**LEARNING PROXIMITY RELATIONS FOR FEATURE SELECTION**

**INTRODUCTION:**

Showing late advances in the machine learning strategies to best in class discrete choice models, we develop an approach to manage incite the extraordinary and complex fundamental authority system of a boss (DM), which is depicted by the DM's needs and attitudinal character, close by the qualities relationship, to give a few illustrations. On the reason of incredible slant information as pairwise examinations of decisions, our method tries to provoke a DM's slant model similarly as the parameters recently discrete choice models. To this end, we decrease our learning ability to a constrained non-direct improvement issue. Our learning strategy is a fundamental one that ponders the relationship among the properties nearby the needs and the intriguing attitudinal character of a DM. The test results on standard benchmark datasets suggest that our technique is normally captivating and successfully interpretable and also forceful to best in class systems.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS**  **LIST OF ABBREVIATIONS**  **LIST OF TABLES** | i  v  vii  xi  xii |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. OBJECTIVE   3. EXISTING SYSTEM   1.3.1EXISTINGSYSTEMDISADVANTAGES  1.3.2 LITERATURE SURVEY  1.4 PROPOSED SYSTEM  1.4.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.2.3 MODULE DIAGRAM  2.2.4GIVEN INPUTAND EXPECTED OUTPUT  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 : SYSTEM DESIGN**  **4.1 general**  4.1.1 activity diagram  4.1.2 USE CASE DIAGRAM  4.1.3 DATA FLOW DIAGRAM  4.1.4SEQUENCE DIAGRAM  4.1.5 COLLABORATION DIAGRAM  4.1.6CLASS DIAGRAM  4.1.7 SYSTEM ARCHITECTURE  4.1.8 OBJECT DIAGRAM  4.1.9 STATE DIAGRAM  4.1.10 COMPONENT DIAGRAM  4.1.11 E-R DIAGRAM  4.2 DATABASE DESIGN (ALL LEVEL |  |
| 5. | **CHAPTER 5 :SOFTWARE SPECIFICATION**  5.1 general  FRONT END  5.2 features of java  5.2.1 the java framework  5.2.2 objective of java  5.2.3 componets of java frmework  5.3 features of the common language runtime  5.4 j2ee  5.4.1 EVALUTION OF J2EE  5.4.2 THE J2EE DATA ARCHITECTURE  BACK END  5.6 features of SQL SERVER |  |
| 6. | **CHAPTER 6 : IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION  6.3 DATA BASE TABLE STRUCTURE |  |
| 7. | **CHAPTER 7 : SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 : SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **APPLICATIONS AND FUTURE ENHANCEMENT**  9.1 GENERAL  9.2 APPLICATIONS  9.3 FUTURE ENHANCEMENTS |  |
| **10** | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 2.3.2 | Module Diagram |  |
| 4.2 | Activity Diagram |  |
| 4.3 | Use case Diagram |  |
| 4.4 | Data flow diagram |  |
| 4.5 | Sequence diagram |  |
| 4.6 | Collaboration diagram |  |
| 4.7 | Class diagram |  |
| 4.8 | Architecture Diagram |  |
| 4.9 | State Diagram |  |
| 4.1 | Component Diagram |  |
| 4.12 | E-R Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | Represents a collection of similar entities grouped together. |
| 2. | Association | name  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor | Class A  Class A  Class B  Class B | It aggregates several classes into a single classes. |
| 5. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processs. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | F inal state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Use case |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which is a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or acion. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**LIST OF ABBREVATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| 1**.** | DB | DataBase |
| 2. | JVM | Java Virtual Machine |
| 3. | JSP | Java Server Page |
| 4. | CB | Collective Behavior |
| 5. | SD | Social Dimension |
| 6. | JRE | Java Runtime Environment |
| 7. | SSD | Sparse Social Dimension |
| 8. | LGP | Line Graph Partition |

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL:**

Showing late advances in the machine learning strategies to best in class discrete choice models, we develop an approach to manage incite the extraordinary and complex fundamental authority system of a boss (DM), which is depicted by the DM's needs and attitudinal character, close by the qualities relationship, to give a few illustrations. On the reason of incredible slant information as pairwise examinations of decisions, our method tries to provoke a DM's slant model similarly as the parameters recently discrete choice models. To this end, we decrease our learning ability to a constrained non-direct improvement issue. Our learning strategy is a fundamental one that ponders the relationship among the properties nearby the needs and the intriguing attitudinal character of a DM. The test results on standard benchmark datasets suggest that our technique is normally captivating and successfully interpretable and also forceful to best in class systems.

**3. Scope of the Project**

The physical significance misfortune, highlight choice plans to choose a highlight subset which can exceptionally protect the first properties of tests for learning errands. Past reducing the impact of condemnation of dimensionality and accelerating the learning preparing, there is a recognizing advantage for highlight choice. It's advantageous to find the potential affiliation among tests by imagining the first traits of focuses in low-measurement. A run of the mill application for highlight choice is Microarray Analysis.

**4. PROBLEM STATEMENT**

Subsequently, highlight choice issue is diminished to angathering learning issue of joining numerous feeble classifiers into an advanced solid classifier. The proposed nearness relations learning calculation performs highlight determination as indicated by the proposed measure of highlight quality utilizing. At long last, in view of the proposed highlight choice calculation, we give a hypothetical examination of the speculation blunder bound which is autonomous on the information dimensionality. This is an exceptionally reassuring result with admiration to "revile of dimensionality" that additionally approves the adequacy of our proposed strategy.

**5. EXISTING SYSTEM**

**5.1 Existing System:**

Most existing popular text mining and classification methods have adopted term-based approaches. However, they have all suffered from the problems of polysemy and synonymy.

Over the years, there has been often held the hypothesis that pattern-based methods should perform better than term-based ones in describing user preferences.

**DRAWBACKS:-**

Response time reduced.

High time complexity

Website user may reduced.

**LITERATURE SURVEY:**

Title : Dynamic Infinite Relational Model for Time-varying Relational Data

Analysis.

Author : Katsuhiko Ishiguro, Tomoharu Iwata Naonori Ueda, Joshua Tenenbaum.

Year : 2008

**Description:**

We propose a new probabilistic model for analyzing dynamic evolutions of relational data, such as additions, deletions and split & merge, of relation clusters like communities in social networks. Our proposed model abstracts observed time- varying object-object relationships into relationships between object clusters. We extend the infinite Hidden Markov model to follow dynamic and time-sensitive changes in the structure of the relational data and to estimate a number of clusters simultaneously. We show the usefulness of the model through experiments with synthetic and real-world data sets.

**LITERATURE SURVEY:**

Title : A probabilistic model of cross-categorization

Author : Patrick Shafto a, Charles Kemp b, Vikash Mansinghka , Joshua

Year : 2012

**Description:**

Most natural domains can be represented in multiple ways: we can categorize foods in terms of their nutritional content or social role, animals in terms of their taxonomic groupings or their ecological niches, and musical instruments in terms of their taxonomic categories or social uses. Previous approaches to modeling human categorization have largely ignored the problem of cross-categorization, focusing on learning just a single system of categories that explains all of the features. We also formalize two commonly proposed alternative explanations for cross-categorization behavior: a features-first and an objects-first approach. The features- first approach suggests that cross-categorization is a consequence of attentional processes, where features are selected by an attentional mechanism first and categories are derived second. The objects-first approach suggests that cross-categorization is a consequence of repeated, sequential attempts to explain features, where categories are derived first, then features that are poorly explained are recategorized. We present two sets of simulations and experiments testing the models’ predictions about human categorization. We find that an approach based on joint inference provides the best fit to human categorization behavior, and we suggest that a full account of human category learning will need to incorporate something akin to these capabilities.

**LITERATURE SURVEY:**

Title : Learning Multiple Tasks with Kernel Methods

Author : Theodoros Evgeniou, Charles A. Micchelli, Massimiliano Pontil

Year : 2013

**Description:**

We study the problem of learning many related tasks simultaneously using kernel methods and regularization. The standard single-task kernel methods, such as support vector machines and regularization networks, are extended to the case of multi-task learning. Our analysis shows that the problem of estimating many task functions with regularization can be cast as a single task learning problem if a family of multi-task kernel functions we define is used. These kernels model relations among the tasks and are derived from a novel form of regularizers. Specific kernels that can be used for multi-task learning are provided and experimentally tested on two real data sets. In agreement with past empirical work on multi-task learning, the experiments show that learning multiple related tasks simultaneously using the proposed approach can significantly outperform standard single-task learning particularly when there are many related tasks but few data per task.

**LITERATURE SURVEY:**

Title : A Framework for Learning Predictive Structures from

Multiple Tasks and Unlabeled Data

Author : Rie Kubota Ando, Tong Zhang tzhang.

Year : 2013

**Description:**

One of the most important issues in machine learning is whether one can improve the performance of a supervised learning algorithm by including unlabeled data. Methods that use both labeled and unlabeled data are generally referred to as semi-supervised learning. Although a number of such methods are proposed, at the current stage, we still don’t have a complete understanding of their effectiveness. This paper investigates a closely related problem, which leads to a novel approach to semi-supervised learning. Specifically we consider learning predictive structures on hypothesis spaces (that is, what kind of classifiers have good predictive power) from multiple learning tasks. We present a general framework in which the structural learning problem can be formulated and analyzed theoretically, and relate it to learning with unlabeled data. Under this framework, algorithms for structural learning will be proposed, and computational issues will be investigated. Experiments will be given to demonstrate the effectiveness of the proposed algorithms in the semi-supervised learning setting.

**LITERATURE SURVEY:**

Title : Multitask Learning

Author : Aarti Singh, ManishaMalhotra.

Year : 2013

**Description:**

Multitask Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better. This paper reviews prior work on MTL, presents new evidence that MTL in backprop nets discovers task relatedness without the need of supervisory signals, and presents new results for MTL with k-nearest neighbor and kernel regression. In this paper we demonstrate multitask learning in three domains. We explain how multitask learning works, and show that there are many opportunities for multitask learning in real domains. We present an algorithm and results for multitask learning with case-based methods like k-nearest neighbor and kernel regression, and sketch an algorithm for multitask learning in decision trees. Because multitask learning works, can be applied to many different kinds of domains, and can be used with different learning algorithms, we conjecture there will be many opportunities for its use on real-world problems.

**LITERATURE SURVEY:**

Title : Feature Selection for Ordinal Text Classification

Author : Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani

Year : 2014

**Description:**

Ordinal classification (also known as ordinal regression) is a supervised learning task that consists of automatically determining the implied rating of a data item on a fixed, discrete rating scale. This problem is receiving increased attention from the sentiment analysis / opinion mining community, due to the importance of automatically rating increasing amounts of product review data in digital form. As in other supervised learning tasks such as (binary or multiclass) classification, feature selection is needed in order to improve efficiency and to avoid overfitting. However, while feature selection has been extensively studied for other classification tasks, is has not for ordinal classification. In this paper we present four novel feature selection metrics that we have specifically devised for ordinal classification, and test them on two datasets of product review data against three metrics previously known from the literature, using two learning algorithms from the “support vector regression” tradition. The experimental results show that all four proposed metrics largely outperform all of the three baseline techniques, on both datasets and for both learning algorithms.

**LITERATURE SURVEY:**

Title : Adaptive oating search methods in feature selection

Author : P. Somol, P. Pudil, J. Novovi\_cov\_a, P. Pacl\_õk.

Year : 2013

**Description:**

A new suboptimal search strategy for feature selection is presented. It represents a more sophisticated version of ``classical'' ¯oating search algorithms (Pudil et al., 1994), attempts to remove some of their potential de®ciencies and facilitates a solution even closer to the optimal one.

**LITERATURE SURVEY:**

Title : On p-norm Path Following in Multiple Kernel Learning for Non-linear

feature Selection

Author : Pratik Jawanpuria, Manik Varma, J. Saketha Nath.

Year : 2009

**Description:**

Our objective is to develop formulations and algorithms for efficiently computing the feature selection path – i.e. the variation in classification accuracy as the fraction of selected features is varied from null to unity. We propose a novel conjecture which states that, for certain lp-MKL formulations, the number of features selected in the optimal solution monotonically decreases as p is decreased from an initial value to unity. We prove the conjecture, for a generic family of kernel target alignment based formulations, and show that the feature weights themselves decay (grow) monotonically once they are below (above) a certain threshold at optimality. This allows us to develop a path following algorithm that systematically generates optimal feature sets of decreasing size. The proposed algorithm sets certain feature weights directly to zero for potentially large intervals of p thereby reducing optimization costs while simultaneously providing approximation guarantees. We empirically demonstrate that our formulation can lead to classification accuracies which are as much as 10% higher on benchmark data sets not only as compared to other lp-MKL formulations and uniform kernel baselines but also leading feature selection methods. In particular, we generate the entire feature selection path for data sets with a hundred thousand features in approximately half an hour on standard hardware. Entire path generation for such data set is well beyond the scaling capabilities of other methods.

**CHAPTER 2**

Learning Proximity Relations for Feature

Selection

**2.1 GENERAL**

IN many pattern recognition and machine learning applications, such as appearance-based image classification, document clustering, data mining, and information retrieval, we are involved to deal with high-dimensionality data which has thousands of features. Learning and classifying in such a high-dimensionality space is extremely difficult due to the curse of dimensionality.In fact, a small fraction among thousands of features is significant and relevant to their classes. The remaining is insignificant which only complicates data learning and modeling. Those insignificant features may seriously degrade the performance of machine learning algorithms. When involved with many insignificant features, even Support Vector Machine (SVM),as one of the most successful classifiers, also works badly in that situation .Thus, those insignificant features are in a way, irrelevant, redundant, and need to be removed. the physical meaning loss, feature selection aims to select a feature subset which can highly preserve the original properties of samples for learning tasks. Beyond alleviating the effect of curse of dimensionality and speeding up the learning processing, there is a distinguishing advantage for feature selection. It’s beneficial to discover the potential association among samples by visualizing the original attributes of points in low-dimension. A typical application for feature selection is Microarray Analysis. Many researchers have explored the microarray technology to build cancer diagnosis, prognosis and prediction from gene expression data. However, the number of gene from microarray data is significantly large, and each gene carries independent genetic instructions for the development of the living organisms. Discovering the underlying associations from gene expressions to cancers needs feature selection techniques not only to reduce efficiently the high-dimensionality gene expression data, but also to preserve the physical integrity of gene for subsequent biological analysis.

**2.2.1 MODULES**

**MODULES:**

1**.** USER INTERFACE DESIGN:

2**.** FILE UPLOADING:

3. STORED IN DATABASE:

4. READ THE FILE AND UPLOADING FILE:

5. DEDUPLICATION:

**MODULE DESCRIPTION**

**1. User Interface Design:**

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. . Logging in is usually used to enter a specific page.

**2. File Uploading**:

In this module the user logged in, then user are going to upload their file

3.**Store In Database**:

In this module, the user uploading all the file where stored in the database. The user retrieve the information from database.

4. **Read The File And Uploading File:**

In this module, he uploading file are stored in database and we are going to read the filename, filesize, filetype etc…, where all the information are stored in database.

**5.Deduplication:**

In this module, the user will uploading the file only once. Incase the user will uploading same file again will became as file already exist.

**MODULE DIAGRAM:**

1. User Interface Design:

Database

Welcome Page

Cloud Appplicatoin

Login

Server

**2.FILE UPLOADING:**

**user login A.txt**

**3. STORE IN THE DATABASE :**

**User login A.txt file Database**

**4. Read The File And Uploading File:**

**database**

**user login A.txt**

**A.txt**

**5.Deduplication:**

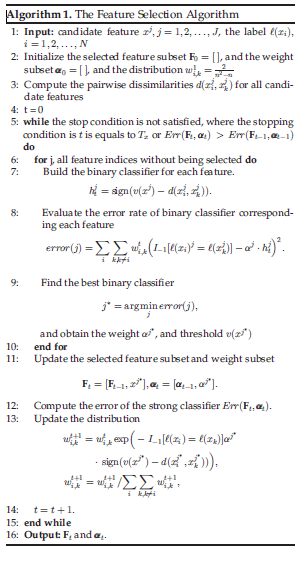
**Database**

**user login A.txt**

**A.txt**

**Duplicate file**

**2.3 SYSTEM TECHNIQUES:**

****

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

These are the requirements for doing the project. Without using these tools and software’s we can’t do the project. So we have two requirements to do the project. They are

1. Hardware Requirements.

2. Software Requirements.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR : PENTIUM IV 2.60 GHz, Intel Dual Core.

RAM : 4 GB DD RAM

MONITOR : 15” LCD, LED MONITOR

HARD DISK : 40 GB

* 1. **SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

Front End : JAVA (j2ee, Servlets, Jsp)

Back End : My SQL

Operating System : Windows, Mac, Linux

IDE : Net Beans, Eclipse

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**Use Case Diagram:**



**EXPLANATION:**

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**Class Diagram**:



**EXPLANATION**:

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code.

**Object Diagram**:

a.txt database

user login file upload

a.txt

**EXPLANATION:**

Object diagram we are telling about the flow of objects how the process is running. In the above digram tells about the flow of objects between the classes.

**State Chart Diagram**:



**EXPLANATION**:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

Activity Diagram:

user

login

File upload

a.txt

a.txt

Cloud

**EXPLANATION:**

In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**Sequence Diagram**:



**EXPLANATION:**

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first Data mining user login into Datamining.

**Collaboration Diagram**:



**EXPLANATION:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.

Data Flow Diagram:

Level 1:

Database

User Login

Level 2:

user

login

database

File uploading

File uploading

user

login

login

user

File uploading

login

login

login

user

user

login

File uploading

File uploading

Duplicate

not access

**EXPLANATION:**

It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel. In the DFDs the level zero process is based on the login validations.

E-R Diagram:

USER

login

File 1

File 2

File 4

File3

**EXPLANATION:**

Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

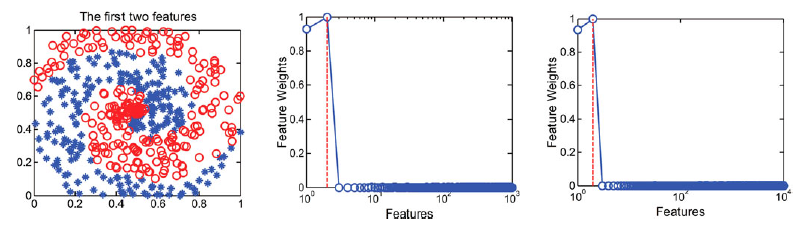
**Component Diagram**:



**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and they are used to illustrate the structure of arbitrarily complex systems.

**System Architecture**:



**4.12 CONCLUSION**:

In this paper, a newly algorithm of feature evaluation is developed to measure the quality of feature, and applied to as a feature selection criterion. A feature subset that gives rise to higher classification ability is considered to be more important. With this criterion, the feature selection task is transformed into an optimization problem. The optimization problem is efficiently solved by following the principle of the AdaBoost-based search method, rather than the exhaustive search. In addition, we also analyze the generalization error bounds of our feature selection algorithm. Various experiments have been conducted on four UCI and 12 microarray data sets to demonstrate the effectiveness of our algorithm, and verify the theoretical results established in this paper.

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 GENERAL**

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA.

**5.2 FEATURES OF JAVA**

**5.2.1 THE JAVA FRAMEWORK**

**Java** is a [programming language](http://en.wikipedia.org/wiki/Programming_language) originally developed by [James Gosling](http://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems)and released in 1995 as a core component of Sun Microsystems' [Java platform](http://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](http://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B) but has a simpler [object model](http://en.wikipedia.org/wiki/Object_model) and fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [bytecode](http://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture). Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applicationsThe java framework is a new platform independent that simplifies application development internet.Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

**5.2.2 OBJECTIVES OF JAVA**

To see places of Java in Action in our daily life, explore java.com.

**Why Software Developers Choose Java**

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

* Write software on one platform and run it on virtually any other platform
* Create programs to run within a Web browser and Web services
* Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
* Combine applications or services using the Java language to create highly customized applications or services
* Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

**Some Ways Software Developers Learn Java**

* Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

**Object Oriented**To be an Object Oriented language, any language must follow at least the four characteristics.

1.Inheritance   :It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse  the existing code and adding addition a features as needed.

2.Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3.Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by thefunctions having the same name based on the signatures of the  methods.

4.Dynamic binding  : Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

**5.2.3**COLLECTIONS:

The Java Collections API's provide Java developers with a set of classes and interfaces that makes it easier to handle collections of objects. In a sense Collection's works a bit like arrays, except their size can change dynamically, and they have more advanced behavior than arrays. In this project we are using Array List for collecting the user input and saving values.

5.2.4 THREAD:

In this project threading concept is very important. A thread is a sequential path of code execution within a program. And each thread has its own local variables, program counter and lifetime. Like creation of a single thread, we can also create more than one thread (multithreads) in a program using class Thread or implementing interface Runnable to make our project efficient and dynamic. In our project we are using request process with the help of multi threading concepts.

5.2.5 SWINGS:

Swing, which is an extension library to the AWT, includes new and improved components that enhance the look and functionality of GUIs. Swing can be used to build Standalone swing gui apps as well as Servlets and Applets. It employs a model/view design architecture. Swing is more portable and more flexible than AWT.

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

In this we implement the coding part using eclipse. Below are the coding’s that are used to generate the domain module for Cloud Computing. Here the proposed techniques are used in the coding part to Cloud to Cloud Interaction.

USER INTERFACE DESIGN**:**

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. . Logging in is usually used to enter a specific page.

**MODULES DIAGRAM:**

* **User Interface Design**

Login

Server

Welcome to Login

Register

New user

Database

Welcome Page

Developer

Application

Registration

Page

Login

Server

Database

Welcome Page

Developer

Application

Registration

Page

Login

Server

Database

Welcome Page

Developer

Application

Registration

Page

Login

Server

Database

Welcome Page

Developer

Application

Registration

Page

Login

Server

Database

Welcome Page

Developer

Application

Registration

Page

Login

Server

**INPUT EXPECTED OUTPUT**

**User Interface Design**:

**Input**:

Developer Login name and Password

**Output**:

If Valid Developer Open the user window otherwise error page.

**Coding**:

package com.Servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.Bean.Bean;

import com.Implementation.Implementation;

import com.Interface.Interface;

/\*\*

\* Servlet implementation class Login

\*/

@WebServlet("/Login")

public class Login extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* Default constructor.

\*/

public Login() {

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session=request.getSession(false);

response.setContentType("text/html");

String username=request.getParameter("username");

String password=request.getParameter("password");

session.setAttribute("username", username);

System.out.println("username="+username);

System.out.println("password="+password);

Bean b=new Bean();

b.setUsername(username);

b.setPassword(password);

Interface i=new Implementation();

boolean m=i.login(b);

System.out.println("m="+m);

if(m==true)

{

response.sendRedirect("fileupload.jsp");

}

else

{

response.sendRedirect("missmatch.jsp");

}

}

}

package com.Servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.Bean.Bean;

import com.Implementation.Implementation;

import com.Interface.Interface;

/\*\*

\* Servlet implementation class Register

\*/

@WebServlet("/Register")

public class Register extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* Default constructor.

\*/

public Register() {

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username=request.getParameter("username");

String email=request.getParameter("email");

String password=request.getParameter("password");

String confirmpassword=request.getParameter("confirmpassword");

String gender=request.getParameter("gender");

String mobile=request.getParameter("mobile");

String address=request.getParameter("address");

System.out.println("username="+username);

System.out.println("email="+email);

System.out.println("password="+password);

System.out.println("confirmpasword="+confirmpassword);

System.out.println("gender="+gender);

System.out.println("mobile="+mobile);

System.out.println("address="+address);

Bean b=new Bean();

b.setUsername(username);

b.setEmail(email);

b.setPassword(password);

b.setConfirmpassword(confirmpassword);

b.setGender(gender);

b.setMobile(mobile);

b.setAddress(address);

Interface i=new Implementation();

int k=i.register(b);

System.out.println("k="+k);

if(k==1)

{

response.sendRedirect("login.jsp");

}

else

{

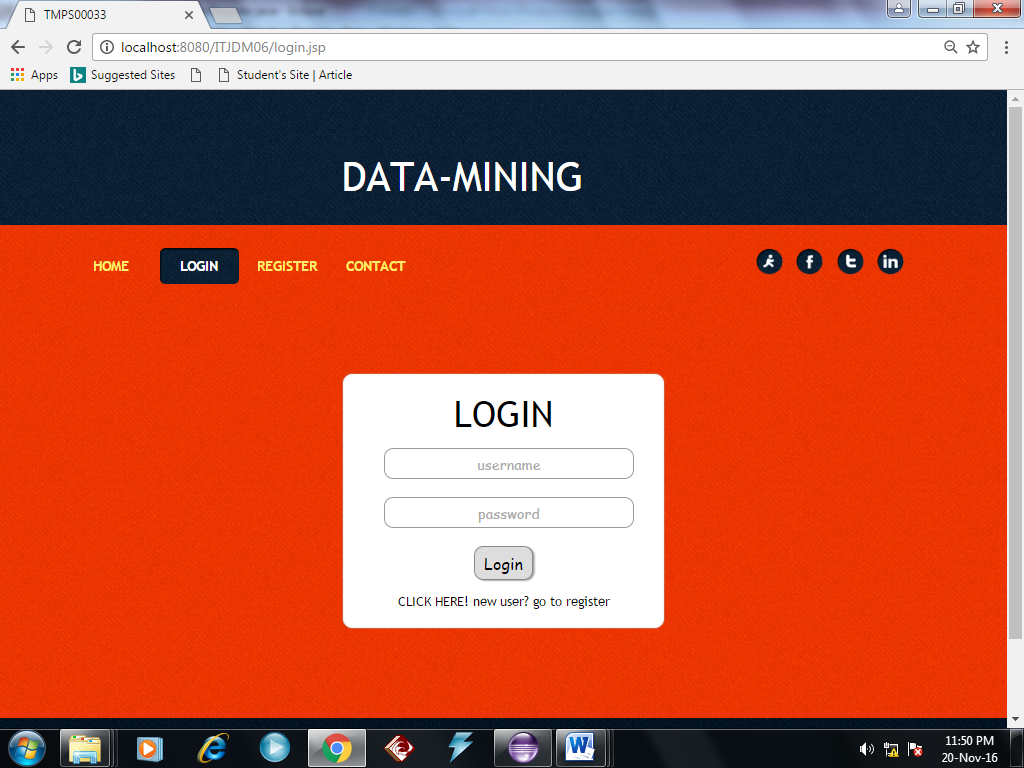
response.sendRedirect("registerfail.jsp");

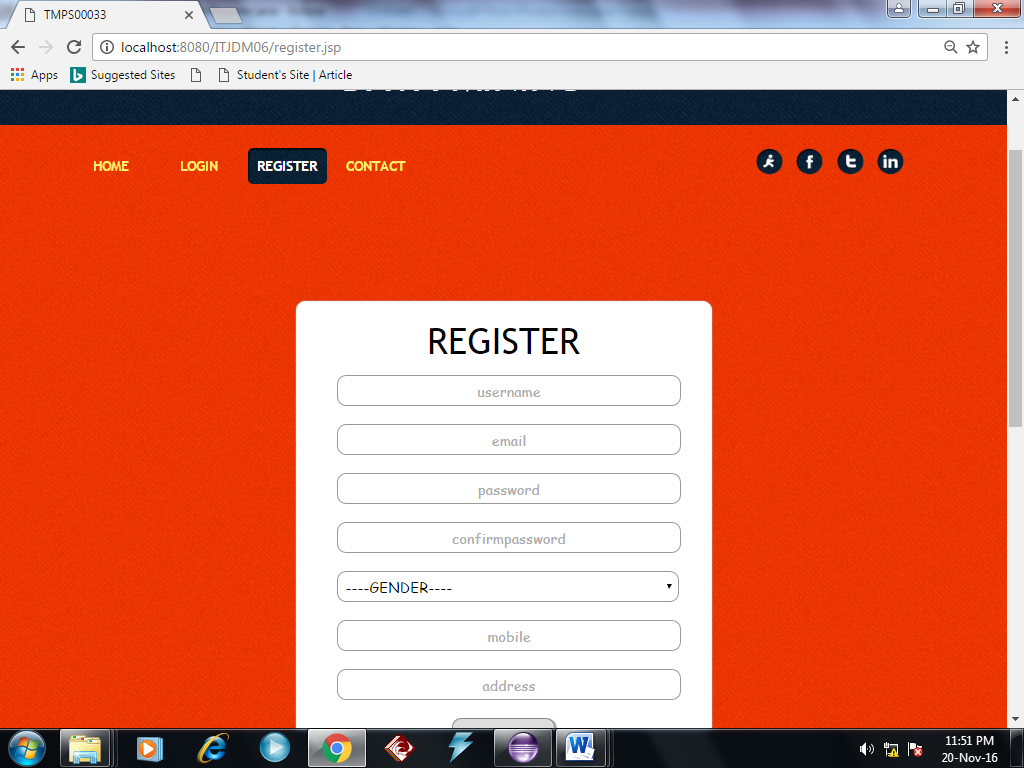
}

}

}

}





**MODULES:**

2. FILE UPLOADING:

**MODULE DESCRIPTION**:

In this module the user logged in, then user are going to upload their file

**MODULE DIAGRAM:**

**2.FILE UPLOADING:**

**user login A.txt**

**Input**:

User uploading a file.

**Output**:

The file will be stored in database.

**Coding**:

package com.Servlet;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.poi.hwpf.HWPFDocument;

import org.apache.poi.hwpf.extractor.WordExtractor;

import com.itextpdf.text.pdf.PdfReader;

import com.itextpdf.text.pdf.parser.PdfTextExtractor;

import com.Bean.Uploadbean;

import com.Implementation.Implementation;

import com.Interface.Interface;

import com.oreilly.servlet.multipart.FilePart;

import com.oreilly.servlet.multipart.MultipartParser;

import com.oreilly.servlet.multipart.Part;

/\*\*

\* Servlet implementation class Fileupload

\*/

@WebServlet("/Fileupload")

public class Fileupload extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Fileupload() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*}

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session=request.getSession(false);

response.setContentType("text/html");

String username=request.getSession().getAttribute("username").toString();

MultipartParser mp=new MultipartParser(request,999999);

Part part;

FilePart filepart;

String filename=null;

long size=0;

String filetype=null;

String filecontent = "";

FileInputStream fis = null;

String path=getServletContext().getRealPath("");

System.out.println(path);

String editpath=path.substring(0,path.indexOf("."));

System.out.println(editpath);

String fullpath=editpath+"ITJDM06\\WebContent\\local\\";

System.out.println(fullpath);

while((part=mp.readNextPart())!=null)

{

if(part.isFile())

{

filepart=(FilePart)part;

filename=filepart.getFileName();

System.out.println(filename);

fullpath=fullpath+filename;

System.out.println(fullpath);

File file=new File(fullpath);

size=filepart.writeTo(file);

System.out.println(size);

filetype=filepart.getContentType();

System.out.println(filetype);

fis = new FileInputStream(file);

if (filename.endsWith(".txt")) {

byte[] b = new byte[fis.available()];

fis.read(b);

String reading = new String(b);

filecontent = filecontent + reading;

System.out.println("txtttttttttttttttttt==="+filecontent);

}

// reading pdf content

else if (filename.endsWith(".pdf")) {

String text = "";

com.itextpdf.text.pdf.PdfReader pdfReader = new PdfReader(fullpath);

int pages = pdfReader.getNumberOfPages();

for (int i1 = 1; i1 < pages; i1++) {

text = PdfTextExtractor.getTextFromPage(pdfReader, i1);

System.out.println("page:" + i1 + " " + text);

filecontent = text;

System.out.println("pdffffffffffff==="+filecontent);

}

// reading doc content

} else if (filename.endsWith(".doc")) {

WordExtractor extractor = null;

FileInputStream fis2 = new FileInputStream(fullpath);

System.out.println("file is" + fis);

HWPFDocument document = new HWPFDocument(fis2);

extractor = new WordExtractor(document);

String mydate = extractor.getTextFromPieces();

System.out.println("THE MYYYYYYYYYYY " + mydate);

System.out.println("doc=============="+filecontent);

}

}

}

Uploadbean up=new Uploadbean();

up.setUsername(username);

up.setFilename(filename);

up.setFiletype(filetype);

up.setFilesize(String.valueOf(size));

up.setFilecontent(filecontent);

Interface i=new Implementation();

int j=i.upload(up,username);

System.out.println("insert>>>>>>============>>>>>>>>>>>>>>>>>>>"+j);

if(j==1)

{

response.sendRedirect("success.jsp");

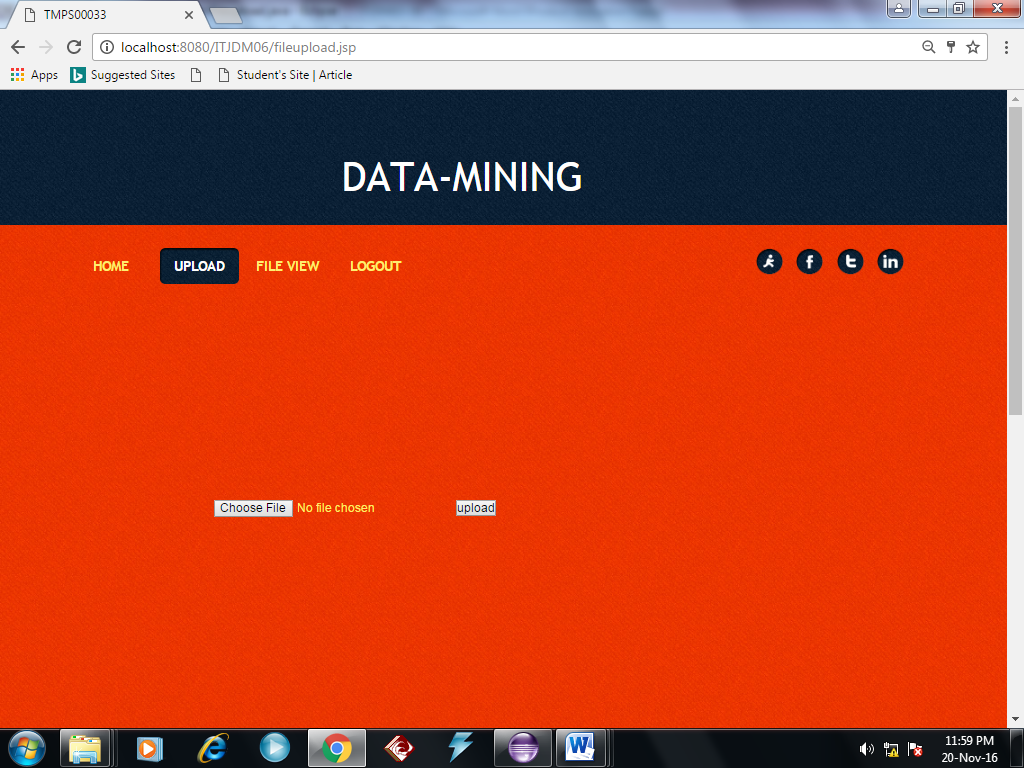
}

else

{

response.sendRedirect("unsuccess.jsp");

}}}



**MODULES**:

3. STORED IN DATABASE:

**MODULE DESCRIPTION:**

3. STORED IN DATABASE:

In this module, the user uploading all the file where stored in the database. The user retrieve the information from database.

**MODULE DIAGRAM:**

**3. STORE IN THE DATABASE :**

**User login A.txt file Database**

**INPUT EXPECTED OUTPUT**

**3. STORE IN THE DATABASE :**

**Input**:

User uploading the file all are stored in the database.

**Output**:

File are stored in the database.

**Coding**:

package com.Servlet;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.poi.hwpf.HWPFDocument;

import org.apache.poi.hwpf.extractor.WordExtractor;

import com.itextpdf.text.pdf.PdfReader;

import com.itextpdf.text.pdf.parser.PdfTextExtractor;

import com.Bean.Uploadbean;

import com.Implementation.Implementation;

import com.Interface.Interface;

import com.oreilly.servlet.multipart.FilePart;

import com.oreilly.servlet.multipart.MultipartParser;

import com.oreilly.servlet.multipart.Part;

/\*\*

\* Servlet implementation class Fileupload

\*/

@WebServlet("/Fileupload")

public class Fileupload extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Fileupload() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*}

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session=request.getSession(false);

response.setContentType("text/html");

String username=request.getSession().getAttribute("username").toString();

MultipartParser mp=new MultipartParser(request,999999);

Part part;

FilePart filepart;

String filename=null;

long size=0;

String filetype=null;

String filecontent = "";

FileInputStream fis = null;

String path=getServletContext().getRealPath("");

System.out.println(path);

String editpath=path.substring(0,path.indexOf("."));

System.out.println(editpath);

String fullpath=editpath+"ITJDM06\\WebContent\\local\\";

System.out.println(fullpath);

while((part=mp.readNextPart())!=null)

{

if(part.isFile())

{

filepart=(FilePart)part;

filename=filepart.getFileName();

System.out.println(filename);

fullpath=fullpath+filename;

System.out.println(fullpath);

File file=new File(fullpath);

size=filepart.writeTo(file);

System.out.println(size);

filetype=filepart.getContentType();

System.out.println(filetype);

fis = new FileInputStream(file);

if (filename.endsWith(".txt")) {

byte[] b = new byte[fis.available()];

fis.read(b);

String reading = new String(b);

filecontent = filecontent + reading;

System.out.println("txtttttttttttttttttt==="+filecontent);

}

// reading pdf content

else if (filename.endsWith(".pdf")) {

String text = "";

com.itextpdf.text.pdf.PdfReader pdfReader = new PdfReader(fullpath);

int pages = pdfReader.getNumberOfPages();

for (int i1 = 1; i1 < pages; i1++) {

text = PdfTextExtractor.getTextFromPage(pdfReader, i1);

System.out.println("page:" + i1 + " " + text);

filecontent = text;

System.out.println("pdffffffffffff==="+filecontent);

}

// reading doc content

} else if (filename.endsWith(".doc")) {

WordExtractor extractor = null;

FileInputStream fis2 = new FileInputStream(fullpath);

System.out.println("file is" + fis);

HWPFDocument document = new HWPFDocument(fis2);

extractor = new WordExtractor(document);

String mydate = extractor.getTextFromPieces();

System.out.println("THE MYYYYYYYYYYY " + mydate);

System.out.println("doc=============="+filecontent);

}

}

}

Uploadbean up=new Uploadbean();

up.setUsername(username);

up.setFilename(filename);

up.setFiletype(filetype);

up.setFilesize(String.valueOf(size));

up.setFilecontent(filecontent);

Interface i=new Implementation();

int j=i.upload(up,username);

System.out.println("insert>>>>>>============>>>>>>>>>>>>>>>>>>>"+j);

if(j==1)

{

response.sendRedirect("success.jsp");

}

else

{

response.sendRedirect("unsuccess.jsp");

}

}

}

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"com.itextpdf.text.log.SysoCounter"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<%@ page import=*"java.util.\*"* %>

<%@ page import=*"com.Dbconnection.\*"* %>

<%@page import=*"java.util.\*"* %>

<%@ page import=*"java.sql.Connection.\*"* %>

<%@ page import=*"java.sql.DriverManager.\*"* %>

<%@ page import=*"java.util.\*"* %>

<%@ page import=*"java.sql.\*"* %>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"* />

<title>TMPS00033</title>

<link href=*"css/styles.css"* rel=*"stylesheet"* type=*"text/css"* />

</head>

<body>

<div class=*"header-wrap"*>

<div class=*"logo"*>

<h1>DATA-MINING</h1>

</div>

</div><!---header-wrap-End--->

<div class=*"menu-wrap"*>

<div class=*"menu"*>

<ul>

<li><a href=*"home.jsp"* >home</a></li>

<li><a href=*"fileupload.jsp"* >upload</a></li>

<li><a href=*"fileview.jsp"* class=*"active"*>file view</a></li>

<li><a href=*"home.jsp"*>LOGOUT</a></li>

</ul>

</div>

<div class=*"socia-wrap"*>

<div class=*"socail"*>

<ul>

<li><a href=*"#"*><img src=*"images/aim.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/facebook.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/twwtter.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/linkedin.png"* alt=*""* /></a></li>

</ul>

</div>

</div>

</div><!---menu-wrap-End--->

<div class=*"page-wrap"*>

<div class=*"wrap3"*>

<div class=*"leftcol"*>

<div class=*"panel"*>

<% String username=request.getSession().getAttribute("username").toString();

System.out.println("username"+username);

%>

<%

PreparedStatement ps ;

ResultSet rs ;

String query = "SELECT \* FROM `itjdm06`.`upload` where username='"+username+"'";

ps=Dbconnection.createConnection().prepareStatement(query);

rs = ps.executeQuery();

%>

<form style="height:*500px*; margin-left: *30px*;">

<table style="width: *100%*" border=*"1"* cellspacing=*"5"*

cellpadding=*"5"*>

<tr>

<th>USERNAME</th>

<th>FILE NAME</th>

<th>FILE TYPE</th>

<th>FILE SIZE</th>

</tr>

<%

**while** (rs.next()) {

%>

<tr>

<td><%=rs.getString(1) %></td>

<td><%=rs.getString(2) %></td>

<td><%=rs.getString(3) %></td>

<td><%=rs.getString(4) %></td>

</tr>

<%

}

%>

</table>

</form>

</div>

<!-- <form action="Fileupload" method="post" enctype="multipart/form-data" style="height:600px;">

<center>

<input type="file" name="file" style="margin-top: 180px;">

</input>

<input type="submit" value="upload" />

</center>

</form> -->

<div id=*"message"*></div>

</div>

</div>

</div>

<!-- <div class="rightcol">

<div class="panel">

<div class="title"><h1>Lorem ipsum <br /> Nulla varius libero</h1></div>

<div class="content">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Fusce commodo </spna>

</div>

<div class="content marTop40">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Donec lacinia </spna>

</div>

<div class="content marTop40">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Vestibulum aliquam</spna>

<p>Vestibulum fringilla nisl sit amet metus tristique</p>

<div class="button marTop30"><a href="#">More Info</a></div>

</div>

</div>

<div class="panel marTop40">

<div class="title"><h1>Nulla facilisi<br />Suspendisse aliquet</h1></div>

<div class="content">

<p>Mauris ornare eros in purus cursus molliam nec gravida magna.</p>

<ul>

<li><a href="#">- Lorem ipsum dolor sit amet, consect</a></li>

<li><a href="#">- Duis varius lectus at nunc dignissim tin</a></li>

<li><a href="#">- Morbi ac mauris lacus, ac mol</a></li>

<li><a href="#">- Etiam sollicitudin nisi a leo porta</a></li>

<li class="bg-bottom"><a href="#">- Pellen sepien gravids loreet ornare</a></li>

</ul>

<div class="button marTop30"><a href="#">More Info</a></div>

</div>

</div>

</div> -->

<div class=*"clearing"*></div>

</div>

</div><!---page-wrap-End--->

<div class=*"footer-wrap"*>

<div class=*"wrap"*>

</div>

<!---footer-wrap--->

<div class=*"clearing"*></div>

<div class=*"copyright-wrap"*>

<div class=*"wrap"*>

<div class=*"content"*>

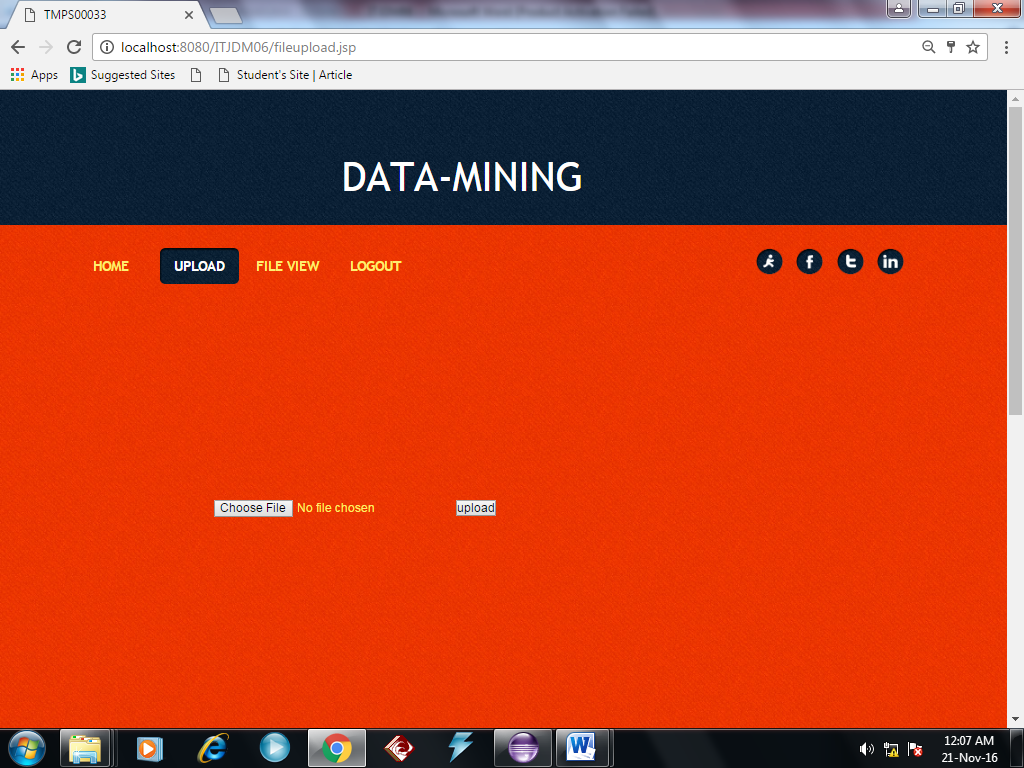
</div>

</div>

</div><!---copyright-wrap-End--->

</body>

</html>



**MODULES**:

4. READ THE FILE AND UPLOADING FILE:

**MODULES DESCRIPTION:**

**4. Read The File And Uploading File**:

In this module, he uploading file are stored in database and we are going to read the filename, filesize, filetype etc…, where all the information are stored in database.

**MODULES DIAGRAM:**

**4. Read The File And Uploading File:**

**database**

**user login A.txt**

**A.txt**

**INPUT EXPECTED OUTPUT**

**4. Read The File And Uploading File:**

**Input**:

User uploading the file all are stored in the database.

**Output**:

File are stored in the database and read.

**Coding**:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"com.itextpdf.text.log.SysoCounter"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<%@ page import=*"java.util.\*"* %>

<%@ page import=*"com.Dbconnection.\*"* %>

<%@page import=*"java.util.\*"* %>

<%@ page import=*"java.sql.Connection.\*"* %>

<%@ page import=*"java.sql.DriverManager.\*"* %>

<%@ page import=*"java.util.\*"* %>

<%@ page import=*"java.sql.\*"* %>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"* />

<title>TMPS00033</title>

<link href=*"css/styles.css"* rel=*"stylesheet"* type=*"text/css"* />

</head>

<body>

<div class=*"header-wrap"*>

<div class=*"logo"*>

<h1>DATA-MINING</h1>

</div>

</div><!---header-wrap-End--->

<div class=*"menu-wrap"*>

<div class=*"menu"*>

<ul>

<li><a href=*"home.jsp"* >home</a></li>

<li><a href=*"fileupload.jsp"* >upload</a></li>

<li><a href=*"fileview.jsp"* class=*"active"*>file view</a></li>

<li><a href=*"home.jsp"*>LOGOUT</a></li>

</ul>

</div>

<div class=*"socia-wrap"*>

<div class=*"socail"*>

<ul>

<li><a href=*"#"*><img src=*"images/aim.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/facebook.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/twwtter.png"* alt=*""* /></a></li>

<li><a href=*"#"*><img src=*"images/linkedin.png"* alt=*""* /></a></li>

</ul>

</div>

</div>

</div><!---menu-wrap-End--->

<div class=*"page-wrap"*>

<div class=*"wrap3"*>

<div class=*"leftcol"*>

<div class=*"panel"*>

<% String username=request.getSession().getAttribute("username").toString();

System.out.println("username"+username);

%>

<%

PreparedStatement ps ;

ResultSet rs ;

String query = "SELECT \* FROM `itjdm06`.`upload` where username='"+username+"'";

ps=Dbconnection.createConnection().prepareStatement(query);

rs = ps.executeQuery();

%>

<form style="height:*500px*; margin-left: *30px*;">

<table style="width: *100%*" border=*"1"* cellspacing=*"5"*

cellpadding=*"5"*>

<tr>

<th>USERNAME</th>

<th>FILE NAME</th>

<th>FILE TYPE</th>

<th>FILE SIZE</th>

</tr>

<%

**while** (rs.next()) {

%>

<tr>

<td><%=rs.getString(1) %></td>

<td><%=rs.getString(2) %></td>

<td><%=rs.getString(3) %></td>

<td><%=rs.getString(4) %></td>

</tr>

<%

}

%>

</table>

</form>

</div>

<!-- <form action="Fileupload" method="post" enctype="multipart/form-data" style="height:600px;">

<center>

<input type="file" name="file" style="margin-top: 180px;">

</input>

<input type="submit" value="upload" />

</center>

</form> -->

<div id=*"message"*></div>

</div>

</div>

</div>

<!-- <div class="rightcol">

<div class="panel">

<div class="title"><h1>Lorem ipsum <br /> Nulla varius libero</h1></div>

<div class="content">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Fusce commodo </spna>

</div>

<div class="content marTop40">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Donec lacinia </spna>

</div>

<div class="content marTop40">

<div class="icon"><img src="images/icon5.png" alt="icon" /></div>

<spna>Vestibulum aliquam</spna>

<p>Vestibulum fringilla nisl sit amet metus tristique</p>

<div class="button marTop30"><a href="#">More Info</a></div>

</div>

</div>

<div class="panel marTop40">

<div class="title"><h1>Nulla facilisi<br />Suspendisse aliquet</h1></div>

<div class="content">

<p>Mauris ornare eros in purus cursus molliam nec gravida magna.</p>

<ul>

<li><a href="#">- Lorem ipsum dolor sit amet, consect</a></li>

<li><a href="#">- Duis varius lectus at nunc dignissim tin</a></li>

<li><a href="#">- Morbi ac mauris lacus, ac mol</a></li>

<li><a href="#">- Etiam sollicitudin nisi a leo porta</a></li>

<li class="bg-bottom"><a href="#">- Pellen sepien gravids loreet ornare</a></li>

</ul>

<div class="button marTop30"><a href="#">More Info</a></div>

</div>

</div>

</div> -->

<div class=*"clearing"*></div>

</div>

</div><!---page-wrap-End--->

<div class=*"footer-wrap"*>

<div class=*"wrap"*>

</div>

<!---footer-wrap--->

<div class=*"clearing"*></div>

<div class=*"copyright-wrap"*>

<div class=*"wrap"*>

<div class=*"content"*>

</div>

</div>

</div><!---copyright-wrap-End--->

</body>

</html>



**MODULES**:

**5.DEDUPLICATION**:

**MODULES DESCRIPTION**:

**5.DEDUPLICATION**:

In this module, the user will uploading the file only once. In case the user will uploading same file again will became as file already exist.

**MODULE DIAGRAM:**

**5.Deduplication:**

**Database**

**user login A.txt**

**A.txt**

**Duplicate file**

**INPUT EXPECTED OUTPUT**

**5.Deduplication:**

**Input**:

The file all are stored in the database and duplicate the file.

**Output**:

File are stored in the database and finding the duplicate file.

**Coding**:

package com.Servlet;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.poi.hwpf.HWPFDocument;

import org.apache.poi.hwpf.extractor.WordExtractor;

import com.itextpdf.text.pdf.PdfReader;

import com.itextpdf.text.pdf.parser.PdfTextExtractor;

import com.Bean.Uploadbean;

import com.Implementation.Implementation;

import com.Interface.Interface;

import com.oreilly.servlet.multipart.FilePart;

import com.oreilly.servlet.multipart.MultipartParser;

import com.oreilly.servlet.multipart.Part;

/\*\*

\* Servlet implementation class Fileupload

\*/

@WebServlet("/Fileupload")

public class Fileupload extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Fileupload() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*}

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session=request.getSession(false);

response.setContentType("text/html");

String username=request.getSession().getAttribute("username").toString();

MultipartParser mp=new MultipartParser(request,999999);

Part part;

FilePart filepart;

String filename=null;

long size=0;

String filetype=null;

String filecontent = "";

FileInputStream fis = null;

String path=getServletContext().getRealPath("");

System.out.println(path);

String editpath=path.substring(0,path.indexOf("."));

System.out.println(editpath);

String fullpath=editpath+"ITJDM06\\WebContent\\local\\";

System.out.println(fullpath);

while((part=mp.readNextPart())!=null)

{

if(part.isFile())

{

filepart=(FilePart)part;

filename=filepart.getFileName();

System.out.println(filename);

fullpath=fullpath+filename;

System.out.println(fullpath);

File file=new File(fullpath);

size=filepart.writeTo(file);

System.out.println(size);

filetype=filepart.getContentType();

System.out.println(filetype);

fis = new FileInputStream(file);

if (filename.endsWith(".txt")) {

byte[] b = new byte[fis.available()];

fis.read(b);

String reading = new String(b);

filecontent = filecontent + reading;

System.out.println("txtttttttttttttttttt==="+filecontent);

}

// reading pdf content

else if (filename.endsWith(".pdf")) {

String text = "";

com.itextpdf.text.pdf.PdfReader pdfReader = new PdfReader(fullpath);

int pages = pdfReader.getNumberOfPages();

for (int i1 = 1; i1 < pages; i1++) {

text = PdfTextExtractor.getTextFromPage(pdfReader, i1);

System.out.println("page:" + i1 + " " + text);

filecontent = text;

System.out.println("pdffffffffffff==="+filecontent);

}

// reading doc content

} else if (filename.endsWith(".doc")) {

WordExtractor extractor = null;

FileInputStream fis2 = new FileInputStream(fullpath);

System.out.println("file is" + fis);

HWPFDocument document = new HWPFDocument(fis2);

extractor = new WordExtractor(document);

String mydate = extractor.getTextFromPieces();

System.out.println("THE MYYYYYYYYYYY " + mydate);

System.out.println("doc=============="+filecontent);

}

}

}

Uploadbean up=new Uploadbean();

up.setUsername(username);

up.setFilename(filename);

up.setFiletype(filetype);

up.setFilesize(String.valueOf(size));

up.setFilecontent(filecontent);

Interface i=new Implementation();

int j=i.upload(up,username);

System.out.println("insert>>>>>>============>>>>>>>>>>>>>>>>>>>"+j);

if(j==1)

{

response.sendRedirect("success.jsp");

}

else

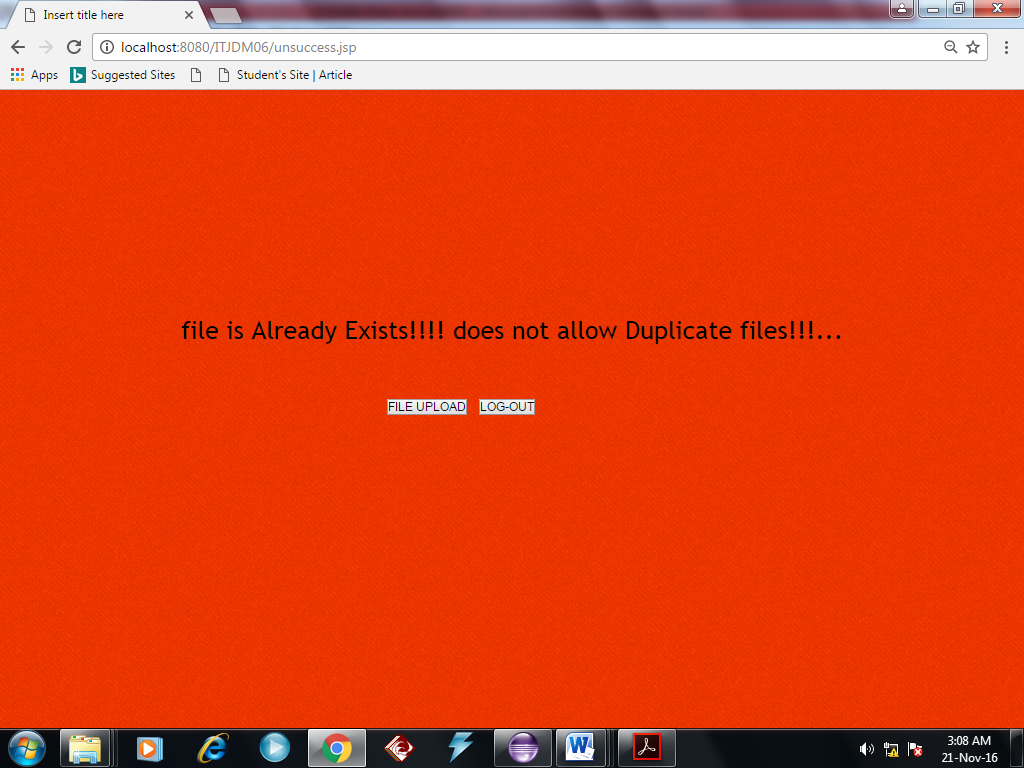
{

response.sendRedirect("unsuccess.jsp");

}

}}





**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

In this paper, we develop achievability protocols and outer bounds for the secure network coding setting, where the edges are subject to packet erasures, and public feedback of the channel state is available to both Eve and the legitimate network nodes. Secure network coding assumes that the underlying network channels are error-free; thus, if our channels introduce errors, we need to first apply a channel code to correct them, and then build security on top of the resulting error-free network. We show that by leveraging erasures and feedback, we can achieve secrecy rates that are in some cases multiple times higher than the alternative of separate channel-error-correction followed by secure network coding; moreover, we develop outer bounds and prove optimality of our proposed schemes in some special cases.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3Types of Tests**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**APPLICATION**

**9.1 GENERAL**

Cloud computing is defined as a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. Cloud storage services have rapidly become increasingly popular. Users can store their data on the cloud and access their data anywhere at any time. It is noted that data owners lose ultimate control over the fate of their outsourced data; thus, the correctness, availability and integrity of the data are being put at risk. On the one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt users’ data.

**9.2 APPLICATION**

**Semantic Web applications:**

LDO is the cornerstone of The Semantic Web, yet there still very few commercial LDO apps. In the latest issue of Nodalities, a magazine about the Semantic Web by UK Company Talis, there is an article by Talis CTO Ian Davis about the state of Semantic Web applications.

**LDO application development for IBM data servers**

An LDOstore in the DB2 database server is a set of user tables within a database schema that stores an LDOdata set. A unique store name is associated with each set of these tables. Each LDOstore has a table that contains metadata for the store. This table has the same name as the store.

**9. FUTURE ENHANCEMENT**

Algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

**Conclusion:**

In this paper, a newly algorithm of feature evaluation is developed to measure the quality of feature, and applied to as a feature selection criterion. A feature subset that gives rise to higher classification ability is considered to be more important. With this criterion, the feature selection task is transformed into an optimization problem. The optimization problem is efficiently solved by following the principle of the AdaBoost-based search method, rather than the exhaustive search. In addition, we also analyze the generalization error bounds of our feature selection algorithm. Various experiments have been conducted on four UCI and 12 microarray data sets to demonstrate the effectiveness of our algorithm, and verify the theoretical results established in this paper.

**REFERENCES:**

[1] A. A. Alizadeh, et al., “Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling,” Nature, vol. 286, no. 5439, pp. 531–537, 1999.

[2] A. Arauzo-Azofra, J. Aznarte, and J. Benıtez, “Empirical study of feature selection methods based on individual feature evaluation for classification problems,” Expert Syst. Appl., vol. 38, no. 7, pp. 8170–8177, 2011.

[3] A. J. Stephenson, A. Smith, M. W. Kattan, J. Satagopan, V. E. Reuter, P. T. Scardino, and W. L. Gerald, “Integration of gene expression profiling and clinical variables to predict prostate carcinoma recurrence after radical prostatectomy,” Cancer, vol. 104, no. 2, pp. 290–298, 2005.

[4] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 4–37, Jan. 2000.

[5] A. Tayal, T. Coleman, and Y. Li, “Primal explicit max margin feature selection for nonlinear support vector machines,” Pattern Recognit., vol. 47, no. 6, pp. 2153–2164, 2014.

[6] H. Almuallim and T. Dietterich, “Learning boolean concepts in the presence of many irrelevant features,” Artif. Intell., vol. 69, pp. 279–305, 1994.

[7] C. Ambroise and G. J McLachlan, “Selection bias in gene extraction on the basis of microarray gene-expression data,” Proc. Nat. Acad. Sci. USA, vol. 99, no. 10, pp. 6562–6566, 2002.

[8] C. Bishop, Neural Networks for Pattern Recognition. Oxford, U.K.: Oxford Univ. Press, 1995.

[9] C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[10]V.Bolon-Canedo, N. Sanchez-Maro~no, and A. Alonso-Betanzos, “A review of feature selection methods on synthetic data,” Knowl. Inf. Syst., vol. 34, no. 3, pp. 483–519, 2013.

