DECLARATION

We, Samir Sheriff and Satvik N bearing USN number 1RV09CS093 and 1RV09CS095 respectively, hereby declare that the dissertation entitled "**Decision Tree Classifier with GA based feature selection**" completed and written by us, has not been previously formed the basis for the award of any degree or diploma or certificate of any other University.

Bangalore

Samir Sheriff
USN:1RV09CS093

Satvik N
USN:1RV09CS095

R V COLLEGE OF ENGINEERING

(Autonomous Institute Affiliated to VTU)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled, "**Decision Tree Classifier with GA based feature selection**", which is being submitted herewith for the award of B.E is the result of the work completed by **Samir Sheriff and Satvik N** under my supervision and guidance.

Signature of Guide   Signature of Head of Department
(Mrs. Shanta R)   (Dr. N K Srinath)

Name of Examiner   Signature of Examiner

1:

2:

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. We would like to take this opportunity to thank them all.

First and foremost we would like to thank Dr. B. S. Satyanarayana, Principal, R.V.C.E, Bengaluru, for his moral support towards completing my project work.

We deeply express my sincere gratitude to our guides Ms. Shanta Rangaswamy, Assistant Professor and Dr. Shobha G, Professor, Department of CSE, R.V.C.E, Bengaluru, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank Dr. N. K. Srinath, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

We thank our Parents, and all the Faculty members of Department of Computer Science & Engineering for their constant support and encouragement.

Last, but not the least, We would like to thank our peers and friends who provided me with valuable suggestions to improve our project.

Samir Sheriff
8th semester, CSE
USN:1RV09CS093
Satvik N
8th semester, CSE
USN:1RV09CS095
Vaishakh B N
8th semester, CSE
USN:1RV09CS114

ABSTRACT

A time series is a sequence of data points, measured typically at successive points in time spaced at uniform time intervals. Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data.In the context of statistics,the primary goal of time series analysis is forecasting. In the context of signal processing it is used for signal detection and estimation, while in the context of data mining, pattern recognition and machine learning time series analysis can be used for clustering, classification, query by content, anomaly detection as well as forecasting. This project is aimed making a time series data mining tool which can be used to accomplish the above goals.

@TODO Data sets names to be added

The project makes use of quite a few data sets in for time series analysis. these include Rainfall Data from Sita Nadi, electricity consumption data from XYZ city, ECG data and Sea Level Data. Anomaly detection, forecasting, Similarity detection is performed on these data sets using the algorithms available in literature.

@TODO Key findings and results come here

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

A time series is a set of observations Xt , each one being recorded at a specific time t. Discrete-time time series is one in which the set T of times at which observations are made is a discrete set. Continuous-time time series are obtained when observations are recorded continuously over some time interval, e.g., when T0 belongs [0,1]. Examples of time series are the daily closing value of the ECG readings and the annual flow volume of the Nile River at Aswan. Time series are very frequently plotted via line charts. Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

1.1 DEFINITIONS AND USAGE

Definition of Time Series: An ordered sequence of values of a variable at equally spaced time intervals. TSDM : Time Series Data Mining tool. @TODO ADD DETAILS OF THIS SECTION

1.2 LITERATURE SURVEY

A time series is a collection of observations made sequentially through time. At each time point one or more measurements may be monitored corresponding to one or more attributes under consideration. The resulting time series is called univariate or multivariate respectively. In many cases the term sequence is used in order to refer to a time series, although some authors refer to this term only when the corresponding values are non-numerical. Throughout this paper the terms sequence and time series are being used interchangeably. The most common tasks of time series data mining methods are: indexing, clustering, classification, novelty detection, motif discovery and rule discovery. In most of the cases, forecasting is based on the outcomes of the other tasks. A brief description of each task is given below.

Indexing: Find the most similar time series in a database to a given query time series.

Clustering: Find groups of time series in a database such that, time series of the same group are similar to each other whereas time series from different groups are dissimilar to each other.

Classification: Assign a given time series to a predefined group in a way that is more similar to other time series of the same group than it is to time series from other groups.

Novelty detection: Find all sections of a time series that contain a different behavior than the expected with respect to some base model.

Motif discovery: Detect previously unknown repeated patterns in a time series database.

Rule discovery: Infer rules from one or more time series describing the most possible behaviour that they might present at a specific time point (or interval).

The temporal aspect of data arises some special issues to be considered and/or imposes some restrictions in the corresponding applications. First, it is necessary to define a similarity measure between two time series and this issue is very important in TSDM since it involves a degree of subjectivity that might affect the final result. A lot of research has focused on defining different similarity measures in order to improve the performance of the corresponding methods. Second, it is necessary to apply a representation scheme on the time series data. Since the amount of data may range from a few megabytes to terabytes, an appropriate representation of the time series is necessary in order to manipulate and analyze it efficiently. The desirable properties that this approach should hold are: (a) the completeness of feature extraction (b) the reduction of the dimensionality curse [1]. More specifically, the method of extraction features should guarantee that there would be no pattern missed, the number of patterns falsely identified as interesting will be minimized and the dimensionality reduction will be substantial. In many cases also, the objective is to take advantage of the specific characteristics of a representation that make specific methods applicable (i.e. inducing rules, Markov models). Consequently, the majority of the researchers are focused on defining novel similarity measures and representation schemes in order to improve indexing performance. Clustering and classification of time series rely heavily on the similarity measure and the representation scheme selected, thus, there are very few papers proposing a novel algorithm [2]. A recent survey on clustering time series is provided by Liao [3]. Novelty detection is a very important task in many areas. Several alternative terms for novelty have been used, such as, anomaly, interestingness, surprising, faults to name a few. Moreover, many problems of finding periodic patterns can be considered as similar problems. The important point here is to provide a clear and concise definition of the corresponding notion. For instance, Keogh et al. [14] describe a pattern as surprising if the frequency with which it appears, differs greatly from that expected given previous experience. The authors present a novel algorithm, called Tarzan, and provide useful pointers to relevant literature. Recently, Aref et al. [4] focus on discovering partial periodic patterns in one or more databases. They present algorithms for incremental mining (how to maintain discovered patterns over time as the database is being expanded). Motif discovery has only recently attracted the interest of the data mining community [9]. Motifs are defined to be previously unknown, frequently occurring patterns in a time series. These patterns may be of particular importance to other data mining tasks, such as, rule discovery and novelty detection. The recent work of Tanaka et al. [7] proposes a new method for identifying motifs from multi-dimensional time series. They apply Principal Component Analysis to reduce dimensionality and perform a symbolic representation. Then, the motif discovery procedure starts by calculating a description length of a pattern based on the Minimum Description Length principle.

- Indexing Indexing approaches are mostly influenced by the pioneer work of Agrawal et al. [1], generalized by Faloutsos et al. [12]. The emerged framework from these papers, referred as GEMINI, can be summarized in the following steps [11]: extract k essential features from the time series map into a point in k-dimension feature space organize points with off-the-shelf spatial access method. discard false alarms The first and second step suggests the application of a representation scheme in order to reduce the dimensionality. However, this mapping should guarantee that it would return all the qualifying objects. This implies that the similarity measure in the k-dimension feature space should lower bound the corresponding similarity measure in the original space [8]. The third step is an opened selection, however most of the times R-tree structures are used. Other indexing structures may be vp-trees [7] [9], hB-trees and grid-files. The fourth step is a consequence of the fact that this approach can not guarantee that there will not be returned unqualified objects, thus these false alarms should be discarded in a post processing phase. Recently, Vlachos et al. [8] presented an external memory indexing method for discovering similar multidimensional time series under time warping conditions. The main contribution of this work is the ability to support various distance measures without the need to reconstruct the index. Two approaches with respect to distance measures are taken under consideration, namely, the Longest Common Subsequence (LCS) and the Dynamic Time Warping (DTW). Their indexing technique works by splitting a set of multiple time series in multidimensional Minimum Bounding Rectangles (MBR) and storing them in an R-tree. For a given query, a Minimum Bounding Envelope (MBE) is constructed, that covers all the possible matching areas of the query under time warping conditions. This MBE is decomposed into MBRs and then probed in the R-tree index.

- Time series representation There have been several time series representations proposed in the literature, mainly on the purpose of reducing the intrinsically high dimensionality of time series. We will refer to some of the most commonly used representations. Discrete Fourier Transform (DFT) [1] was one of the first representation schemes proposed within data mining context. DFT transforms a time series from the time domain into the frequency domain whereas a similar representation scheme, Discrete Wavelet Transform (DWT) [8], transforms it into the time/frequency or space/frequency domain. Singular Value Decomposition (SVD)

[5] performs a global transformation by rotating the axes of the entire dataset such that the first axis explains the maximum variance, the second axis explains the maximum of the remaining variance and is orthogonal to the first axis etc. Piecewise Aggregate Approximation (PAA) [3] divides a time series into segments of equal length and records the mean of the corresponding values of each one. Adaptive Piecewise Constant Approximation (APCA) [10] is similar to PAA but allows segments of different lengths. Piecewise Linear Approximation (PLA) approximates a time series by a sequence of straight lines. Recently, more representation schemes have been proposed in order to reduce dimensionality. The first class of these schemes consists of symbolic representations. Lin et al. [11] propose a Symbolic Aggregate Approximation (SAX) method, which uses as a first step the PAA representation and then discretizes the transformed time series by using the properties of the normal probability distribution. Bagnal[5] assess the effects of clipping original data on the clustering of time series. Each point of a series is mapped to 1 when it is above the population mean and to 0 when it is below. This representation is called clipping and has many advantages especially when the original series is long enough. It achieves adequate accuracy in clustering, it efficiently handles outliers and it provides the ability to employ algorithms developed for discrete or categorical data. Megalooikonomou et al. [30] introduce a novel dimensionality reduction technique, called Piecewise Vector Quantized Approximation (PVQA). This technique is based on vector quantization that partitions each series into segments of equal length and uses vector quantization to represent each segment by the closest codeword from a codebook. The original time series is transformed to a lower dimensionality series of symbols. This approach requires a training phase in order to construct the codebook, a data-encoding scheme and a distance measure. Cole et al. [9] provide a work that addresses the task of discovering correlated windows of time series (synchronously or with lags) over streaming data. They concentrate in the case where the time series are uncooperative, meaning that there does not exist a fundamental degree of regularity that would allow an efficient implementation of DFT transformations. The proposed method involves a combination of

several techniques sketches (random projections), convolution, structured random vectors, grid structures, and combinatorial design in order to achieve high performance. Gionis and Mannila [7] introduce a different approach, which is mainly motivated from research on human genome sequences. However, this approach is more general and involves multivariate time series. The notion behind their approach is that, the high variability that some time series very often exhibit, may be explained by the existence of several different sources that affect different segments of this series. More specifically, the task is to find a proper way to segment a time series into k segments, each of which comes from one of h different sources (k ¿¿h). This task is analogous to clustering the points of a time series in h clusters with the additional constraint that a cluster may change at most k-1 times. Gionis and Mannila provide three algorithms for solving this problem and they test them on synthetic and genome data. Finally, Vlachos et al. [3] propose to represent a time series by applying discrete Fourier transformations and retain the k best Fourier coefficients instead of the first few ones. Although this paper is motivated by mining knowledge from the query logs of the MSN search engine, the proposed methods may be applied for time series data mining in general.

- Similarity Measures The definition of novel similarity measures has been one of the most researched areas in the TSDM field. Generally, they are strongly related to the representation scheme applied to the original data. However, there are some similarity measures that appear frequently in the literature. Most of the researchers choices are based on the family of Lp norms, that include the Euclidean distance. Yi and Faloutsos [3] presented a novel and fast indexing scheme when the distance function is any of the arbitrary Lp norms (p = 1, 2, ,). Another similarity measure that attracted a lot of attention, Dynamic Time Warping (DTW), comes from the speech recognition field [6]. The main advantage of this measure is that it allows acceleration-deceleration of a series along the time dimension (nonlinear alignments are possible), however it is computationally expensive. Markov models have been constructed and experimented. Another family of distance measures, Longest Common Subsequence Measures (LCS), often used in speech recognition and text pattern matching. As an example of this approach, we refer to the work of Agrawal et al. [2] who define two sequences as similar when they have enough, non-overlapping, time-ordered pairs of subsequences that are similar. Li et al. [6] propose an algorithm for fast and efficient recognition of motions in multi-attribute continuous motion sequences. The main contribution of this paper is the definition of a similarity measure based on the analysis of Singular Value Decomposition (SVD) properties of similar multi-attribute motions. The proposed measure deals with noise and takes into account the different rates and durations of each motion. The authors also propose a five-phase algorithm for handling segmentation and recognition in real-time. Sakurai et al. [5] propose the Fast search method for dynamic Time Warping (DTW) that satisfies the following criteria: (a) it is fast (b) it produces no false dismissals (c) it does not pose any restriction on the series length (d) it supports for any, as well as for no restriction on warping scope. Their approach is based on a new lower bounding distance measure. They represent the sequence with approximate segments, not necessary of equal length, and operate on them. Three segments, the lower bound, the upper bound, and the time interval, correspond to each one of these approximate segments. In order to fulfill all of the above criteria, they provide algorithms for dynamic programming and searching adjusted to the properties of this representation. Fu et al. [14] propose a new technique to query time series that incorporates global scaling and time warping. The argument is that most real world problems require the ability to handle both types of distortion simultaneously. The approach is to scale the sequence by a bounded scaling factor and also to find nearest neighbor or evaluate range query by applying time warping. The authors provide definitions and proofs of the necessary lower bounds. Furthermore, there is the expected contribution to defining similarity measures by papers that propose novel representation schemes, since these two tasks are interrelated to each other.

1.3 MOTIVATION

mr. Manju. @TODO Add video link here..

1.4 PROBLEM STATEMENT

Make a paper and publish water data everywhere. :/ @TODO add problem statement.

1.5 OBJECTIVE

The ability to model and perform decision modelling and analysis is an essential feature of many real-world applications ranging from emergency medical treatment in intensive care units to military command and control systems. Existing formalisms and methods of inference have not been effective in real-time applications where trade-offs between decision quality and computational tractability are essential. The objective of this project is to fill the void that exists and help in proper analysis of time varying data.

1.6 SCOPE

The scope of a time series data mining tool is two fold. The first is to obtain an understanding of the underlying forces and structure that produced the observed data. The second is to fit a model and proceed to forecasting, monitoring or even feedback and feed forward control. The time series data mining tool can be used in the following fields.

- **Economic Forecasting**
- **Sales Forecasting**
- **Rainfall Analysis**
- **Stock Market Analysis**
- **Yield Projections**

- **Process and Quality Control**
- **Census Analysis**

1.7 METHODOLOGY

Time series analysis of data requires the user to able to view the different algorithms and the result obtained from each algorithm along with the graphs which help the user understand the time varying nature of the data. Hence, the representation of data becomes very important. Having understood this requirement in the early phase of the project, we adopted a methodology that will accomplish the objectives in a neat and intuitive way. A GUI was developed in the form of Java Server Pages and the back end was coded in Java which helped us exploit the object oriented paradigm in design of algorithms.

1.8 ORGANIZATION OF REPORT

@TODO after all chapters are complete.

Chapter 2

SOFTWARE REQUIREMENTS SPECIFICATION

Software Requirement Specification (SRS) is an important part of software development process. It includes a set of use cases that describe all the interactions of the users with the software. Requirements analysis is critical to the success of a project.

2.1 PRODUCT PERSPECTIVE

Time Series Data Mining tool is a unique product that makes use of different algorithms to predict, view similarities, and points out the anomalies in different time varying data sets. It is built in a pluggable fashion where the only requirement at the users end is the browser and a working internet connection.

2.2 PRODUCT FEATURES

The time series data mining tool has many features that distinguishes it from the others available already in the open world. It provides accurate results using the similarity finding, anomaly finding algorithms. The back propagation neural network helps us predicting the future values. On the front end, the user has options to choose the algorithm of her choice. Also, the charts which depict the output are carefully plotted using the google charts API which has been made available by Google Inc. Also, Java beans along with servlets and java server pages and best practices of coding have been followed.

2.3 CONSTRAINTS

During the development of this product, constraints were encountered. Some specific constraints under which the time series data mining tool has are :

- Add Constraints
- Add Constraints

2.4 ASSUMPTIONS AND DEPENDENCIES

- It is assumed that the user of this tool has basic understanding of time series data mining.
- Also, the user must have a decent knowledge of the interpretation of line graphs.

2.5 SPECIFIC REQUIREMENTS

This section shows the functional requirements that are to be satisfied by the system. All the requirements exposed here are essential to run this tool successfully.

2.5.1 FUNCTIONAL REQUIREMENTS

The functionality requirements for a system describe the functionality or the services that the system is expected to provide. This depends on the type of software system being developed. The requirements that are needed for this project are :

- The data sets should be normalized so that the algorithms can be applied effectively.

- A good representation of the results should be made available to the users through proper representation media like graphs.
- TODO ADD SOMETHING HERE.

2.5.2 SOFTWARE REQUIREMENTS

DEVELOPERS MACHINE

- Front End: Java Enabled Browser
- Back End: Java
- Operating System: Windows 7
- JDK 7.0
- Apache Tomcat Server version 7.0
- Eclipse IDE for J2EE Developers
- Active Internet Connection
- JQuery UI and Ajax Libraries

END USERS MACHINE

- Java Enabled Browser
- Active Internet Connection

2.5.3 HARDWARE REQUIREMENTS

- Processor: Intel Pentium 4 or higher version
- RAM: 512MB or more
- Hard disk: 5 GB

SOFTWARE INTERFACES

The Java Runtime Environment (JRE) is required to run the software.

Chapter 3

HIGH LEVEL DESIGN

The software development usually follows Software Development Life Cycle (SDLC). The second stage of SDLC is the design phase. The design stage involves two substages namely High level design and Detailed level design.

High level design gives an overview of how the system works and top level components comprising the system.

3.1 SYSTEM ARCHITECTURE

This section provides an overview of the functionality and the working of the time series data mining tool. The overall functionality of the application is divided into different modules in an efficient way. The system architecture is shown in Figure 3.1

3.2 DATA FLOW DIAGRAMS

A DFD is a figure which shows the flow of data between the different processes and how the data is modified in each of the process. It is very important tool in software

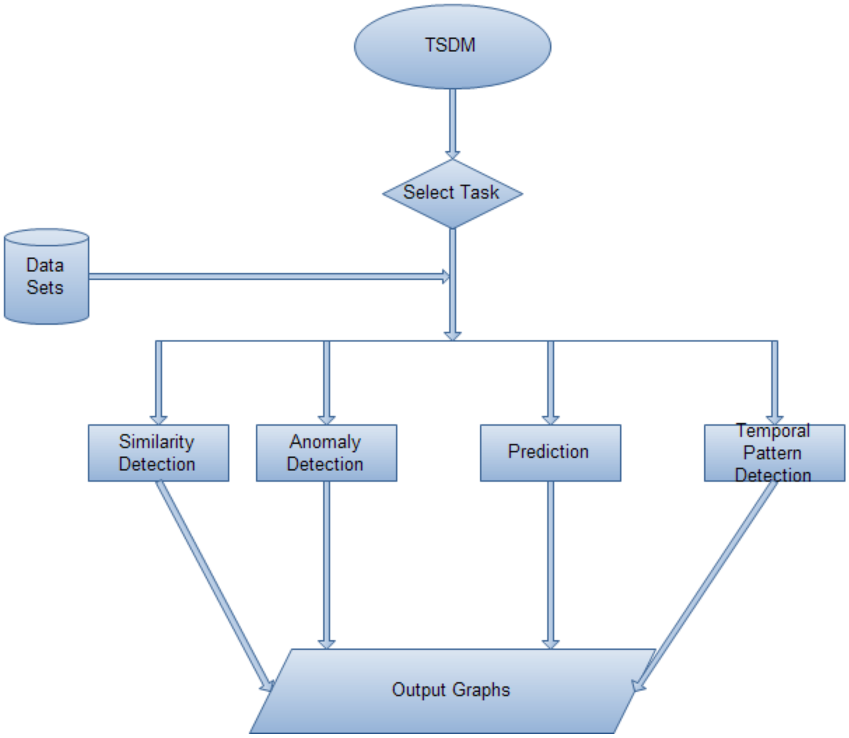

Figure 3.1: System Architecture

engineering that is used for studying the high level design.

There are many levels of DFDs. Level 0 gives the general description and level 1 gives the detailed description. Going higher in the level numbers greater description of the processes will be given.

3.2.1 DFD LEVEL 0

The level 0 DFD is shown in Fig. 3.2 below ehich gives the general operation of the steganographic system There are three major components. Two external entities called sender and receiver and one major system called the steganographic system.

- Sender : The sender is the one responsible to send the data to the receiver. The data to be sent is hidden using the Steganography system. The data is hidden in an image.
- Receiver : The receiver receives the data that is sent to him in the embedded format. The receiver then extracts the data from the embedded format to get the actual data. The receiver makes use of the Steganography system to extract the data.
- Steganography System : This the software which is used to embed the data into the image and also to extract the data from the image.

3.2.2 DFD LEVEL 1

There are two major processes in level 1 DFD as shown in Figure 3.3. The processes invloved in here are :

- Data Embedding: This process represents the actual embedding the data. The inputs given to this process are the data files, cover image. Input image is where the data is to be embedded and the key must also be shared between the sender and the receiver.
- Data Extraction: This process is used to extract the data back from the stego image. This is the reverse process of embedding. Here the input is the stego image and the key.

3.2.3 DFD LEVEL 2

Embedding Phase

The major processes in Level 2 DFD of Embed process on the sender side are shown in Figure 3.4.

- Data Holder: Data holder contains all the data bits, which upon invocation will give the required number of bits of data need to be embedded.
- Pixel Retriever: This retrieves the individual pixels from the cover image.
- Processor: Takes the key as the input, encrypts the data and embeds it into the pixels received from the retriever.
- A final image is formed as a result of this process.

Extraction Phase

The major processes in the Level 2 DFD of the Extraction module on the receiver side is shown in Figure 3.5.

- Data Retriever: The data retriever module extracts the data file from the morphed image which it receives. The key should be present for proper retrieval.
- Pixel Retriever: Retrieves the pixels from the morphed image and pprovides it to the Data Retriever.

Chapter 4

DETAILED DESIGN

4.1 STRUCTURED CHART

Structure charts are used to specify the high level design or architecture of a computer program. As a design tool, they help the programmer in dividing and conquering a large software problem, i.e. recursively breaking a problem down into parts that are small enough to be understood by a human brain. The process is called top-down design or functional decomposition.

Programmers use a structure chart to build a program in a manner similar to how an architect uses a blueprint to build a house. In the design stage, the chart is drawn and used as a method for the client and various software designers to communicate. During the actual building of the program, the chart is continuously referred to as master plan. Often, it is modified as programmers learn new details about the program. After a program is completed, the structured chart is used to fix bugs and to make changes.

The entire program starts with user entering his choice of input as to either embed or extract. Based on this, the GUI module responds appropriately with success or failure. The digital media chosen by user is taken as input for embedding phase. The digital media is encrypted using AES algorithm before embedding to stego image.This encrypted data is embedded onto image using embedding algorithm to get the stego image containing secret data. A key is generated using Diffie Hellman Key exchange which acts as the input for the embedding algorihm.

At the receiver end, the Diffie Hellman key is generated once again. From the stego images, secret data is extracted. It will be in encrypted form. The original input data is extracted by using decompression module of AES algorithm. This gives the hidden file's original content.

4.2 MODULES DESCRIPTION

This section describes the main modules that are used in developing the project. This will help us in understanding the working of individual components.

4.2.1 GUI MODULE

Definition:

This module is the core module which takes the users input to decide upon which operation to be performed. Based upon user choice, the necessary inputs will be taken in this module. If any errors are generated during any operation, then suitable report is displayed to the user.

Resources:

The input files are bitmap images and are chosen to embed data. A session key is generated using Diffie Hellman protocol. Also, the data file to be hidden is taken.

Functionality:

This is the main flowchart which shows all the functions provided by the software. In the beginning, the user is given two options, according to which the user either embeds extracts or exits from the application. This is shown in the decision symbol. According to the decision taken, the operations are performed. If the user chooses to embed, then the functions to select cover image, the secret data, and key generation are performed. Before embedding the secret data encryption is done. After performing these operations, the stego images are sent to the receiver using any wireless medium or LAN. At the receivers end, the receiver implements the functions to extract stego image values, extract secret digits data. After the implementation of these functions, if the user wants to exit from the software, that particular option is also provided to the user. This is shown in Fig 4.2.

Chapter 5

IMPLEMENTATION

The implementation phase of any project development is the most important phase and yields the final solution which solves the problem at hand.The implementation phase involves the actual materialization of the ideas, which are expressed in a suitable programming language. The factors concerning the programming language selection and platform chosen are described in the following sections.

5.1 PROGRAMMING LANGUAGE SELECTION

The programming language chosen must reflect the necessities of the project to be completely expressed in terms of the analysis and the design documents. Therefore before choosing the language, features to be included in the project are decided. The time series data mining project needs the following features in a language to be implemented. Some of the features required are stated as follows:

- J2EE provides us with servlets and JSP which help in dynamically constructing web pages.
- J2EE provides us with Java Beans which help in proper data manipulation.
- JSP and servlets make use of Java backend in a very optimal manner. They have special tags which help us exploit these features.

- Java's core classes are designed from scratch to meet the requirements of an object oriented system.

With these necessities in mind, J2EE is selected as the optimal programming language to implement the project.

5.2 PLATFORM

The TSDM tool was built and designed on Windows Operating system family. They were specifically tested on Windows 7 with Google Chrome and Mozilla Firefox browsers. Because the product is browser based, any user with the broswers mentioned above will be able to run the tool. The product is hence platform independent in the true sense.

5.3 CODE CONVENTIONS

The code standards for the Java programming Language document contains the standard conventions that follows. It includes file names, file organizations, indentation, comments, declarations, naming conventions and programming practices. Code conventions improve the readability of the software.

5.3.1 Naming Conventions

@TODO Add shitty things here

5.3.2 File Organization

@TODO Add shitty things here

5.3.3 Class Declarations

@TODO Add shitty things here

5.3.4 Comments

@TODO Add shitty things here

5.4 DIFFICULTIES ENCOUNTERED AND STRATEGIES USED TO TACKLE

There were a number of challenges that were faced while implementing the Time Series Data Mining tool.Some challenges were challenging and ended up in helping us think innovatively and come up with efficient solutions. Some major problems that were encountered have been stated in brief along with their solutions.

Problem 1

In the initial stages of the project charts4j libraries were used to plot graphs. There were some internal problems with the URL rendering.

Solution

This Problem was solved later by making use of Google's Charts API and java script.

Problem 2

@TODO Add Difficulties Encountered here.. more bullshitting to be done here :P ... (Cool)

Initially the 10 digit key was being entered by the user could not be transmitted to the other end.

Solution

Diffie Hellman key exchange module was later developed which helped in key distribution.

Problem 3

Initially it was difficult to view both the morphed and the original image at the same time.

Solution

This problem was later solved by developing a module in SWT to compare the two images side by side.

Chapter 6

Implementation

Nature seems to have an uncanny knack for problem-solving. Life began as a handful of simple, single-celled organisms barely equipped to survive the harsh environment of planet Earth. However, in the short span of a few billion years, nature has adapted and evolved them into beings complex enough to ponder their own origins. While this is indeed amazing, the truly incredible part is that it all happened according to a simple plan–allow individuals with favorable traits to survive and reproduce, and let die all the rest. This, in short, is the basis for a genetic algorithm.

Chapter 7

Software Testing

7.1 Test Environment

7.1.1 Unit Testing of Data Center Management Model

7.1.2 Unit Testing of Metrics Adapter and Analyzer Module

7.1.3 Unit Testing of Actuator Module

7.2 Integration Testing of the Modules

7.2.1 Integration of Data Center Management Model Service with Workload Metric Generator

7.2.2 Integration of Metrics Adapter and Analyzer with Workload Metric Generator

7.2.3 Integrate Actuator service with RTIM software

7.3 System Testing

7.4 Functional Testing of the GUI

7.4.1 Design Data Center Resources Artifacts

7.4.2 Design Business Services Artifacts

Chapter 8

Experimental Analysis and Results

8.1 Evaluation Metric

8.2 Experimental Dataset

8.3 Performance Analysis

8.4 Inference from the Results

Chapter 9

CONCLUSION AND FUTURE WORK

9.1 Summary

In this mini project, we were able to successfully implement and test the performance of Decision Tree-based classifiers. The Decision Tree classifier was optimized using a Genetic Algorithm to select a subset of the features that were to be used in constructing an optimal decision tree.

Although our program works with generic data samples, it must be noted that when we started this project, our main intention was to classify ground water samples into two classes, namely Potable and Non-Potable Water. However, thanks to the miracle of Object-Oriented Programming Concepts, we were able to extend our application, which was developed in Java and Java SWT. We were able to extend this application to work with any generic samples. Two other samples/ Classification problems were addressed:

- Diagnosing whether a Horse has colic or is healthy, based on its Blood Sample Data.
- Classifying/Determining the quality of a wine based on Data Samples containing itś quality parameters

The hybrid GA /decision tree algorithm needs to be tested further to realize its true potential. Clearly more work needs to be done. The test results show that the Decision Trees constructed using the Genetic algorithm-based feature selector, were more efficient and accurate in classifying the data than the Decision Trees constructed by selecting features manually.

9.2 Limitations

9.3 Future enhancements

Some of the future enhancements are :

1. The application could be made more responsive by using Threads and Parallel/-Cloud Computing
2. The Decision Tree Classifier of this application could be optimized using Neural Networks which are more efficient than Genetic Algorithms.
3. An interesting extension to be explored is the possibility of additional feedback from ID3 concerning the evaluation of a feature set.

Bibliography

[1] Genetic Algorithms -http://en.wikipedia.org/wiki/Genetic_algorithm

[2] Genetic Algorithm for constructing DT - http://www.jprr.org/index.php/jprr/article/viewFile/44/25

[3] Decision Trees - http://web.cecs.pdx.edu/~mm/MachineLearningWinter2010/pdfslides/DecisionTrees.pdf

[4] Project brief for the DT using Horse data sets - https://cs.uwaterloo.ca/~ppoupart/teaching/cs486-spring06/assignments/asst4/asst4.pdf

[5] Supervised and Unsupervised Discretization of Continous Features - http://robotics.stanford.edu/users/sahami/papers-dir/disc.pdf

[6] Hybrid learning using Genetic Algorithms and Decision Trees for pattern sification - http://cs.gmu.edu/~eclab/papers/ijcai95.pdf

[7] Kardi Tutorials on Decision Trees- http://people.revoledu.com/kardi/tutorial/DecisionTree/index.html

Appendices

Appendix A : Source Code

Appendix B : Screen Shots