1. INTRODUCTION

Data mining is the extraction of knowledge from large amounts of data. It can be viewed as a natural result of evolution of information technology. The early decades witnessed a lot of focus on data collection followed by an upsurge in the interest on data management. After this, the focus has shifted to advanced data analysis tasks. According to Han and Kamber[35], the abundance of data coupled with the need for advanced analysis tools has created a “data rich, information poor” situation.

Although the terms “data mining” and “knowledge discovery from data” are sometimes used synonymously, data mining is actually a step in the process of knowledge discovery from data. Knowledge Discovery from Data (KDD) comprises the following steps:

1. Data cleaning (Removal of noise and inconsistent data)
2. Data integration (Combination of multiple data sources)
3. Data selection (Retrieval of relevant data)
4. Data transformation (Consolidation of data into a form appropriate for mining)
5. Data mining (Application of intelligent methods to extract patterns)
6. Pattern evaluation (Identification of truly interesting patterns)
7. Knowledge presentation (Presentation of mined knowledge to the user)
1.1 COMPONENTS OF A DATA MINING SYSTEM

According to Han and Kamber[35], a data mining system consists of the following components:

- Database, Data warehouse, World Wide Web (WWW) or other information repository - Contains data on which data cleaning and data integration techniques are applied
- Database or Data warehouse server - Responsible for the fetching of relevant data
- Knowledge base - Domain knowledge used to guide the search for interesting patterns. Examples: User beliefs, concept hierarchies
- Data mining engine - Consists of a set of modules for characterization, association, classification, clustering etc.
- Pattern evaluation module - Contains interestingness measures and interacts with the data mining engine to focus the search toward interesting patterns

1.2 DATA MINING FUNCTIONALITIES

Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks[35]. Data mining tasks can be categorized as descriptive and predictive. Descriptive tasks are used to characterize the general properties of data in the database, while predictive tasks perform inference on current data to make predictions.
1.2.1 Characterization and Discrimination

Data characterization is the summarization of the general characteristics of a target class of data. For example, it may be interesting to study the characteristics of programs which report a very low reliability. Data discrimination is the comparison of the general features of objects of a target class with those of objects of one or more contrasting classes. For example, the features of programs with low reliability can be compared with the features of programs with high reliability.

1.2.2 Mining Frequent Patterns, Associations and Correlations

Patterns that occur frequently in the data are called frequent patterns. A frequent itemset refers to a subset of items that occur together frequently. For example, the mining of a software engineering database might reveal the following association rule:

 Exists (review,S) => Exists (Correctness,S) [support=1% Confidence=50%]

The association rule states that if there exists reviews for Software S, there is a 50% chance that the software will deliver the correct functionality. A 1% support indicates that of all the data available, 1% of data indicates that reviews and correctness occur together.

1.2.3 Classification and Prediction

Finding a model that distinguishes classes is termed as classification. The model thus found is used to predict the class of objects for which the class label is unknown. The model can be expressed as a series of if-else rules or a
decision tree. For example, a classifier may identify the number of errors found during testing as a major factor in deciding if a program belongs to the class of “reliable” or “unreliable”.

1.2.4 Cluster Analysis

The objects are grouped into classes based on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity. For example, the software developed by an organization can be clustered so that software within a cluster has a high degree of similarity.

1.2.5 Outlier Analysis

Data in the database that do not comply with the general behavior of other objects are called outliers, and the mining of these outlier data is called outlier analysis. For example, it may be interesting to analyze the properties of programs with exceptionally high reliability.

1.3 TYPES OF DATABASES

Data mining is not specific to any kind of data. Zaiane[97] claims that data mining should be applicable to any kind of information repository. But the challenges of mining posed by different kinds of data vary significantly.

1.3.1 Flat Files

Flat files containing text or binary data are the most obvious candidates for data mining. Mining of text data is referred to as text mining. It generally entails analyzing a large volume of textual data to ascertain correlations or other patterns. In the domain of software engineering, the mining of source
code is generally performed using text mining techniques. Software requirement specifications and test case documents containing textual information are attractive candidates for text mining.

1.3.2 Relational Databases

Relational databases containing information structured as tables where each row is termed as a tuple and each column as an attribute provide excellent support for several data mining algorithms. Data mining algorithms that target relational databases are more versatile than those for flat files[97].

Structured Query Language (SQL) is the standard language for accessing relational databases, and data mining algorithms can also leverage the capabilities of SQL for data transformation and consolidation.

1.3.3 Data Warehouses

Data warehouses are structured repositories of data from multiple, heterogeneous sources. Data warehouses facilitate analysis of data from different dimensions. A data cube facilitates analysis of data along multiple dimensions and each cell usually contains the value of some aggregate measure. As Zaiane[97] states, because of their structure, the pre-computed summarized data they contain and the hierarchical attribute values of their dimensions, data cubes are well-suited for fast interactive querying and analysis of data at different conceptual levels, known as On-Line Analytical Processing (OLAP). OLAP operations allow the navigation of data at different levels of abstraction, such as drill-down, roll-up, slice, dice etc.
1.3.4 Transaction Databases

A transaction database contains information pertaining to day-to-day transactions including a time stamp, identifier and the associated items. Transaction information is generally stored in flat files or in two normalized relational tables - one containing the transactions and the other containing the transaction items. A typical example for the scenario is the market-basket analysis that attempts to track transactions that occur together or in a sequence.

1.3.5 Multimedia Databases

Mining of multimedia data such as audio, video and graphics stored on a flat file or object-oriented or object-relational databases is even more challenging due to the high dimensionality of the involved data. This may entail application of techniques from computer vision and computer graphics.

1.3.6 Spatial Databases

A spatial database stores a large amount of space-related data, such as maps, preprocessed remote sensing or medical imaging data. They carry topological and distance information, usually organized by sophisticated, multidimensional spatial indexing structures.

1.3.7 Time-Series Databases

Time-series databases containing time-related information like market share prices have a continuous flow of data feeds that presents novel challenges, and the mining of these databases entails evolution analysis and trend prediction.
1.3.8 World Wide Web

The World Wide Web (WWW) is a huge repository of information, and mining this is commonly classified into - Web Content Mining - which encompasses the documents, Web Structure Mining - which focuses on the hyperlinks and relationships between documents and Web Usage Mining - which focuses on the usage patterns of web pages. Web Mining can greatly enhance the usability of the WWW.

1.4 APPLICATIONS OF DATA MINING

The potential applications of data mining are enormous and are limited only by the creativity of the researcher. A few applications adapted from Padhy et al.[67] are listed below.

1.4.1 Health care

Data mining techniques can be very useful for the health care domain but the applicability is mainly constrained by the non-availability of clean health care data. Text mining can be applied as many health care documents are text based. Multimedia mining for MRI (Magnetic Resonance Image) images can also be explored to uncover previously unknown patterns that can greatly aid early diagnosis of serious diseases like cancer.

1.4.2 Education

Data mining tools can greatly aid in bridging the knowledge gap between various higher educational institutions and can serve as good decision
support systems for the government in the formulation of various educational policies.

**1.4.3 Manufacturing Industry**

Data mining can be applied to industrial use, and the knowledge derived can be of great utility to the manufacturing industry. But relationships and patterns in manufacturing data tend to be complex, and this dictates the need for specialized tools for the manufacturing sector.

**1.4.4 Business**

Data mining techniques like the market-basket analysis can be used to enhance the profitability of business organizations. Data can be of utmost help in strategic decision making of business organizations. Classification can help classify potential customers on the basis of their characteristics. Clustering can be utilized to build clusters of customers upon which targeted marketing can be done based on the common characteristics of the customers in each cluster. Data Mining can be applied in various novel ways to improve profitability of businesses. It can also be used to reduce costs. Data mining can also be applied for risk management.

**1.4.5 Agriculture**

In countries like India, where agriculture continues to be the mainstay of many people, data mining tools can be utilized to help the farmers in making sound decisions. Agricultural data collected over a period of time can be mined to uncover novel patterns that can greatly aid the farmer in making wise
decisions, especially in a country like India, where the whims and fancies of
the monsoon wreak havoc on the life of farmers. Data mining can be used to
uncover novel ways of making effective use of monsoon rains whose patterns
can be found out from monsoon related data accrued for a period of time by
applying data mining techniques.

1.4.6 Customer Relationship Management (CRM)

Data mining can be applied to various aspects of CRM particularly
customer retention. The application of data mining in CRM has attracted a
good deal of attention of the researchers in the industry. Customer profiling can
be greatly aided by data mining. Customers with common characteristics can
be clustered and marketing campaigns can be organized targeting the cluster of
customers highly likely to be interested in the product or service. Sending alerts
to customers on the basis of their likelihood of interest in a particular product
or service, suggesting other products and services that a customer may be
interested in are some of the facets of CRM that can be effectively addressed
by Data mining.

1.4.7 Computer and Network Security

Data mining techniques can greatly aid researchers in the domain of
counter and network security. Intrusion detection techniques can greatly
benefit from data mining tools. The data pertaining to traffic pattern on a
network can be mined to generate useful information about anomaly detection.
Data leakage detection is one of the most attractive domains for application of
Data mining. Data pertaining to security related breaches on a software product can be mined to gain valuable knowledge about the attackers and that can be effectively utilized to prevent recurrence of such incidents in future. Building profiles of attackers and the kinds of attacks they can potentially launch on applications can go a long way in ensuring security. Data mining can be effectively applied in this area.

1.4.8 National Security

Data pertaining to known terrorists and criminals can be mined using techniques like clustering to identify criminals engaging in terrorist activities throughout the world. Elovici et.al.[22] provide an example of using data mining techniques to identify terrorist-related activities on the web.

1.4.9 E-Commerce

E-Commerce is one of the most potential and promising candidates for the application of data mining, thanks to the availability of plentiful, reliable data and the possibility of quick measurement of the return on investment.

1.4.10 Prediction for various Engineering Applications

Data mining tools can serve as valuable predictors for many engineering applications. The most difficult problem in many engineering projects is the estimation of cost and schedule, and data mining tools can be applied to data pertaining to past projects to predict cost and schedule for the current project.
1.5 SOFTWARE ENGINEERING

Software engineering is the application of systematic, disciplined and quantifiable approach to the design, development, operation and maintenance of software. It had its origins in the late 1960’s and was primarily aimed at addressing the issue then described as “Software Crisis”. Software engineering is primarily concerned with the development of quality software on time and within the cost estimates.

An underlying principle of software engineering is the distinction between a software product and a software process. In the words of Pressman[70], “a computer software is a product that software engineers design and build. It encompasses the programs that execute, together with the associated documents that contain not only text and numbers but also pictorial representations”. A software process “is a series of predictable steps that one goes through in the development of software”. A software process provides a road map to be followed in the engineering of software.

Pressman[70] views software engineering as a layered technology with the following layers:

- The bottommost layer is the “quality focus” layer as any engineering approach must be based on an organizational commitment to quality.
- The next layer is the “process” layer that provides a framework with a number of Key Process Areas (KPAs), and these KPAs provide the basis for management.
• The next layer – the “methods” layer is concerned with the “how to” details. These methods rely on a set of basic principles.

• The “tools” layer provides automated or semi-automated support for the various software engineering activities. Computer Aided Software Engineering (CASE) is the integration of the tools so that they provide continuous support throughout the development of the software.

Pressman[70] classifies the work associated with software engineering into three phases:

• The definition phase that focuses on “what”, and “where” the key requirements of the software and the system are identified. Three common activities of this phase are: system engineering, planning and software requirements analysis.

• The development phase focuses on “how”. Three specific activities of this phase are software design, code construction and testing.

• The support phase deals with the changes to the software that occur due to discovery of problems or new requirements. Four specific activities of this phase are: correction that entails the correction of faults discovered in the system by the customer; adaptation that modifies the software to adapt to changing business environment; enhancement that deals with extending the functionality provided by the original software, and prevention that entails making corrections so that future enhancements and adaptations are easy to accomplish.
1.6 APPLICATION OF DATA MINING TO SOFTWARE ENGINEERING

The broad applicability of data mining for a variety of disciplines has raised questions about its applicability to the domain of software engineering. Alvarez et al.[5][6] apply data mining tasks to software development projects. They suggest an interesting possibility of application of data mining to software engineering: if there is a database that contains data pertaining to various attributes of software in the initial phases and in the final phases, it might be possible to discover novel and interesting association rules between various parameters, and these rules can be used for prediction for a newly started development project.

Alvarez observes that the lack of databases containing the values of various parameters that condition a software development project is the main challenge when attempting to apply data mining to software engineering. But now a days, this challenge is overcome thanks to the availability of many simulation systems.

Hayes et al.[39][40] state that there are two ways by which data mining can be applied to software engineering. The first is in exploratory studies of existing artifacts like source code, test case documentation and the like to find out novel patterns. The second is for the improvement and automation of software life cycle processes. This can improve the speed of performance of
many activities but this is not error-tolerant. So in this case, the results are presented to the analyst who assesses the correctness and gives the final results. Wahidah Husain et al.[89] categorize software engineering data into:

- **Sequences** such as execution traces collected at run time. For example, the method calls data.
- **Graphs** such as dynamic call graphs generated from the source code.
- **Texts** such as code comments and documentation.

Mining of sequential data can be helpful in flaw or bug detection. Frequent item-set mining and frequent sequence mining can be very useful for this purpose.

Most Software Engineering(SE) data can be conveniently expressed as a graph, and for this reason graph data mining is an active area of research in SE data mining. Mining of software behavior graphs collected from program execution can be used to disclose traces of bugs.

According to Gegick et al.[26], 80% of information in computers is stored as text. In the context of software engineering, useful text data include software requirements specification, bug reports and the like. As an example of applying text mining to software engineering, Wahidah Husain et al.[89] provide an example of applying text mining to classify bug reports (which contain text data) into Security Bug Reports(SBRs) and Non-Security Bug Reports(NSBRs).
Decision making in software engineering is generally done based on intuition and past experiences. Hassan[37] states that software repositories contain a wealth of information and classifies repositories into:

- Historical repositories such as source code, bug reports, and communications pertaining to a project.
- Run-time repositories such as deployment logs containing information about the execution of a particular application at a particular deployment site.
- Code repositories such as google code which contain source code developed by various developers.

According to Hassan[37] notwithstanding the availability of repositories, the data available in these repositories have not been the focus of software engineering researchers due to the following reasons:

- Access Limitations – The reluctance of software organizations to grant access to repositories pertaining to their software to researchers is the main road block. Available data pertain to academic projects which are small and uninteresting at times.
- Data Extraction Difficulties – As the repositories are not built keeping in mind the requirement of automated extraction, they tend become difficult to mine.
The availability of open source systems has greatly alleviated the first problem. With the advent of open source systems, quick access to code developed by hundreds of developers is available.

Hassan and Xie[36][38] suggest the usage of some data mining techniques like classification, association-rule mining to software engineering data. Association rule mining can be used to find highly correlated method call pairs and identification of conflicting patterns that can greatly aid error discovery. For example, if out of 1000 runs, 999 times method A is called when method B is called. The single case when method A is not called may point to an error condition. Data mining can be also applied to the detection of copy-pasted code segments and identify bugs in those segments.

Sharma and Sharma[78] given an amalgamation of automated testing and data mining. They state the advantages and challenges of test automation and explain how data mining can be efficiently used to model tested systems, recover system requirements and designing a minimal set of regression tests.

Jensen and Scacchi[44] illustrate the application of data mining in process discovery of open source software systems. Given the difficulty of process discovery, data mining can prove to be extremely useful in this area particularly in the context of open source software as data pertaining to such systems are publicly available. The open source software development web repositories contain process data encoded as structure, content and usage and
update patterns. They used a combined application of text and link analysis techniques for the purpose.

Lo and Khoo[55] described a solution to the complex problem of software specification discovery from program artifacts. As software is prone to change and these changes are often carried out by people not originally involved in the development, program comprehension takes enormous time. Often the original software specification is out-dated or missing. A solution to this problem is to apply data mining techniques to discover software specifications from program artifacts.

There is still enormous literature available on applying data mining to solve complex software engineering problems. All these studies indicate that data mining holds huge promises in discovering solutions to complex problems in software engineering.

1.7 CHALLENGES IN MINING SOFTWARE ENGINEERING DATA

Xie et al.[94] state the following as challenges in mining SE data.

- Requirements Unique to SE – Most available data mining tools are general purpose, and applying these tools to mine software engineering data undermines the requirements unique to software engineering. On the one hand, software practitioners lack the ability to modify data mining algorithms to tune them for software engineering, and on the other hand, data mining researchers lack the knowledge about software engineering preventing them from developing tools fit for SE data.
• Complex Data and Patterns – Existing data mining algorithms may not be able to produce desired pattern representations in the context of SE. In many cases, there is a need to mine multiple kinds of data together.

• Large Scale Data – Execution traces produced from an even average sized program can be huge, thereby challenging the abilities of data mining algorithms.

• Just-In-Time Mining – According to Xie et al.[94] provision of rapid feedback is the necessity of the day and for this software, engineers must be able to mine SE data on the fly.

1.8 SCOPE OF THE RESEARCH

The potential applications of data mining in software engineering are enormous, and the research is not an exhaustive attempt to address all of these. This research attempts to apply Genetic k-means clustering for software quality estimation. Here, the Total Within Cluster Variation (TWCV) serves as the objective function. The effectiveness of applying Genetic k-means clustering is demonstrated by comparing the results obtained using this approach and k-means clustering.

This research work applies Genetic Algorithm (GA) based association rule mining for identifying important human factors which impact software quality. It also discusses software quality, software quality factors and human factors in software quality.
This research also attempts to develop a GA based software reliability classification. It discusses classification and different classification algorithms. It also provides the details of reliability and corresponding classification algorithms.

1.9 ORGANIZATION OF THESIS

The rest of the thesis is divided into six chapters.

Chapter 2 discusses the use of data mining in software engineering. It also discusses the applications of data mining for software, Text mining, prediction, data mining for testing and reviews the literature on application of data mining to software engineering.

Chapter 3 presents the application Genetic k-means clustering for software quality estimation. Details about clustering and different types of clustering are given. Automation clustering of software system using genetic algorithm and details of genetic algorithm are described. Finally, software quality estimation using k-means and Genetic algorithm is given and results are discussed.

Chapter 4 deals with usage of GA based association rule mining for identifying important human factors which impact software quality. It also discusses software quality, software quality factors and human factors in software quality. At last, a GA based association rule mining approach is given for recognizing important human factors and the results are discussed.
Chapter 5 describes about GA based software reliability classification. It discusses classification and different classification algorithms. It also provides details of reliability and corresponding classification algorithms. Finally, a GA based software reliability approach is given and the results are discussed.

Chapter 6 provides the conclusion and future enhancement of this work. The contributions of the proposed work and their future trends are discussed.

Lists of references and publications containing details are furnished afterwards.