**Department of Computer Science & Enginnering**

**COMPUTER NETWORKS LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Study of Network devices in detail and connect the computers in Local Area Network. |
| Exercise-2 | Write a Program to implement the data link layer farming methods such as  i) Character stuffing ii) bit stuffing |
| Exercise-3 | Write a Program to implement data link layer farming method checksum. |
| Exercise-4 | Write a program for Hamming Code generation for error detection and correction. |
| Exercise-5 | Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP |
| Exercise-6 | Write a Program to implement Sliding window protocol for Goback N. |
| Exercise-7 | Write a Program to implement Sliding window protocol for Selective repeat. |
| Exercise-8 | Write a Program to implement Stop and Wait Protocol. |
| Exercise-9 | Write a program for congestion control using leaky bucket algorithm |
| Exercise-10 | Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph. |
| Exercise-11 | Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes). |
| Exercise-12 | Write a Program to implement Broadcast tree by taking subnet of hosts |
| Exercise-13 | Wireshark  i. Packet Capture Using Wire shark  ii. Starting Wire shark  iii. Viewing Captured Traffic  iv. Analysis and Statistics & Filters. |
| Exercise-14 | How to run Nmap scan |
| Exercise-15 | Operating System Detection using Nmap |
| Exercise-16 | Do the following using NS2 Simulator  i. NS2 Simulator-Introduction  ii. Simulate to Find the Number of Packets Dropped  iii. Simulate to Find the Number of Packets Dropped by TCP/UDP  iv. Simulate to Find the Number of Packets Dropped due to Congestion  v. Simulate to Compare Data Rate& Throughput |

**Department of Computer Science & Enginnering**

**CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DevOps (R20)**

**(Skill Oriented Course)**

**LIST OF EXPERIMENTS**

**Size – (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Get an understanding of the stages in software development lifecycle, the process models, values and principles of agility and the need for agile software development. This will enable you to work in projects following an agile approach to software development. |
| Exercise-2 | Get a working knowledge of using extreme automation through XP programming practices of test first development, refactoring and automating test case writing. |
| Exercise-3 | It is important to comprehend the need to automate the software development lifecycle stages through DevOps. Gain an understanding of the capabilities required to implement DevOps, continuous integration and continuous delivery practices. |
| Exercise-4 | Configure the web application and Version control using Git using Git commands and version control operations. |
| Exercise-5 | Configure a static code analyzer which will perform static analysis of the web application code and identify the coding practices that are not appropriate. Configure the profiles and dashboard of the static code analysis tool |
| Exercise-6 | Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit test cases and deploy the application to a web application server like Tomcat. |
| Exercise-7 | Configure the Jenkins tool with the required paths, path variables, users and pipeline views. |
| Exercise-8 | Configure the Jenkins pipeline to call the build script jobs and configure to run it whenever there is a change made to an application in the version control system. Make a change to the background color of the landing page of the web application and check if the configured pipeline runs. |
| Exercise-9 | Create a pipeline view of the Jenkins pipeline used in Exercise 8. Configure it with user defined messages. |
| Exercise-10 | In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static analysis of code. |
| Exercise-11 | In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static unit testing. |
| Exercise-12 | In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for code coverage. |

**Department of Computer Science & Enginnering**

**DATA WAREHOUSING AND DATA MINING LAB (R20)**

**LIST OF EXPERIMENTS**

**Size – (3\*4) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Creation of a Data Warehouse.   * Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool,¬ Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.,) * Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for¬ any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc). * Write ETL scripts and implement using data warehouse tools. * Perform Various OLAP operations such slice, dice, roll up, drill up and pivot |
| Exercise-2 | Explore machine learning tool “WEKA”   * Explore WEKA Data Mining/Machine Learning Toolkit. * Downloading and/or installation of WEKA data mining toolkit. * Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface. * Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify¬ panel, Cluster panel, Associate panel and Visualize panel) * Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather¬ dataset, Iris dataset, etc.) * Load each dataset and observe the following:¬   1. List the attribute names and they types  2. Number of records in each dataset  3. Identify the class attribute (if any)  4. Plot Histogram  5. Determine the number of records for each class  6. Visualize the data in various dimensions |
| Exercise-3 | Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets   * Explore various options available in Weka for preprocessing data and apply Unsupervised filters¬ like Discretization, Resample filter, etc. on each dataset * Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values. * Study the rules generated. Apply different discretization filters on numerical attributes and run the¬ Apriori association rule algorithm. Study the rules generated. * Derive interesting insights and observe the effect of discretization in the rule generation process.¬ |
| Exercise-4 | Demonstrate performing classification on data sets   * Load each dataset into Weka and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic. * Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix. * Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbour¬ classification. Interpret the results obtained. * Plot RoC Curves * Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and¬ deduce which classifier is performing best and poor for each dataset and justify. |
| Exercise-5 | Demonstrate performing clustering of data sets   * Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters). * Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights. * Explore other clustering techniques available in Weka. * Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain. |
| Exercise-6 | Demonstrate knowledge flow application on data sets   * Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms * Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm * Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree |
| Exercise-7 | Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations |
| Exercise-8 | Write a java program to prepare a simulated data set with unique instances. |
| Exercise-9 | Write a Python program to generate frequent item sets / association rules using Apriori algorithm |
| Exercise-10 | Write a program to calculate chi-square value using Python. Report your observation |
| Exercise-11 | Write a program of Naive Bayesian classification using Python programming language. |
| Exercise-12 | . Implement a Java program to perform Apriori algorithm |
| Exercise-13 | Write a program to cluster your choice of data using simple k-means algorithm using JDK |
| Exercise-14 | Write a program of cluster analysis using simple k-means algorithm Python programming language. |
| Exercise-15 | Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python |
| Exercise-16 | Visualize the datasets using matplotlib in python.(Histogram, Box plot, Bar chart, Pie chart etc.,) |

Name Board: **Size – (1\*1) Count- 1**

M Sowmya

Programmer