boosting algorithm (AdaBoost/Gradient Boosting) for prediction tasks.
11. Perform stacking or blending of multiple models and compare performance.
12. Develop a mini-project using generative or ensemble learning techniques on a real-world dataset.

Text Book:

1. Foster, D. (2022). Generative Deep Learning. O'Reilly Media.

8819
Shuchi Pritya
Rajiv

2. Koller, D., & Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. MIT Press.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
4. Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
5. Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms. CRC Press.
- 6.

Reference Book:

1. Xiong, J., Hsiang, E. L., He, Z., Zhan, T., & Wu, S. T., 2021. Augmented Reality and Virtual Reality Displays: Emerging Technologies and Future Perspectives. Light: Science & Applications.
2. Mystakidis, S., 2022. Metaverse. Encyclopedia.
3. Gill, S. S., Xu, M., Ottaviani, C., et al. Emerging AI technologies and applications (relevant journal articles).

SSK
Pearl
Shuchi Singh
Ajay
D
Rashmi

Course Title	Advanced Operating System and Shell Programming
Course Code	MCACS - 103
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To provide an in-depth understanding of the Unix/Linux architecture and kernel-shell relationship.
2. To master advanced file system management, including deep-level attributes and inode manipulation.
3. To develop proficiency in complex text processing using stream editors and pattern scanning languages.
4. To explore process management, signals, and job scheduling within a multi-user environment.
5. To engineer robust automation solutions using advanced shell scripting and environment customization.

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1** Students can Analyze the internal structure of the Unix Operating System and its interaction with hardware.
- CO2** Students will be able to execute complex file operations and manage system permissions/security using the command line.
- CO3** Students can design and implement efficient data-processing pipelines using sed, awk, and regular expressions.
- CO4** Students will be able to monitor and control system processes, handling inter-process communication via signals and traps.
- CO5** Students will be able to construct professional-grade shell scripts to automate system administration and repetitive tasks.

Syllabus

Module I:

(9 Hours)

Unix architecture (Kernel and Shell), System boot process, User environment, File types (Ordinary, Directory, Device), Pathnames, Inodes, Hard and Soft links, File attributes, File permissions (absolute and symbolic modes), Commands: chmod, umask, chown, chgrp

884
D. S. G.
Shubhi Singh
Rajiv

Module II:

(9 Hours)

Data pipelines: Standard input/output, Redirection, Pipes, tee command, Basic filters: head, tail, cut, paste, sort, uniq, tr, Stream editing using sed: Addressing, Substitution, Deletion, and scripting

Module III:

(9 Hours)

Grep family (grep, egrep, fgrep), Construction of regular expressions, AWK programming: Syntax, Built-in variables (FS, RS, NR, NF), Associative arrays, Report generation, Logical operations, Data analysis and field-based filtering

Module IV:

(9 Hours)

Process lifecycle: Process creation (fork/exec), Parent-child relationships, Zombie and orphan processes, Monitoring tools: ps, top, Process attributes, Signals: kill, trap, Process scheduling using cron and at

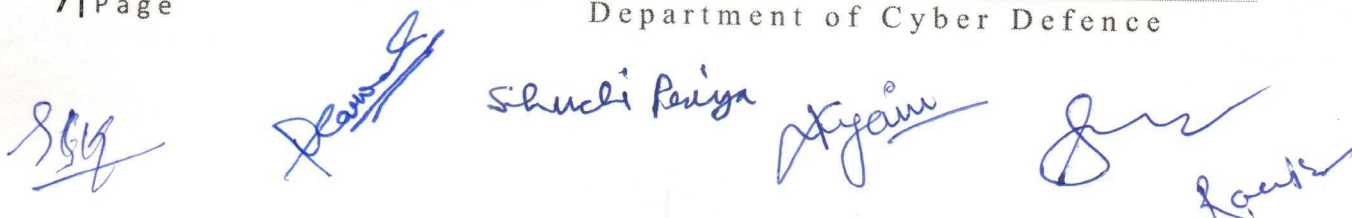
Module V:

(9 Hours)

Shell variables, Positional parameters, set and shift commands, Control structures: if, case, while, until, for loops, Functions, Here documents (<<), Command substitution, Debugging techniques

Practical's

1. Develop a shell script to list all files in a directory that were modified within the last 24 hours and exceed a specified size.
2. Write a script to identify all files in the current directory with "777" permissions and change their permissions to "644".
3. Use cut and paste commands to merge two CSV files based on a common ID field.
4. Write a sed script to convert date format from MM/DD/YYYY to DD-MON-YY in a given file.
5. Develop a script to extract all "ERROR" entries from a system log file, excluding "Handled Errors," and store them in a summary file.
6. Create an awk script to process a student marks file, calculate the average marks, and assign grades to each student.
7. Build a shell script to monitor CPU and memory usage at regular intervals and display an alert if usage exceeds a defined threshold (e.g., 90%).
8. Develop an interactive shell script that lists all processes owned by the current user and allows termination of a selected process using its PID.
9. Write a script to create a temporary directory and use the trap command to ensure its deletion upon script termination or interruption.



10. Develop a shell script using expr and bc to compute the roots of a quadratic equation based on user input.
11. Create a menu-driven shell script to add, delete, or search user entries in the /etc/passwd file (simulation recommended for safety).
12. Write a script to recursively archive all .sh files in a directory structure and store them in a "Backup" folder with a timestamped filename.

Text Book:

1. Advanced Programming in the UNIX Environment by W. Richard Stevens and Stephen A. Rago, Addison-Wesley.
2. Linux Kernel Development by Robert Love, Addison-Wesley Professional.

Reference Book:

1. Your UNIX/LINUX: The Ultimate Guide by Sumitabha Das, Tata McGraw-Hill.
2. Introduction to Unix and Shell Programming by M. G. Venkateshmurthy, Pearson Education India.

SSB ~~Devised~~ Shuchi Priya ~~Again~~ ~~Dr~~
Ravi

Course Title	Cryptography for Cyber Security
Course Code	MCACS - 104
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L-T-P Structure	3-0-2
Classes Per Week	3 Lectures + 1-2 Practical Sessions

Course Objectives:

1. To understand fundamental principles of cryptography and information security.
2. To study classical and modern encryption techniques.
3. To understand public key cryptography and key management systems.
4. To analyze cryptographic protocols and secure communication mechanisms.
5. To apply cryptographic techniques in real-world cyber security applications.
6. To explore emerging trends in cryptography and security systems.

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1 Explain core cryptographic concepts and security goals.
- CO2 Implement classical and modern encryption algorithms.
- CO3 Analyze public key systems and digital signatures.
- CO4 Evaluate cryptographic protocols and their security.
- CO5 Apply cryptographic techniques in practical security scenarios.
- CO6 Understand advanced and emerging cryptographic technologies

Syllabus

Module I:

(9 Hours)

Security goals: Confidentiality, Integrity, Availability, Types of attacks: Passive and active attacks, Cryptographic primitives and services, Mathematical foundations: Number theory, Modular arithmetic, Euclidean algorithm, Primality testing techniques, Integer factorization methods, Euler's totient function, Fermat's theorem and Euler's theorem, Chinese Remainder Theorem.

Module II:

(9 Hours)

Substitution ciphers: Caesar, Monoalphabetic, Transposition ciphers, Playfair cipher, Hill cipher, Cryptanalysis techniques, Limitations of classical encryption methods. Block ciphers:

SSH  Shuchi Raiya   

DES, AES, Feistel structure, Modes of operation: ECB, CBC, CFB, OFB, CTR, Stream ciphers, Strengths and limitations of symmetric encryption.

Module III:

(9 Hours)

Principles of public key cryptography, RSA algorithm, Diffie-Hellman key exchange, ElGamal cryptosystem, Elliptic curve cryptography, Key distribution and key management techniques

Module IV:

(9 Hours)

Cryptographic hash functions: MD5, SHA family, Message authentication codes (MAC, HMAC), Digital signatures: RSA, DSA, Authentication mechanisms, Certificates and Public Key Infrastructure (PKI), Authentication applications, Kerberos protocol, Key management and distribution, Public key distribution mechanisms, X.509 digital certificates and trust models

Module V:

(9 Hours)

Secure communication protocols: SSL/TLS, HTTPS, Email security (PGP), IP security (IPSec), Blockchain fundamentals and cryptography, Emerging trends: Post-quantum cryptography

Practical's

1. Perform basic encryption and decryption using bitwise operations and analyze their impact on plaintext.
2. Implement a simple substitution-based encryption technique and evaluate its security limitations.
3. Design and implement the Caesar Cipher and analyze its vulnerability to brute-force attacks.
4. Implement classical encryption techniques (Substitution and Transposition) and perform basic cryptanalysis.
5. Implement the Hill Cipher using matrix operations and study its effectiveness for block encryption.
6. Simulate the working of a symmetric key encryption algorithm and analyze different modes of operation.
7. Implement a modern symmetric encryption technique and compare its performance with classical methods.
8. Design and implement a public key cryptographic system and demonstrate encryption and decryption.
9. Simulate the Diffie-Hellman Key Exchange mechanism and analyze secure key establishment.
10. Generate and compare message digests using different hash functions and analyze collision resistance.

SSG *Plav* *Shubh Pruthi* *Agarwal* *Sur* *Ravish*

11. Implement a digital signature scheme and verify message authenticity and integrity.
12. Analyze a secure communication protocol (e.g., SSL/TLS) and study its cryptographic components.

Text Book:

1. William Stallings, "Cryptography and Network Security – Principles and Practice", Seventh Edition, Pearson Education, 2017
2. Nina Godbole, Sunit Belapure, "Cyber Security: Understanding Cyber crimes, Computer Forensics and Legal Perspectives", First Edition, Wiley India, 2011.

Reference Book:

1. Behrouz A. Ferouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security", 3rd Edition, Tata Mc Graw Hill, 2015.
2. Charles Pfleeger, Shari Pfleeger, Jonathan Margulies, "Security in Computing", Fifth Edition, Prentice Hall, New Delhi, 2015.

884
Shuchi Singh
Kajal
Rajiv

Course Title	Advanced Digital Forensics
Course Code	MCACS - 105
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To understand advanced digital forensic investigation processes
2. To analyze system, network, and mobile evidence
3. To apply forensic tools and techniques in cybercrime investigation
4. To study legal frameworks and admissibility of digital evidence
5. To explore emerging trends in digital forensics

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1 Perform acquisition and analysis of digital evidence
- CO2 Conduct memory, system, and network forensics
- CO3 Analyze mobile and cloud-based evidence
- CO4 Apply forensic tools and frameworks effectively
- CO5 Interpret cyber laws and present forensic findings

Syllabus

Module I:

(9 Hours)

Principles of digital forensics, Rules and services of digital forensics, Daubert standard, Locard's Exchange Principle, ISO/IEC 27037:2012 guidelines, Forensic investigation lifecycle, Evidence storage formats, Standard operating procedures, Challenges in cybercrime investigation

Module II:

(9 Hours)

Volatile vs non-volatile evidence, Memory acquisition techniques and tools, Live response methodologies, Analysis of running processes, DLLs, handles, network artefacts, Detection of code injection and rootkits, Suspicious processes and drivers, Advanced memory analysis using Volatility, Malware detection in memory, RAM acquisition (LiME), Volatility profiles, Kernel and process analysis

SSK *Deepa* *Sandhi Prasad* *Agarwal* *Sanjay* *Ravi*

Module III:

(9 Hours)

Windows internals and registry analysis, In-memory registry examination, Command-line and process tracking, Windows services and persistence mechanisms, Credential and hash extraction, Prefetch and Shimcache analysis, Event logs and system artefacts, Timeline reconstruction, User activity analysis, File system artefacts, Slack space and Alternate Data Streams (ADS), Linux forensic artefacts

Module IV:

(9 Hours)

Network forensics fundamentals, Packet analysis and traffic monitoring, Network artefacts and intrusion detection, Web forensics and browser artefacts, e-Business and e-Commerce security, Cybercrimes: Cybersquatting, espionage, warfare, terrorism, defamation, social media forensics and digital footprints, Email forensics and tracing, Log analysis and correlation

Module V:

(9 Hours)

Mobile device forensics: Architecture, identifiers, SIM and memory analysis, Logical, file system and physical acquisition, Android memory acquisition, JTAG and chip-off techniques, Cellular networks (GSM, LTE, VoLTE), Cell site analysis and location tracking, Cloud computing architecture, Cloud forensic challenges, Evidence acquisition in virtualized environments, IoT forensics and embedded systems, multi-jurisdiction and data segregation issues

Practical's

1. Perform forensic imaging of a storage device and verify integrity using hashing techniques.
2. Acquire and analyze volatile memory to identify running processes and suspicious activities.
3. Examine Windows registry and system logs to extract forensic artefacts.
4. Analyze Prefetch, Shimcache, and other system artefacts for user activity reconstruction.
5. Capture and analyze network traffic to detect anomalies and potential intrusions.
6. Perform browser and email forensic analysis to extract user activity and communication details.
7. Conduct mobile device forensic acquisition and analyze stored data and metadata.
8. Analyze SIM card data, Call Detail Records (CDR), and IP Detail Records (IPDR).
9. Recover deleted files and hidden data from storage media.
10. Detect and analyze malware behaviour using basic forensic techniques.
11. Perform steganography analysis to identify hidden information in digital files.
12. Conduct a complete forensic investigation case study and prepare a detailed report.

Text Book:

1. Eoghan Casey, Digital Evidence and Computer Crime

8816

Shuchi Priya

Shuchi Priya

Shuchi Priya

Shuchi Priya

Shuchi Priya

1. SO/IEC 27037 Digital Evidence Guidelines
2. NIST Digital Forensics Framework
3. SANS Digital Forensics Resources

8816 ~~David~~ Shuchi Prinja
Hydram

Course Title	Advanced Java Programming
Course Code	MCACS - 201
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To understand the concepts of multiple inheritance and code organization using Interfaces and Packages.
2. To master multi-threaded programming and robust error handling techniques.
3. To develop graphical user interfaces (GUI) using AWT and Swing frameworks.
4. To gain proficiency in server-side programming and database connectivity using JDBC and Servlets/JSP.
5. To explore modern Java features like Generics, Lambda expressions, and Collections for efficient data management.

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1** Students can design scalable applications using Java packages and interfaces.
- CO2** Students will be able to develop high-performance applications using multi-threading and concurrency controls.
- CO3** Students can construct interactive desktop applications with professional GUI components.
- CO4** Students can implement complex data manipulation logic using the Java Collections Framework and Generics.
- CO5** Students can build dynamic, data-driven web applications using JDBC and JSP.

Syllabus

Module I:

(9 Hours)

Interfaces (Defining, extending, and implementing interfaces; Accessing interface variables), Packages (Java API packages, naming conventions, creating and accessing user-defined packages, hiding classes), Advanced OO Concepts (Assertion and Design by Contract)

SSK *Devi* *Shuchi Singh* *Arjun* *Raj* *Raj*

Module II:

(9 Hours)

Multithreaded Programming (Thread life cycle, thread exceptions, priority, synchronization, and implementing the Runnable interface), Managing Errors:(Types of errors, Exception handling Try, Catch, Finally, throwing user-defined exceptions)

Module III:

(9 Hours)

Graphics Programming (Working with AWT and Swing components, Layout Managers, and Event handling),File Management(Input/Output streams, reading/writing files, and random access files)

Module IV:

(9 Hours)

Java Collections Framework (List, Set, Map interfaces and implementation classes), Generic Programming (Generic classes and methods, bounded types), Functional Programming (Lambda Expressions and Iterators in Java)

Module V:

(9 Hours)

Database Connectivity (JDBC), JDBC architecture, establishing connections, executing SQL queries), Web Programming, Introduction to Java Servlets and Java Server Pages (JSP), life cycle, and request/response handling

Practical's

1. Implement a program to demonstrate multiple inheritance by defining Camera and MusicPlayer interfaces and implementing them in a SmartPhone class.
2. Write a program to create a user-defined package com.university.management, containing a Student class with different access modifiers to demonstrate package-level data hiding.
3. Implement a multithreaded program to solve the Producer-Consumer problem using synchronized blocks and wait()/notify() methods for thread communication.
4. Write a program to create a custom exception named InsufficientFundsException for a banking application that triggers when a withdrawal exceeds the account balance.
5. Implement a GUI-based calculator using Java Swing components and ActionListener to perform basic arithmetic operations.
6. Write a program to implement a file copier that reads data from a source file and writes it to a destination file using FileInputStream and FileOutputStream.
7. Implement a program using RandomAccessFile to search for and update specific records in a binary file without overwriting the entire file.
8. Write a generic class Repository<T> that can perform add, remove, and display operations for any object type.
9. Implement a program that uses the ArrayList collection and a custom Comparator to sort and display a list of Student objects based on their marks.

8813  Shuchi Pragn   

10. Write a program to filter a list of products by price using Lambda Expressions and the Java Stream API.
11. Implement a JDBC program to perform CRUD operations (Insert, Select, Update, Delete) on a database table using PreparedStatement.
12. Write a web-based application using Servlets and JSP to implement a user login validation system that interacts with a backend database.

Text Book:

1. Programming with JAVA, 7th Edition by E. Balagurusamy
2. Java: The Complete Reference, 12th Edition by Herbert Schildt (McGraw Hill).

Reference Book:

1. Core Java Volume II–Advanced Features by Cay S. Horstmann.
2. Effective Java by Joshua Bloch (Addison-Wesley Professional).

SSR
Dhansh
Shuchi Arora
Arjun
S
Ravi

Course Title	Advanced Machine Learning and Data Analytics
Course Code	MCACS - 202
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To develop strong foundations in statistical and machine learning methods
2. To understand supervised, unsupervised, and advanced learning techniques
3. To analyze complex datasets using analytical and mathematical approaches
4. To study deep learning and modern ML frameworks
5. To design data-driven solutions for real-world problems

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1** Apply statistical and machine learning models to real-world data
- CO2** Evaluate and optimize predictive models
- CO3** Perform data preprocessing and feature engineering
- CO4** Understand advanced ML techniques including deep learning
- CO5** Analyze data and interpret results for decision-making

Syllabus

Module I:

(9 Hours)

Introduction to machine learning: Definition, importance, types (supervised, unsupervised, reinforcement learning), Applications of machine learning, Data preprocessing: Data cleaning, handling missing values and outliers, Feature scaling and normalization, Introduction to data representation and visualization concepts

Module II:

(9 Hours)

Descriptive statistics, Probability distributions (Binomial, Poisson, Normal), Sampling distributions (Chi-square, t, F), Inferential statistics: Estimation and hypothesis testing, Regression: Linear, polynomial, ridge, lasso, Classification: Logistic regression, k-Nearest Neighbors (k-NN), Decision Trees, Random Forests, Model evaluation metrics: MAE, MSE, RMSE, Accuracy, Precision, Recall, F1-score, ROC-AUC, Model validation: Cross-validation, hyperparameter tuning, Overfitting and underfitting

SS16

Plone

Shuchi Arora

Arjun

Dr

Ravi

Module III:

(9 Hours)

Clustering techniques: K-Means, Hierarchical clustering, Evaluation of clustering results, Dimensionality reduction: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), t-SNE, Association rule learning: Apriori algorithm, Market basket analysis, Pattern discovery and feature extraction

Module IV:

(9 Hours)

Ensemble methods: Bagging, Boosting, Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost – conceptual), Support Vector Machines (SVM): Linear and non-linear SVM, Kernel methods, Neural networks: Fundamentals, Deep learning concepts, CNN and RNN architectures (conceptual), Model optimization and regularization techniques

Module V:

(9 Hours)

Exploratory Data Analysis (EDA), Data visualization principles, Statistical inference and hypothesis testing, Time series analysis: ARIMA and forecasting concepts, Evaluation of time series models, Case studies in real-world data analytics, Model deployment concepts and performance monitoring. Bias-variance tradeoff, Model validation approaches, Active learning, Reinforcement learning fundamentals, Prescriptive analytics and decision-making models, Ethical issues and challenges in AI and data analytics

Practical's

1. Perform data preprocessing and feature selection on a given dataset.
2. Implement regression techniques and evaluate model performance using standard metrics.
3. Apply classification algorithms and compare their accuracy and performance.
4. Perform clustering using different techniques and evaluate clustering results.
5. Implement association rule mining and analyze patterns using support and confidence measures.
6. Design and analyze a simple neural network model for prediction tasks.
7. Explore deep learning concepts by constructing and evaluating a basic multi-layer model.
8. Perform model validation using cross-validation and analyze overfitting and underfitting.
9. Apply dimensionality reduction techniques to improve model performance.
10. Analyze large datasets to identify challenges and performance bottlenecks in big data analytics.
11. Implement active learning strategies for efficient data selection and model improvement.
12. Analyze decision-making processes using reinforcement learning concepts in a simulated environment.

Text Book:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning

8819, Please, check's design, Again, [Signature], [Signature]

2. Kevin P. Murphy, Probabilistic Machine Learning
3. Ian Goodfellow et al., Deep Learning

Reference Book:

1. Hastie, Tibshirani, Friedman, The Elements of Statistical Learning
2. Gareth James et al., An Introduction to Statistical Learning
3. Aurélien Géron, Hands-On Machine Learning

884
Devi
Shuchi Parida
Ajay
Sauri

Course Title	Database and Network Security
Course Code	MCACS - 203
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L-T-P Structure	3-0-2
Classes Per Week	3 Lectures + 1-2 Practical Sessions

Course Objectives:

1. To understand principles of network and database security
2. To analyze network vulnerabilities and defense mechanisms
3. To study authentication, cryptography, and secure communication
4. To explore database security models and privacy techniques
5. To evaluate modern challenges in cyber security

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1 Analyze and secure network infrastructures
- CO2 Apply authentication and cryptographic protocols
- CO3 Identify and mitigate network attacks
- CO4 Design secure database systems
- CO5 Evaluate privacy-preserving mechanisms

Syllabus

Module I:

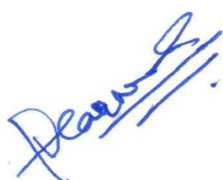
(9 Hours)

Introduction to database security, Threats and vulnerabilities (SQL injection, inference attacks, privilege misuse), Security requirements (Confidentiality, Integrity, Availability), Access control mechanisms: DAC, MAC, RBAC, Authentication and authorization in databases, Database encryption: Data-at-rest and data-in-transit, Key management basics, Database auditing, monitoring, and secure design practices

Module II:

(12 Hours)

Network security models, TCP/IP protocol stack and security layers, IPSec (AH, ESP, IKE), Network vulnerabilities: Media-based and device vulnerabilities, Backdoors, Denial of Service (DoS), Spoofing, Man-in-the-Middle attacks, Replay attacks, Protocol-based attacks, DNS attacks (spoofing, poisoning), ARP poisoning, TCP/IP hijacking, VLAN security, Demilitarized Zone (DMZ), Honeypots, Intrusion Detection and Prevention Systems (IDS/IPS), Protocol analyzers, Internet content filtering, Integrated network security hardware

8816    

Module III:

(7 Hours)

Authentication mechanisms, Kerberos protocol, X.509 authentication service, Port scanning and port knocking (advantages and limitations), Peer-to-peer security, Email security: Distribution lists, key management, Privacy, authentication, message integrity, non-repudiation, Proof of submission and delivery, Confidentiality and anonymity, Pretty Good Privacy (PGP)

Module IV:

(8 Hours)

Database security concepts, Access control mechanisms, Access control models for XML databases, Trust management and trust negotiation, Security in data warehouses and OLAP systems, Database re-engineering for security, Database watermarking for copyright protection, Trustworthy data retention, Damage control, quarantine and recovery in database systems, Hippocratic databases and privacy compliance

Module V:

(9 Hours)

Privacy protection techniques, Backup and anti-theft mechanisms, Web server and host security, Secure web applications, Privacy in database publishing (Bayesian perspective), Location-based access control, Policy enforcement in mobile environments, Emerging trends in security and privacy

Practical's

1. Perform database user management and implement access control policies.
2. Demonstrate SQL injection attacks and implement prevention techniques.
3. Apply database encryption techniques for data-at-rest and data-in-transit.
4. Configure and analyze database auditing and logging mechanisms.
5. Perform port scanning and vulnerability assessment of network systems.
6. Capture and analyze network traffic to identify suspicious activities.
7. Simulate common network attacks such as DoS and spoofing.
8. Configure firewall rules and analyze intrusion detection/prevention systems.
9. Implement secure communication using authentication and encryption mechanisms.
10. Analyze email security and digital signatures.
11. Perform web application security testing and vulnerability analysis.
12. Conduct a case study on database or network security breach and prepare a report.

Text Book:

1. Database System Concepts – Abraham Silberschatz, Henry F. Korth, S. Sudarshan
2. Computer Security: Principles and Practice – William Stallings, Lawrie Brown, Michael E. Whitman

Reference Book:

SSP *Devi* *Shuchi Bhat* *Agarwal* *[Signature]* *[Signature]*

1. Security in Computing – Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies
2. Introduction to Modern Cryptography – Jonathan Katz, Yehuda Lindell, Mike Rosulek

SSC
Dawn
Shuchit Ranga
Kishan
D
Ravi

Course Title	Deep Learning for Computer Vision
Course Code	MCACS - 204
Credit Hours	4 Credits
Total Contact Hours	60 Hours (45 Lectures + 30 Practical)
L–T–P Structure	3–0–2
Classes Per Week	3 Lectures + 1–2 Practical Sessions

Course Objectives:

1. To provide a fundamental understanding of convolutional neural networks and their role in modern AI.
2. To teach students how to preprocess and augment image data for robust model training.
3. To introduce state-of-the-art architectures for object detection, tracking, and segmentation.
4. To demonstrate the power of transfer learning in solving real-world problems with limited data.
5. To prepare students for deploying vision models into production environments.

Course Outcomes:

After successful completion of this course, students will be able to:

- CO1** Students can analyze and select the appropriate neural network architecture for a given vision task.
- CO2** Students can implement deep learning models using industry-standard frameworks like TensorFlow or PyTorch.
- CO3** Students can optimize model performance through rigorous hyperparameter tuning and regularization techniques.
- CO4** Students can evaluate vision systems using metrics like mAP (mean Average Precision) and IoU (Intersection over Union).
- CO5** Students can construct an end-to-end computer vision pipeline from data collection to deployment.

Syllabus

Module I:

(9 Hours)

Introduction to Image Processing (pixels, color spaces, kernels), Linear Algebra for Vision, The Perceptron, Multi-Layer Perceptrons (MLP)(Activation Functions (ReLU, Softmax), Loss Functions and Backpropagation).

Module II:

(9 Hours)

The Convolutional Layer, Pooling and Stride, Padding, Building a CNN from scratch, Standard Architectures (LeNet, AlexNet, VGG), Batch Normalization and Dropout.

[Handwritten signatures and initials]

Module III:

(9 Hours)

ResNet and Skip Connections; Inception Modules; MobileNet for Edge Devices; Fine-tuning vs. Feature Extraction, Data Augmentation techniques to prevent overfitting.

Module IV:

(9 Hours)

Localization vs. Detection; Regional CNNs (R-CNN, Fast R-CNN, Faster R-CNN); YOLO (You Only Look Once) and SSD; Semantic vs. Instance Segmentation; U-Net architectures.

Module V:

(9 Hours)

Autoencoders, Generative Adversarial Networks (GANs), Style Transfer, Image Captioning with RNNs/LSTMs, Deploying models to Cloud and Mobile (TensorFlow Lite).

Practical's

1. Write a script to perform geometric transformations (rotation, scaling) and color filtering using OpenCV.
2. Implement a single-neuron classifier to distinguish between two shapes.
3. Build and train an MLP to recognize handwritten digits with >95% accuracy.
4. Design a 3-layer Convolutional Network for the CIFAR-10 dataset.
5. Experiment with different learning rates and optimizers (Adam vs. SGD) on a vision model.
6. Use a pre-trained VGG16 model to classify a custom dataset (e.g., types of flowers).
7. Implement a pipeline that uses flipping, rotation, and brightness shifts to double a small dataset's size.
8. Use a pre-trained YOLO model to detect and count people in a video stream.
9. Implement a basic U-Net to mask specific objects in an image.
10. Create an application that applies the "artistic style" of a famous painting to a webcam feed.
11. Build a system that identifies registered faces using Siamese Networks or FaceNet.
12. Export a trained model to a web application using Flask or Streamlit.

Text Book:

1. Deep Learning for Computer Vision by Rajalingappaa Shanmugamani.
2. Deep Learning with Python by François Chollet.

Reference Book:

1. Deep Learning by Ian Goodfellow et al.
2. Computer Vision: Algorithms and Applications by Richard Szeliski.

The bottom of the page features several handwritten signatures in blue ink. From left to right, there are approximately six distinct signatures, some of which appear to be names like 'Sanku', 'Rajini', and 'Ravi'.