LNCT University B.TECH-AIML

V Sem

Subject: Agile Methodology AL- 501

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Understand Agile software development principles and practices
CO2	Apply Scrum values and practices to relevant software application
CO3	Analyse product management agile approach for real life problem
CO4	Assess software for agile requirements and architecture
CO5	Design risk management plan with agile approach

UNIT I AGILE SOFTWARE DEVELOPMENT Basics and Fundamentals of Agile Process Methods, Values of Agile, Principles of Agile, stakeholders, Challenges. Lean Approach: Waste Management, Kaizen and Kanban, add process and products add value. Roles related to the lifecycle, differences between Agile and traditional plans, differences between Agile plans at different lifecycle phases. Testing plan links between testing, roles and key techniques, principles, understand as a means of assessing the initial status of a project/ How Agile helps to build quality

UNIT II AGILE AND SCRUM PRINCIPLES Agile Manifesto, Twelve Practices of XP, Scrum Practices, Applying Scrum. Need of scrum, working of scrum, advanced Scrum Applications, Scrum and the Organization, scrum values

UNIT III AGILE PRODUCT MANAGEMENT Communication, Planning, Estimation Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement, Escalating issue. Quality, Risk, Metrics and Measurements.

UNIT IV AGILE REQUIREMENTS AND AGILE TESTING User Stories, Backlog Management. Agile Architecture: Feature Driven Development. Agile Risk Management: Risk and Quality Assurance, Agile Tools. Agile Testing Techniques, Test-Driven Development, User Acceptance Test

UNIT V SCALING AGILE FOR LARGE PROJECTS Agile Metrics and Measurements, The Agile approach to project estimating Agile Control: the 7 control parameters. Agile approach to Risk management, The Agile approach to Configuration Management, The Atern Principles, AternPhilosophy, Refactoring, Continuous integration, Automated Build Tools.

References

1.Robert C. Martin, Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.

2. Kenneth S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison Wesley, 2012.

3. James Shore and Shane Warden, The Art of Agile Development, O'Reilly Media, 2007.

4. Craig Larman, ―Agile and Iterative Development: A manager's Guide, Addison-Wesley, 2004.

5. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2001.

6. Cohn, Mike, Agile Estimating and Planning, Pearson Education, 2006.

7. Cohn, Mike, User Stories Applied: For Agile Software Development Addison Wisley, 2004.

E-references

1. "Agile Software Development", https://www.edx.org/course/agile-software-development

2. "Agile Software Development", https://www.coursera.org/learn/agile-software-development

3. "The Complete Guide to Agile Software Development" https://clearbridgemobile.com/completeguideagile-software-development

4. "Agile Fundamentals Ebook: A Complete Guide for Beginners", https://agileken.com/agilefundamentals-ebook/

Suggested List of Experiments

- Prepare a simple project with its objectives vision statement use case and UML diagrams, banking system, E-Commerce, Healthcare
- Design project release map, user stories for the same
- Design product road map for the proposed project
- Design story mapping for the proposed project
- Compile the product blocking for the proposed project
- Compile the spirit backlog for the proposed project

LNCT University B.TECH-AIML

Modern Machine Learning – From Ensemble to Insight (AL 502)

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Apply resampling, model selection, and regularization techniques in predictive modeling.
CO2	Implement and evaluate tree-based methods like decision trees, bagging, and boosting.
CO3	Understand and apply linear Support Vector Machine classifiers.
CO4	Use non-linear SVMs and multi-class classification strategies.
CO5	Apply unsupervised learning techniques such as PCA, clustering, and HMMs.

Course contents

UNIT – I

Resampling Methods, Model Selection and Regularization: Cross-validation, leave-one-out cross validation, k-fold cross-validation, the bootstrap, subset
selection, shrinkage methods, ridge and lasso regression, dimension reduction methods, principal components regression, partial least square

UNIT – II

Tree Based Methods: Advantages and disadvantages of trees, regression Trees, classification trees, bagging, random forest, boosting

UNIT – III

Support Vector Machine: Maximum margin classifier, classification using a separating hyperplane, the maximal margin classifier, support vector classifier,
Support vector machines, classification with non-linear decision boundaries, support vector machine, one-versus-one classification, one-versus-many classification

UNIT – IV

Unsupervised Learning: Principle component analysis, what are principal components, clustering methods, k-means clustering, hierarchical clustering,

UNIT – V

Independent component analysis, latent semantic indexing, Markov Models, Hidden Markov Models

REFERENCES:

1. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer-Verlag New York Inc., 2nd Edition, 2011.
2. Tom M. Mitchell, “Machine Learning”, McGraw Hill Education, First edition, 2017.
3. Ian Goodfellow and YoshuaBengio and Aaron Courville, “Deep Learning”, MIT Press, 2016
4. Aurelien Geon, “Hands-On Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems”,
   Shroff/O'Reilly; First edition (2017)

LNCT University B.TECH-AIML

List of Experiment

1. Implement k-Fold Cross-Validation on a regression dataset.
2. Perform Leave-One-Out Cross-Validation and compare results with k-fold CV.
3. Apply Bootstrap Sampling to estimate the accuracy of a model.
4. Perform Subset Selection on a linear regression model and evaluate performance.
5. Implement Ridge Regression and plot coefficient shrinkage.
6. Apply Lasso Regression and observe variable selection behavior.
7. Perform Principal Component Regression (PCR) and compare with linear regression.
8. Apply Partial Least Squares (PLS) for dimensionality reduction and regression.
9. Build and Evaluate a Regression Tree using a real-world dataset.
10. Build and Evaluate a Classification Tree using a labeled dataset.
11. Implement Bagging and compare its performance with a single decision tree.
12. Implement Random Forest and analyze feature importance.
13. Apply Boosting (e.g., AdaBoost or Gradient Boosting) to a classification problem.
14. Train a Linear SVM classifier and visualize the decision boundary.
15. Implement Non-Linear SVM using kernel trick (RBF kernel) on a 2D dataset.
16. Perform One-vs-One and One-vs-All Classification using SVM.
17. Apply Principal Component Analysis (PCA) on a high-dimensional dataset.
18. Implement K-Means Clustering and visualize clusters on 2D/3D data.
19. Apply Hierarchical Clustering and plot a dendrogram.
20. Simulate a Hidden Markov Model (HMM) and decode a sequence using the Viterbi algorithm

LNCT University B.TECH-AIML

Deep Learning and Neural Networks with TensorFlow (AL503)

COURSE OUTCOMES:
After Completing the course student should be able to

CO1	Understand the structure and components of neural networks, including various types of neurons
CO2	Apply training algorithms such as gradient descent and backpropagation in feed-forward neural
CO3	Construct and manage computational graphs using TensorFlow for implementing machine learning
CO4	Develop and evaluate deep neural network models using the Keras and other related python
CO5	Create real world applications with LSTM and GAN deep learning frameworks.

Objectives: The objective of this course is to teach students the basic concepts of TensorFlow, Tensors, Neural Network, Implementing Neural Networks, Deep Learning

Course contents

UNIT – I Tensorflow and Tensors About TensorFlow, TensorFlow Architecture, TensorFlow 1.x vs 2.x, Setting Up TensorFlow, Introduction to Tensor: Tensor, Real-Word Analogy, Tensor vs. Numpy Array, Tensor Data Types, Ranks, Shape, Attributes, Tensor Operations & Broadcasting: Element-wise Operation, Basic Arithmetic, Matrix Manipulation, Broadcasting, Tensor Slicing, Indexing, and Reshaping, expand_dims(), squeeze(), TensorFlow Variables, Automatic Differentiation, tf.GradientTape(), Functions and Graphs: @tf.function, eager execution, graph execution

UNIT – II Neural Networks What is Neural Network, Neural Networks vs. Traditional Programming, Biological vs Artificial Neurons, Architecture of Neural Networks, Structure of Neural Network, Input Layer, Hidden Layer, Output Layer, Activation Functions, Sigmoid Function, Tanh Function, ReLU, Softmax Function, Forward Propagation, How Forward Propagation Works? Weights and Bias, Nodes and Layers, Loss Functions, MSE, Cross Entropy, Common Loss Functions, Gradient Descent & Learning Rate, Gradients, Local vs Global Minimum, Local Maximum, Saddle Point, Optimizers, SGD, Momentum, Adam Optimizer, RMS Prop, Backpropagation, Chain Rule, Gradients in Backpropagation, Epochs, Batches, Iterations

UNIT – III Implementing and Training Neural Network Data Pre-processing, Missing Data, Feature Scaling, Encoding Categorical Data, Shuffle Data, Train, Validation & Test Split, Feature Engineering, Feature Selection, Feature Transformation, Feature Creation, Keras for Neural Networks: Build a Neural Network with Keras, Regularization Techniques, Overfitting, L1 Regularization, L2 Regularization, Dropout, Batch Normalization, Early Stopping, Model Checkpointing, Hyperparameter Tuning, Layers, Neurons per Layer, Learning Rate, Batch Size, Grid Search, Random Search, Model Evaluation, Confusion Matrix, Accuracy, Precision, Recall, F1 Score

Unit-IV About Deep Learning, CNN Overview, How CNN See Image, CNN Architecture, Convolution, Filter, Activation Function, Pooling, Stride, Padding, Flattening, Softmax Layer, Applications of CNN, RNN, RNN Architecture, Vanishing Gradient Problem,

Unit-V Deep Learning cntd. LSTM, How LSTM Works, Foget, Inout, Output Gates, Transfer Learning, GANs (Generative Adversarial Networks), How GANs works, Generator and Discriminator, Real World Applications of Deep Learning

Reference Books

1. Ian Goodfellow and YoshuaBengio and Aaron Courville, “Deep Learning”, MIT Press, 2016
2. Aurelien Geon, “Hands-On Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems”,
   Shroff/O'Reilly; First edition (2017)

LNCT University B.TECH-AIML

List of Experiment

1. Implement a basic artificial neuron using Python.
2. Build a Linear Perceptron for binary classification.
3. Implement different activation functions (Sigmoid, Tanh, ReLU) and plot their curves.
4. Design a feed-forward neural network from scratch using NumPy.
5. Visualize the limitation of linear neurons on non-linearly separable data.
6. Build a Softmax output layer for multi-class classification.
7. Implement cross-entropy loss function and compute error manually.
8. Demonstrate the effect of different learning rates using Gradient Descent.
9. Implement Backpropagation algorithm manually for a small neural network.
10. Compare stochastic, batch, and mini-batch gradient descent on a dataset.
11. Train a neural network on a dataset using TensorFlow and evaluate its accuracy.
12. Construct and visualize a TensorFlow computational graph.
13. Use TensorFlow placeholders and variables to build a simple regression model.
14. Perform linear regression using TensorFlow on a real-world dataset.
15. Train a logistic regression classifier using TensorFlow.
16. Build and train a neural network using Keras on the MNIST dataset.
17. Apply data preprocessing techniques like normalization and standardization in Keras.
18. Evaluate model performance using accuracy, confusion matrix, and loss curves in Keras.
19. Demonstrate overfitting and underfitting using different model architectures.
20. Apply dropout and regularization to prevent overfitting in a Keras model.

LNCT University B.TECH-AIML

Cyber Security (AL- 504)

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Students will be able to understand the concept of cybercrime
CO2	Students will be able to analyze different types of cyber offenses
CO3	Students will be able to examine the legal provisions related to cybercrimes under the IT
CO4	Students will be able to compare the Indian Evidence Act, 1872 with the IT Act, 2000,
CO5	Students will be able to identify and explain common cybercrime tools and methods

Course Contents

UNIT 1: Introduction to Cyber Crime :: Introduction of Cyber Crime, Challenges of cyber crime, Classifications of Cybercrimes: EMail Spoofing, Spamming,
Internet Time Theft, Salami attack/Salami Technique,
UNIT 2: Categories and Perception of Cyber Crimes :: Web jacking, Online Frauds, Software Piracy, Computer Network Intrusions, Password Sniffing,
Identity Theft, cyber terrorism, Virtual Crime, Perception of cyber criminals: hackers, insurgents and extremist group etc. session hijacking.
UNIT 3 : Cyber Crime and Legal Framework :: Cyber Crime and Criminal justice: Concept of Cyber Crime and the IT Act, 2000, Hacking, Teenage Web Vandals,
Cyber Fraud and Cheating, Defamation, Harassment and E-mail Abuse, Other IT Act Offences, Monetary Penalties, jurisdiction and Cyber Crimes,
Nature of Criminality, Strategies to tackle Cyber Crime and Trends.
UNIT 4 : Cyber Evidence and Legal Admissibility :: The Indian Evidence Act of 1872 v. Information Technology Act, 2000: Status of Electronic Records
as Evidence, Proof and Management of Electronic Records; Relevancy, Admissibility and Probative Value of E-Evidence, Proving Digital Signatures,
Proof of Electronic Agreements, Proving Electronic Messages.
UNIT 5 : Cybercrime Tools and Techniques :: Tools and Methods in Cybercrime: Proxy Servers and Anonymizers, Password Cracking, Key loggers and Spyware,
virus and worms, Trojan Horses, Backdoors, DoS and DDoS Attacks , Buffer and Overflow, Attack on Wireless Networks, Phishing : Method of Phishing,
Phishing Techniques.

REFERENCES:

1. Principles of Cyber crime, Jonathan Clough Cambridge University Press
2. John R. Vacca, Computer Forensics:Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005
3. Cyber Law Simplified, VivekSood, Pub: TMH.
4. Cyber Security by Nina Godbole, SunitBelapure Pub: Wiley-India
5. Information Warfare: Corporate attack and defense in digital world, William Hutchinson, Mathew Warren, Elsevier.
6. Cyber Laws and IT Protection, Harish Chander, Pub:PHI.

LNCT University B.TECH-AIML

AL 506 Minor Project Syllabus

The minor project focuses on introducing students to the fundamentals of full stack development. Students will begin by understanding the architecture of web applications, including how the frontend and backend communicate. They will then design a simple user interface that captures and displays data dynamically. The backend will include basic server setup, data handling, and implementation of CRUD (Create, Read, Update, Delete) operations with a connected database. Emphasis will be placed on developing clean code, maintaining basic security practices, and validating user input. Finally, students will test the application for core functionalities, deploy it to a hosting platform, and present a concise documentation highlighting their learning journey.