CHAPTER 5
COMPARATIVE EXAMINATION OF DATA MINING TOOLS IN REPRESENTATIVE BASED SYSTEMS

5.1 INTRODUCTION
The Worldwide technological advancement has brought in an extensive transform in adoption and consumption of open source tools. Since, a large amount of the organizations across the globe convention with a large amount of data to be updated online and dealings are made every second, managing, mining and giving out this dynamic data is very complex. The Successful achievement of the data mining technique requires a cautious assessment of the variety of tools and algorithms available to mining experts. This work provides a relative study of open source data mining tools obtainable to the professionals. The Parameters influencing the preference of apt tools in addition to the real time challenges are discussed. However, it’s glowing established that Representative aid in improving the presentation of data mining tools. This chapter provides the information on Representative-based framework for data preprocessing with execution particulars for the development of enhanced tool in the market.

5.2 OBTAINABLE REPRESENTATIVE REPLICATION TOOLS
The Java-based autonomous Representative developed by IBM, which make available the basic capabilities required for mobility and has a globally unique name. A travel schedule is used to specify the destinations to which the Representative must travel and what actions it must take at each location. In this order, for an aglet to run on a particular system, the target system must be running an aglet host application which provides a platform-neutral execution
environment for the aglet. The aglet workbench includes a configurable Java security manager. The Aglets can communicate using a whiteboard that allows Representative to collaborate and share information asynchronously. Synchronous and asynchronous message passing is also supported for aglet communication. Aglets are streamed using standard Java serialization or externalization. A network Representative class loader is supplied which allows an aglet’s byte code stream and state to travel across a network. The FTP Software Representative Technology is Java-based software designed to manage heterogeneous networks across the Internet using Representative technology. The Representative are autonomous and mobile, and can move to any system in the network which has Representative Responder installed. As the Representative moves from system to system, its tasks may change, depending on the environment of the system it is visiting. The Representative can interrelate with other Representative or with the user, as desirable. But FTP Representative do not require any user interaction based on push technology, they can move from system to system, respond to events, and perform tasks according to criteria predefined by the user. Representative Manager is responsible for launching the Representative. The explorer, from Object Space, Representative-enhanced Object Request Broker (ORB) coded in Java. An ORB provides the capability to create objects on a remote system and invoke methods on those objects. Voyager augments the traditional ORB with Representative capabilities. Voyager Representative has mobility and autonomy which is provided in the base class representative. Representative can move itself from one location to another and can leave behind a forwarding address with a secretary so that future messages can be forwarded to its new location. Specialized Representative, called Messengers are used to deliver messages. Messages can be synchronous, one-way similar to asynchronous, or future, which are asynchronous but return a placeholder that can be used to retrieve a return value at a later time. Representative’s itinerary instructs the Representative as to what operations it needs to perform at each
location. Voyager has a security manager which can be used to restrict the operations Representative can perform.

5.3 ARCHITECTURE DEVELOPMENT

Due to the appearance of online transactions, World Wide Web, data streams, large amount of data is created and is being stored in databases, which is one of the reasons for complexity of data management. Apprentice users may find it difficult to determine the best technique for pre-processing their data to perform data mining. Hence, experts are required to identify the best-suited pre-processing technique. At present, neither data mining tools are efficient in handling dynamic and complex data nor do the users have sufficient knowledge in pre processing the data in terms of data mining domain. However, with the integration of Representative based data mining system, the Representative determines the technique and the parameters that provide the best model for good decision making. Since, the current mining tools are domain specific, this research focused to propose a generic architecture that can pre-process data using Representative of any domain of application. In addition, even a novice user can use the proposed architecture.

Figure 5.1: Proposed Pre-processing Architecture

The proposed architecture in figure 5.1 is the basic model of Pre-processing architecture. The framework is designed with five major Representatives which include User Interface Representative (UI Representative), Coordinator Representative, Clean Representative, Transformation Representative and
Reduction Representative. The responsibilities and capabilities are specific of each Representative. The User Interface Representative provides the user interface with the system. It provides solution analysis autonomously and helps the user to give queries according to the requirement of user. Coordinator Representative is responsible for the coordination of all the tasks that is performed in the system. It determines the pre-processing method to be used based on the data mining task given by the user which is generated according to the Meta knowledge which the Representative maintains. It has access to the data repository that can be updated dynamically and provides data to the other Representative. Coordinator Representative also provides adaptive profiling user data and checks to identify the data types and attributes in the database. Its identifies the problems the data has and save the knowledge and corresponding pre-processing technique that is best for it.

5.4 PERFORMANCE OF THE REPRESENTATIVE BASED PREPROCESSING ARCHITECTURE

The proposed architecture model is a client server model with a basic Model View Controller Architecture. The exceeding proposed architecture can be implemented in similar ways like implementing both Representative and request or to modify the existing code of an application to enable the necessary communication.

![Pre-processing Architecture](image)

Figure 5.2: Pre-processing Architecture
The pre-processing methods are all available in various open source data mining tools and open source Representative simulation tools. The Representative framework which we proposed will have the following requirements

1. Capability to add intelligence to applications.
2. Intelligent Representative framework that is practical to solve real-world problems.
3. Architecture must be flexible enough to support any applications.
4. Intelligent Representative increases the functionality of the application and can communicate with each other and other applications.
5. The Representative can call the shots and monitor and drive the applications.

Useful requirement of Representative Framework
The functionality of Representative Framework for all time contain the following condition

1. It must be easy to add an intelligent Representative to an existing application.
2. A graphical construction tool must be available to compose Representative out of other Java components and other Representative.
3. The Representative must support a relatively sophisticated event processing capability. Representative will need to handle events from the outside world, other Representative, and signal events to outside applications.
4. Domain knowledge can be added to Representative using if-then rules, and support forward and the direction of the back rule-based processing with sensors and effectors.
5. The Representative must be able to learn to do classification, clustering, and prediction using learning algorithms.
6. Multi Representative Applications must be supported using a KQML-like message protocol. The Representative should be determined. That is, once Representative is constructed, there must be a way to save it in a file and reload its state at a later time. Due to the availability of open source data mining tools and Representative simulation tools for the implementation of the architecture, it is now possible to append Representative to the existing application, thereby extending the basic capabilities of the application which requires a minimum of pre planning of the application developer.

5.5 CONCLUSION
State of the art advancement in technology enables organizations to store, process and update huge and complex data dynamically. The development and application of data mining techniques requires the use of right choice of software tools. Additional, recent data mining tools are expensive to get the updated knowledge model. This work provides information on two aspects of data mining tools namely to elucidate the comparative analysis of various open source data mining tools and to put forth the challenges which exist in the existing data mining tools. However, Representative is known to aid in improving the presentation of data mining tools. This work has therefore projected to integrate existing data-mining tool with Representative in order to execute an effective data pre-processing architecture. The useful specifications elucidate here enable the application developer to correctly analyse, assess and develop the data pre processing tool for improved data management.