CHAPTER 1
INTRODUCTION

The development in the field of information technology witness generation of huge amount of data of different kinds. Analysis of this huge pile of data for arriving specific information or for the latest trend analysis requires development of new data analysis technologies. The analytical output of the data is useful for data scientist for trend analysis and forecasting. Data mining is a highly used technique for data analysis in the current scenario. This analytic process is specially designed to explore large amounts of data, generated from different domains, like business and market-related aspects. The analytical output consists of specific patterns and systematic relationships that differ between themselves in huge ways. The final goal of data mining is the prediction of most direct outcome of current business prospects. This technique is discussed in detail in the following sections.

1.1 About Data mining

Data mining is a process of extraction of desired or previously unknown and useful information from huge pile of raw data. It is a semi-automatic or fully automatic method to discover patterns and rules of data. Data mining can also be used for prediction and forecasting of data-trends. In data mining suitable computer programs are built to automatically scan the data for specific patterns based on defined rules. Resulting patterns help accurately predict the future data trend. Machine learning provides the technical basis for data mining. Using data mining, one can deduce hidden knowledge by examining, the data structure. The data mining is comparable to On-Line Analytical Processing (OLAP) which is a model-based approach that builds a data model in advance. Data mining is a data-driven approach,
where the pile of data is examined to look for specific desired data structure using data mining algorithms.

Data mining technique is also known by many alternative names such as Knowledge Discovery in Databases (KDD) and predictive analytics. Though originally, data mining is not same as machine learning, nowadays data mining is synonymous with machine learning. This is because data mining gives users the required business insights for effective and firm decision making. In data mining, the hidden predictive information is an extraction from the large database which is a powerful new technology that has a great potentiality to help companies focus on the most important information in their data warehouses. The data mining tools predict future trends and behaviours, allowing business you to make proactive knowledge-driven rational decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that were traditionally too time-consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. Most companies already collect and refine massive quantities of data.

1.2 Data Mining Concepts

Data mining is popularly known as KDD. It is an automated extraction of patterns representing knowledge implicitly captured and stored in large databases, data warehouses, the web and other massive information repositories or data streams. The core of the KDD technique is the application of specific data-mining methods for pattern discovery and extraction. KDD refers to the overall process of discovering the useful knowledge from data mining. Data mining is the application of specific algorithms applied for extracting patterns from data. It involves the evaluation and interpretation of patterns to make the decision qualifying the knowledge. It also
includes the choice of encoding schemes, pre-processing, sampling and projections of the data before the data mining step.

- The noise and inconsistent data is removed in data cleaning.
- Multiple data sources may be combined in data integration.
- Data relevant to the analysis task are retrieved from the database in data selection.
- Data is transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations in data transformation process.
- Data mining is an essential process where intelligent methods are applied to extract data patterns.
- Pattern evaluation is done to identify the truly interesting patterns representing knowledge based on interesting measures.
- Visualization and knowledge representation techniques are used to present mined knowledge to users through knowledge presentation.

The term Knowledge Discovery in Databases or KDD, in short, refers to the broad process of finding knowledge in data and emphasizes the "high-level" application of particular data mining methods. There are two "high-level" primary goals of data mining, in practice prediction and description. They are

- Prediction involves using variables or fields in the database to predict unknown or future values of other variables of interest.
- Description focuses on finding human-interpretable patterns describing the data.

The relative importance of prediction and description for particular data mining applications can vary considerably. However, in the context of KDD, description tends to be more important than a prediction.
This is in contrast to pattern recognition and machine learning applications (such as speech recognition) where prediction is often the primary goal of the KDD process. Data mining refers to an application of algorithms for extracting patterns from data without the additional KDD process.

Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources that can be integrated with new products and systems as they are brought on-line. Data mining techniques are the result of a long process of research and product development. This evolution begins when business data is first stored on computers, continued with improvements in data access and more recently with generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process beyond retrospective data access and navigates to prospective and proactive information delivery. Data mining is ready for application in the business community because it is supported by three technologies that are now sufficiently mature. Data mining is the process of sorting through large data sets identifying patterns and establishing relationships to solve problems through data analysis. Data mining tools allow enterprises to predict future trends.

1.3 Stages of Data Mining

The data mining process consists of the following three stages.

- Preliminary exploration of the data,
- Pattern identification with validation of data and
- Model deployment (i.e. the model to generate predictions).

For high-speed data streaming the supervised and unsupervised techniques of data mining concepts are used. The supervised data mining (also known as predictive or directed) and unsupervised data mining (also known as descriptive or undirected)
categories encompass functions capable of finding different hidden patterns in large data sets.

The figure 1.1 explains that the data resources are a component of information technology infrastructure that represents all the data available to an organization, whether they are automated or non-automated. Information can be gathered from a range of sources. Likewise, there are a variety of techniques to use when gathering primary data. A data model is an abstract model that organizes elements of data and standardizes how they relate to one another and properties of the real world entities. Deployment model is the use of data mining within a target environment. In the deployment phase, insight and actionable information can be derived from data. Deployment can involve scoring (the application of models to new data), the extraction of model details (for example the rules of a decision tree), or the integration of data mining models within applications, data warehouse infrastructure, or query and reporting tools.

1.4 Data Mining Techniques

1.4.1 Classification Analysis

The classification analysis is used to retrieve important and relevant information about data and metadata. It is used to classify different data in different classes. Classification is similar to clustering in a way that it also segments data records into different segments called classes. But unlike clustering, here the data
analysts would know different classes or cluster. So, in classification analysis algorithms are used to decide how new data should be classified.

1.4.2 Analysis of Clustering based classification

The cluster is a collection of data objects; those objects are similar within the same cluster. That means the objects are similar to one another within the same group and they are rather different or they are dissimilar or unrelated to the objects in other groups or other clusters. Clustering analysis is the process of discovering groups and clusters in the data in such a way that the degree of association between two objects is highest if they belong to the same group and lowest otherwise. A result of this analysis can be used to create customer profiling.

Data mining techniques are divided into two main classes.

- The directed or supervised method: In this approach known examples are used and gathered information is applied to unknown examples to predict selected target variable(s).
- The undirected or unsupervised method: In this method, new patterns are discovered as a whole in the dataset.

Some of the most important directed techniques include classification, estimation and forecasting. Classification means to examine a new case and assign it to a predefined discrete class. Examples are assigning keywords to articles and assigning customers to known segments. Very similar estimations are trying to estimate a value of a variable of a new case in a continuous defined pool of values. For example, estimate the number of children or the family income. Forecasting is somewhat similar to classification and estimation. The main difference is the forecasted value that cannot be checked.
Clustering and affinity grouping is the most common undirected techniques known so far. In undirected data mining, the user has no predictable agenda an example of clustering is scanning through a large amount of undifferentiated customer information to see if they fall into certain natural groupings with meaningful structures. Affinity grouping is a special kind of clustering method; it identifies events or transactions occurring simultaneously. Affinity grouping is market basket analysis and it is a well-known example of it. The market basket analysis identifies the items that are sold together at the same time.

1.5 Data Mining Parameters

In data mining, association rules are created by analyzing the data for frequent patterns and use the support and confidence criteria to locate the most important relationships within the data. Support is how frequently the items appear in the database, while Confidence is the number of times if/then statements that are accurate. Other data mining parameters include Sequence or Path Analysis, Classification, Clustering & Forecasting. Sequence or Path Analysis parameters look for patterns where one event leads to another later event. A Sequence is an ordered list of sets of items and it is a common type of data structure found in many databases. Classification algorithms predict variables based on other factors within the database. A Classification parameter looks for new patterns and might result in a change in the way the data is organized.

1.6 Machine Learning

The machine learning terminology has been classified into the following categories given below in figure 1.2

A supervised learning investigates the training data and generates the desired output, which can be used for mapping given data. Unsupervised learning is the machine learning process of inferring a function to describe the unknown structure
from unlabeled data. Reinforcement learning is the machine learning for deciding by behaviourist psychology.

![Diagram of Types of Machine Learning](image)

**Figure 1.2 Types of Machine Learning**

### 1.6.1 Classification

The classification algorithm identifies categories of a set of a new item belonging to base on a training data set where the classification is known. Classification is a supervised learning task. Some of the most widely used classifiers are neural networks, support vector machines, k-nearest neighbours, naïve Bayes, decision trees and radial basis functions. The decision tree classifiers do not give good theoretical basis like in neural networks for classification but work extremely well in practice.

Especially used with methods like Random Forests, the decision trees are known to outperform most of the other classifiers. As a supervised data mining method, classification begins with the method described above. To use the data on customers who have and have not defaulted for the extended period as a build data (or training data) to generate a classification model. The algorithms will look for customers whose attributes match the attribute patterns of previous defaulters/non-defaulters and categorize them according to which group they most closely match. Using these groupings as indicators of which customers are most likely to default. Similarly, a classification model can have more than two possible values in the target attribute.
Classification is the most important and frequently used technique in data mining. It is a process of finding a set of models that describe and distinguish data classes or concepts. The derived model may be represented in various forms such as classification (IF-THEN) rules, decision tree, neural networking, etc. In data mining, classification is one of the most important tasks. It maps the data in to predefined targets. It is a supervised learning as targets are predefined. The aim of the classification is to build a classifier based on some cases with some attributes to describe the group of the objects. Then, the classifier is used to predict the group attributes of new cases from the domain based on the values of other attributes. The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next, the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes.

The classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks. A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments and so on.

Credit rating would be the target, the other attributes would be the predictors and the data for each customer would constitute a case. Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm. The simplest type of classification problem is binary classification. In binary classification, the target attribute has only
two possible values, for example, high credit rating or low credit rating. Multiclass targets have more than two values for example, low, medium, high, or unknown credit rating. In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to the data set in which the class assignments are unknown.

The classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model. The scoring in a classification model results in-class assignments and probabilities for each case. For example, a model that classifies customers as low, medium, or high value would also predict the probability of each classification for each customer. The classification has many applications in customer segmentation, business modelling, marketing, credit analysis and biomedical and drug response modelling. Classification consists of assigning a class label to a set of unclassified cases.

- Supervised Classification: The set of possible classes is known in advance.
- Unsupervised Classification: Unsupervised classification is called clustering. In this, the set of possible classes is not known but after classification can try to assign a name to that class.

The classification is the process of data analysis that extracts models describing important data classes. A bank credit card officer needs to analyze the data to learn about the safety of credit card applicants of a bank. Data classification is classified into two steps. They are learning step and classification step. Learning step is used to
construct the classification model as shown in figure 1.3. Classification step is used to predict class labels for giving data as shown in figure 1.3.

![Classification model](image)

**Figure 1.3 Classification model**

All the raw data are classified into useful information according to the specification. The classification is one of the best ways to categorize the data into the useful format to make the decision as well as prediction. So the classification step is a necessity to make decision and prediction process in the data mining. If the raw data is classified once, then the computation makes the result very easily according to the classified data. The classification step is used to predict class labels for giving data as shown in figure 1.4.

![Classification predicts its class label](image)

**Figure 1.4 Classification predicts its class label.**
The classification has numerous applications such as fraud detection, medical diagnosis and performance prediction and so on. Evaluating and comparing the performance of the classification techniques are generally classified into four major techniques are frequency table, covariance matrix, Similarity and other functions as shown in figure 1.5.

**ZeroR (Zero Error Report)**

It is the simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class). Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for another classification method.

**Decision tree**

It is like a tree structure. It consists of nodes. Attributes are represented by internal nodes, each node outcome is passed to next node by the branch and class labels are represented by terminal nodes. The decision tree is used to examine data and induce the tree and its rules which will be used to make decisions. The output of the decision tree can be understood by the user as well as nontechnical person.
**K-Nearest Neighbor**

It is a nonparametric method because estimation parameters are not involved in it. It includes two steps such as inductive steps and deductive steps. Inductive steps can be used to construct a classification model from given data. Deductive steps can be used to test the model.

Several anomaly detections are obtained by the concept of K-nearest neighbour. The K-nearest neighbour is one of the best methods that are used in the credit card fraud detection. It is used to support supervised learning algorithm. Three main factors are used to manipulate the performance of K-nearest neighbour; the nearest neighbours are located by the distance metric. The classification is obtained by it through the applied rules and the new model is classified by the number of neighbours. High performance has been obtained always by its rule. It can be used to improve the performance through a genetic algorithm. If the value of K is too large then it will reduce the noise data set. It consists of training data, such as genuine and fraudulent.

**Support Vector Machine**

Both linear and nonlinear data classification can be obtained through Support Vector Machine. It belongs to a class of supervised learning algorithm. It has a hyper plane to separate two classes. It produces high accurate results. It can be used for numeric prediction as well as classification. It belongs to a class of supervised learning algorithm. It is considered in the research work due to its nature of separating the two classes.

**1.6.2 Ensemble Classification:** Ensemble classification includes the following.

- Set of Classifiers
- Decisions combined in ”some” way
- Often more accurate than the individual classifiers
- Properties of base learners
1.6.3 Methods of Ensemble Classifiers

Numerous methods have been suggested for the creation of an ensemble of classifiers. Some of them are given below.

- Using different subset of training data with a single learning method
- Using different training parameters with a single training method (e.g. using different initial weights for each neural network in an ensemble)
- Using different learning methods.

**Bagging Ensemble Classification**

Bagging ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. In Bagging ensemble, if there is a training set of size $t$, then it is possible to draw $t$ random instances from it with replacement (i.e. using a uniform distribution), these $t$ instances can be learnt and this process can be repeated several times. As shown in figure 1.6, Bagging (Bootstrap Aggregate) is an ensemble method that creates random samples of the training data set (subsets of training dataset) to build a classifier for each example. Finally, results of these multiple classifiers are combined using average or majority voting. Bagging helps to reduce the variance error.

![Figure 1.6 Bagging Ensemble Methods](image-url)
Figure 1.6 shows the block diagram of bagging which combines multiple classifiers to perform ensemble classification.

**Boosting Ensemble Classification**

Another method of the first category is called boosting. AdaBoost is a practical version of the boosting approach. Boosting is similar in overall structure to bagging, except the one that keeps track of the performance of the learning algorithm and forces it to concentrate its efforts on instances that have not been correctly learnt. Instead of choosing the t training instances randomly using a uniform distribution, one chooses the training instances in such a manner as to favour the instances that have not been accurately learnt. After several cycles, the prediction is performed by taking a weighed vote of the predictions of each classifier, with the weights being proportional to each classifier's accuracy on its training set.

Boosting algorithms are considered stronger than bagging on noise-free data. However, there are strong empirical indications that bagging is much more robust than boosting in noisy settings. For this reason, to build an ensemble using a voting methodology of bagging and boosting ensembles are used that gives better classification and accuracy. The volume and velocity of big data streams make this even more crucial regarding prediction accuracies and resource requirements.

The boosting algorithm provides sequential learning of the predictors. The first predictor is learned on the whole data set, while the following are learned on the training set based on the performance of the previous one. It starts by classifying original data set and giving equal weights to each observation. If classes are predicted incorrectly using the first learner, then it gives higher weight to the missed classified observation. Being an iterative process, it continues to add classifier learner until a limit is reached in the number of models or accuracy. Boosting has shown better predictive accuracy than bagging, but it also tends to over-fit the training data as
well. Most common example of boosting is AdaBoost and Gradient Boosting. Also, look at these articles to know more about boosting algorithms.

**Stacking classifier**

Stacking is an ensemble learning technique to combine multiple classification models via a higher order classifier. The individual classification models are trained based on the complete training set. Stacking works in two phases first multiple base classifiers to predict the class. Second, a new learner is used to combine their predictions with the aim of reducing the generalisation error.

![Figure 1.7 Stacking Method](image)

Stacking is concerned with combining multiple classifiers generated by using different learning algorithms \( d_1, \ldots, d_L \) on a single dataset \( x \), which consists of examples \( s_i=(x_i, y_i) \), i.e., pairs of feature vectors \( x_i \) and their classifications \( y_i \). Figure 1.7 shows the flow diagram of stacking based ensemble classification. Recently in the area of machine learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers. These classifiers could be based on a variety of classification methodologies and could achieve a different rate of correctly classified individuals.
Size of an Ensemble

While the number of component classifiers of an ensemble has a great impact on the accuracy of prediction, there is a limited number of studies addressing this problem. A priori determining of ensemble size and the volume and velocity of big data streams make this even more crucial for online ensemble classifiers. Most statistical tests were used for determining the proper number of components. More recently, a theoretical framework suggested that there are an ideal number of component classifiers for an ensemble which has more or less number of classifiers that would worsen the accuracy. It is called "the law of diminishing returns in ensemble construction." Their theoretical framework shows that using the same number of independent component classifiers gives the highest accuracy.

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a weighed vote of their predictions. The original ensemble method is Bayesian averaging but more recent algorithms include error-correcting, output coding, bagging and boosting. This research work reviews these methods and explains why ensembles can often perform better than any single classifier. Some previous studies comparing ensemble methods are reviewed and some new experiments are presented to uncover the reasons that AdaBoost does not overfit rapidly.

1.6.4 Clustering

The cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is the main task of exploratory in data mining and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression and computer graphics.
The cluster analysis itself is not one specific algorithm, but a general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can, therefore, be formulated as a multi-objective optimisation problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data pre-processing and model parameters until the result achieves the desired properties. Clustering parameters find and visually document groups of facts that were previously unknown. Clustering groups and aggregates the set of objects based on their similarity.

There are different ways a user can implement the cluster, which differentiate them between the clustering model. Fostering parameters within data mining can discover patterns in data that can lead to reasonable predictions about the future known as predictive analysis.

**Data Stream**

One of the early works on successful clustering for the data stream is done according to the above idea and is called stream algorithm. The stream algorithm divides the stream into a sequence of batches. For each of the batches, the algorithm uses an efficient clustering algorithm called local search to produce local clusters.

The local search is similar to k-means but performs better than k-means regarding memory and time. These cluster points created for each batch of data can be
fed again into the local search clustering algorithm and this will give the global
clustering across the whole stream. Here is the streaming algorithm, the data stream is
divided into chunks and for each chunk the algorithms create $k$ centres.

**Evolving Data Streams**

The previous clustering algorithm discussed uses the entire data stream for
calculating the clusters. These algorithms are one-pass algorithms over the data
stream. There are applications where such a view of the entire stream is not suitable.
For example, in some applications, the underlying clusters may change considerably
with time. So the real clusters at the point of analysis may be different from the
clusters that were seen at a previous time and taking that previous information to
create the current clusters may produce incorrect results. Users may also want to look
at the clusters that occurred in the previous month, previous year or previous ten
years. Catering to these requirements, a two-phase clustering algorithm is developed.
The algorithm consists of an online phase and an offline phase. In the online phase,
the algorithms create micro clusters at given time intervals and these clusters are
saved for the second phase. In the next phase, the algorithm uses the saved micro
clusters along with the summary statistics to create a macro view of the clusters. The
second phase can be carried out over any given period with input data from users.

The micro clusters are maintained over the whole history of the data stream. At
the initial step of the algorithm, a sufficient amount of incoming data is put to the
disk and a traditional K-Means clustering algorithm is run on this data to get the
initial micro clusters. After this initial step when the new data points arrive, the
micro clusters are updated to reflect these new data points. An existing cluster can
absorb a new data point, or a new data point can create a new cluster. Micro-
clusters are a method in stream clustering, which is used to record summary
information about the data objects in the streams.
The new data point belonging to an existing cluster is checked by its distance to each of the clusters from the new data point and it is calculated. If this distance is less than the distance of a cluster's maximum boundary and if the distance is the minimum between the cluster objects then the new data point can be assigned to a cluster. If the existing clusters cannot absorb the new point, the new data point must be an outlier, or it must belong to a new cluster.

These two cases cannot be distinguished without accessing further data. So a new cluster has to be created. To create a new cluster, an existing two clusters must be merged or an existing cluster must be deleted. The micro clusters are saved to disk periodically for the macro cluster algorithm to work. The micro clusters only contain very little information compared to the data used to create the clusters. Each micro cluster maintains the number of data points belonging to that cluster as a weight.

The macro cluster creation is a process invoked by the user and it is done for a period. This step is done offline. The micro clusters in the previous section were created using the whole data stream. Since macro clusters are created for a specific period the previous clusters must be subtracted from the clusters in the time window to make the clusters get created in that time window. After the micro clusters are obtained for a time window, these micro clusters are fed into a K-Means clustering algorithm to get the overall clusters during the period.

**Hoeffding Trees for classification**

Domingo’s and Hulten introduced the Hoeffding Trees for data stream classification. The main contribution of this research work was Hoeffding tree, which is a new decision tree learning method for streams. The algorithm induces a decision tree from the stream data incrementally spending very limited time on each new data item. The algorithm does not require the items to be stored after the items are
processed and a tree is updated. The algorithm only keeps the tree in the memory. The tree stores all the information required for its growth in the leaves and can classify items while it is in the building phase.

Each node of a decision tree contains a test on an attribute of the dataset. This test determines the path in which a data item should travel. A decision tree classifier sends a data item from its root to leaves based on the tests at each node along the path. The leaves are the classes of the classifier. When constructing decision trees usually the algorithms start from an empty node set and construct nodes based on the attributes that do the best split in the tree. There are heuristics like information gain Gini index for choosing the attribute that does the best split. In a batch algorithm, because all the training data is available the algorithm can calculate information gain or Gini index for each of the attribute to choose the best.

For a stream setting, because it cannot access all the data, the problem is to decide how many data points has to be seen before splitting based on an attribute. The Hoeffding tree gives an innovative method for making this decision in a stream setting. The Hoeffding bound is a statistical result used by the Hoeffding trees to achieve the results.

1.7 Applications of Data Mining

1.7.1 Database-oriented data sets

- The relational database, data warehouse, transactional databases.
- The object-relational databases, heterogeneous databases and legacy databases are some of the database-oriented data sets.

1.7.2 Advanced Data Sets and its applications

The list of advanced data sets and its applications are given below
- Data Streams and sensor data.
- Time-series data, temporal data, sequence data.
- Structure data, graphs, social networks and information networks.
- Spatial data and spatiotemporal data
- Multimedia database.
- Text databases.

Data mining tools and techniques can be used to take advantage of the historical data. Warehoused information can be filtered with the help of pattern recognition technologies, statistical and mathematical techniques in it. It can be used to find the relationships, trends, patterns, exceptions and anomalies that might otherwise go unnoticed. Data mining can be used in the business to discover relationships and pattern in the data to make better decisions. It can be used to predict customer loyalty and develop the smarter marketing promotions. Specific uses of data mining include:

### 1.7.3 Other Applications

- Market Detachment: It is used to identify the common characteristics of customers who buy the particular products regularly.
- Customer Churn: Used for predicting the customer mentality who likes to leave your company and go to a competitor.
- Fraud Detection: Identiﬁes the fraud based on the unusual behaviour.
- Direct Marketing: It is used to identify prospects which should be included to obtain the highest response rate. Figure 1.8 shows the uses of data mining.
- Interactive Marketing: It is used to predict each customer’s mentality based on accessing a Web site which is frequently accessed.
- Market Basket Analysis: It is used to account the kinds of products or services that are commonly purchased together.
Data mining techniques are used in many research areas, including mathematics, cybernetics, genetics and marketing.

Web mining, a type of data mining used in customer relationship management, integrates information gathered by traditional data mining methods and techniques over the web.

1.8 Existing Algorithms used in Data Mining

Data mining is the most advanced part of business intelligence. With statistical and other mathematical algorithms, automatically discover patterns and rules in the data that are hard to notice with online analytical processing and reporting. However, need to thoroughly understand how the data mining algorithms work to interpret the results correctly.

1.8.1 Decision Tree Algorithm

The most popular Data Mining algorithm, predicts discrete and continuous variables. It uses the discrete input variables to split the tree into nodes in such a way
that each node is purer regarding the target variable, i.e. each split leads to nodes where a single state of a target variable is represented better than other states.

Decision tree algorithm is a kind of data mining model to make induction learning algorithm based on examples. It is easy to extract display rule, which has smaller computation amount and could display important decision property and own higher classification precision. The Decision Tree (DT) algorithm has the following advantages.

- Interpreting data is easy due to the tree structure
- Understanding is easy because of tree-rules
- This tree extracts predictive information
- Decision tree can explain the logic using “If…. Then…” rules
- Decision tree is reliable and robust
- Decision tree is simple to implement.

### 1.8.2 The decision tree Growth algorithm

The input to this algorithm consists of the training records and the attribute sets. The algorithm works recursively selecting the best attribute to split the data and expanding the leaf nodes until the stop criteria is met.

### 1.8.3 Clustering algorithm

Given a set of items clustering algorithms compile the items more alike to item groups called clusters such that the items in a cluster are more close to the items in the same cluster than the items in other clusters. The closeness of the items is defined by various metrics and they depend on the item set and the application. Clustering is an important process in data analysis. Most of the clustering algorithms are unsupervised learning algorithms and can find the clusters without any training. Clustering algorithms can discover the hidden relationships among the items. There are various models developed for clustering data items. Some of these models are based on
Connectivity models, Centroid models, Graph-based models and Group models. For our discussion, the clustering problem can be formally defined as, given an integer \( k \) and a collection of \( n \) points in a metric space.

The \( k \) median finds the closest median point such that each point in the data collection is assigned. The quality of the clusters is defined by the sum of the squared distance from the assigned medians. This form of strict clustering is known as the K-Median problem and it is known to be NP (Nondeterministic Polynomial) -hard. One of the most famous and widely used clustering algorithms is k-Means clustering which is developed as a heuristic version of the k-Medians problem and is guaranteed to produce only a local optimum solution. The k-Means algorithms require \( \Omega(\frac{1}{\epsilon}) \) and random access to the data. Because of the time requirements and the random access requirements, k-means is not suitable for a stream setting.

1.8.4 K-Means Algorithm

The k-means algorithm is a simple iterative method to partition a given dataset into a user-specified number of clusters, \( k \). This algorithm has been discovered by several researchers across different disciplines. The algorithm iterates between two steps till convergence.

Step 1: Data Assignment. Each data point is assigned to its closest Centroid, with ties broken arbitrarily. This results in a partitioning of the data.

Step 2: Relocation of “means”. Each cluster representative is relocated to the centre (mean) of all data points assigned to it. If the data points come with a probability measure (weights), then the relocation is to the expectations (weighted mean) of the data partitions.

1.8.5 Naive Bays Algorithm

This calculates probabilities for each possible state of the input attribute for every single state of the predictor variable. Those probabilities predict the target attribute
based on the known input attributes of new cases. The Naïve Bayes algorithm is quite simple; it builds the models quickly. Therefore, it is very suitable as a starting point in your predictive analysis.

1.8.6 Streaming Machine Learning Algorithm

Streaming machine learning algorithm can be interpreted as performing machine learning in the streaming setting. In this case, the streaming setting is characterized by: High data volume and rate, such as transactions logs in Automatic Teller Machine (ATM), credit card operations, call log in Telecommunication Company and social media data, i.e. Twitter-tweet stream or Face book status update stream. Unbounded, which means these data always arrive the algorithm and will not be able to fit them in memory or disk for further analysis with the techniques. Therefore, this characteristic implied is limited to analyze the data once and there is little chance to revisit the data. Given these characteristics, conventional machine learning algorithms (which require all the data to be available in memory) are not suitable to handle. The requirements to handle streaming setting vary between methods. For classification algorithms, they need to bind these following four requirements.

- Process an example at a time and inspect it only once (at most)
- Use limited amount of memory
- Work in limited amount of time
- Be ready to predict at any point

Another aspect of streaming machine learning is change detection, i.e. since the input data is unbounded it needs to have the mechanism to handle and react to changes in incoming data characteristics. In classification, this aspect yields questions modifying the classifier to fit data characteristics changes and kinds of modification that should be performed. The streaming decision tree induction is called Hoeffding Tree. The name is derived from the Hoeffding bound that is used in the tree induction.
The main idea is, Hoeffding bound gives a certain level of confidence on the best attribute to split the tree, hence the model can be built based on the certain number of instances.

1.8.7 AdaBoost Algorithm

The AdaBoost algorithm works as follows. First, it assigns equal weights to all the training examples \((x_i, y_i) (i \in \{1, \ldots, m\})\). Denote the distribution of the weights at the \(t\)-th learning round as \(D_t\). From the training set and \(D_t\), the algorithm generates a weak or base learner \(h_t: X \rightarrow Y\) by calling the base learning algorithm. Then, it uses the training examples to test \(h_t\) and the weights of the incorrectly classified examples will be increased. Thus, an updated weight distribution \(D_{t+1}\) is obtained.

From the training set and \(D_{t+1}\), AdaBoost generates another weak learner by calling the base learning algorithm again. Such a process is repeated for \(T\) rounds and the final model is derived by weighed majority voting of the \(T\) weak learners, where the weights of the learners are determined during the training process. In practice, the base learning algorithm may be a learning algorithm which can use weight training examples directly; otherwise, the weights can be exploited by sampling the training examples according to the weight distribution \(D_t\). Another popular multi-class version of AdaBoost is AdaBoost.MH (Multi-class Hamming Trees) which works by decomposing multi-class task to a series of binary tasks. AdaBoost algorithms dealing with regression problems have also been studied. Since many variants of AdaBoost have been developed during the past decade, Boosting has become the most important “family” of ensemble methods.

1.8.8 K-Nearest Neighbor (Or KNN) Algorithm

In KNN, one of the simplest and rather trivial classifiers is the Rote classifier, which memorizes the entire training data and performs classification only if the attributes of the test object match one of the training examples exactly. There are
three key elements of this approach: a set of labelled objects, e.g., a set of stored records, distance or similarity metric to compute the distance between objects and the value of $k$, the number of nearest neighbours. To classify an unlabeled object, the distance of this object to the labelled objects is computed, its $k$-nearest neighbours are identified and the class labels of these nearest neighbours are then used to determine the class label of the object.

1.9 Need for the study

The present methods of mining data stream techniques have several drawbacks. In most of the present methods, the required information is being obtained by processing the stored data, in batches, but it fails in case of real time streaming data. But most of the organizations are required to forecast their future using real time data only. For example, a marketing company should identify the potential causes that impact the products and services of an organization have to be understood in a short time, to change its marketing strategy. The company needs to analyze lots of data, ranging from information about what customers are buying and reason for purchase.

Similarly, in banking sectors, active fund transactions have to be analyzed in real time by analysing the data stored in an operational data store. However, in other situations, those transactions have been executed instantaneously. Many pharmaceutical organizations are using big data analytics to discover new medicines. An insurance company may want to compare the patterns of traffic accidents across a broad geographic area with weather statistics, based on commercial third party data available in data warehouse or data mart. In these cases, the analysis has to be fast and practical. Also, organizations will analyze the data to see whether new patterns emerge.
Mining high speed data is an analytic computing platform that is focused on speed. This is because these applications require a continuous stream of often unstructured data to be processed. Therefore, data is continuously analyzed and transformed in memory before it is stored on a disk. Data streaming is the transfer of data at a steady high-speed rate sufficient to support such applications as High-Definition Television (HDTV) or the continuous backup copying to a storage medium of the data flow within a computer. Data streaming requires some combination of bandwidth sufficiency and, in a real-time human perception of the data, the ability to make sure that enough data is being continuously received without any noticeable time lag is important for efficient classification.

Data streaming is useful when analytics need to be done in real time while the data is in motion. In fact, the value of the outcome (and often the data) decreases with time. For example, if the outcome cannot be analyzed and acted upon immediately, a sales opportunity might be lost, or a threat might go undetected. Such high speed data streaming also requires some changes in the hardware technology. Many engineers utilize streaming for numerous other applications as well, in situations, where data cannot be generated or acquired fast enough, the engineers has to use a slower sample rate to transfer data by sampling at the necessary high-speeds for the short periods of time. But, this kind of sacrifice is not desirable.

The above discussed drawbacks of the current methods justifies that there is a need to search for new classification techniques for mining high-speed streaming data. This aspect of the research is very important because, now-a-days most companies have huge amounts of data and they have little or no idea about streamlining them. If a significant amount of data has to be quickly processed in real time, to gain better insights on the trend, data in motion or free streaming form of data is the best answer. In general, the mining data stream process consists of four essential stages, preliminary exploration of data, pattern identification from a training data set,
developing a model and finally model deployment. In the present work we focus on development of new classification method to mine high speed data streams. The current data classification methods have several drawbacks. The accuracy of the output is not sufficient enough and it should be enhanced further. The processing is more time consuming, it is not robust enough to analyze diverse type of data. It should also be noted that most of the classification algorithms work well with small data, but they fail miserably when applied on large and streaming data sizes. Thus it is important that the new attempts should also aim at increasing its scalability and also interpretability of the output. Herein we report new classification algorithm aimed at minimizing the above said drawbacks in big data environment.

1.10 Problem Statement

Recently, the continuous data stream technology evolved as a hot research topic, however, presents the data mining algorithms used do not fully support mining the big data sizes, which occupies much greater memory space than usual. Thus it is a big challenge to use current data mining technologies for real-time data mining. In this research work is aimed at minimizing the above problems in data mining. This is achieved using decision tree and classification algorithms. This is expected to improve the performance of the mining data stream algorithms, especially, time, memory and space utilization for generating accurate patterns. The developed application along with high speed data streams is being deployed in the proposed system. The important points in our development of new algorithms are listed below.

- Develop a suitable methodology for the application in mining high speed data stream in big data environment.
- Successful application of the proposed algorithms in mining high speed data stream
- Presentation of advantages of the method reported here as compared to earlier algorithms.
Identify possible limitation and scope of improvement of the proposed algorithms.

Scope for extension of the proposed algorithms to other mining areas of big data environment.

Thus, our proposed classification algorithms would accomplish above mentioned goals in big data environment.

1.1 Objectives of the Study

The main objective of this research work is to find out the relevant and accurate data from a big data set. This can be achieved through analysis of the live high-speed data streams. The first step is the classification of data into different classes. However, this, in turn, requires solving problems associated with time, memory, sample size, data rate and accuracy of a data pattern. All these processes should be done without compromising data security. Having motivated by the limitations of the existing system and algorithms, this dissertation using the latest algorithms aims at efficiently processing the high-speed data streams. To fulfill the above-mentioned research objectives, the following measures were carried out.

- Analysis of the continuous flow of data and
- Efficient classification of the data stream with the help of Dynamic Ensemble Generation and Classification (DEGC) algorithm.

1.12 Methodologies of the study

Work in this thesis proposes development of five novel classification algorithms, they are, Dynamic Random Decision Tree algorithm (DRDT), Decision Tree and Ensemble Learning (DTEL) algorithm, Efficient Hierarchical Clustering
(EHC) algorithm, Evaluation of new Incremental Classification Tree (ICT) algorithm, Dynamic Ensemble Generation and Classification (DEGC) algorithm.

The basic design of a typical data streaming algorithm is that the input data to be analyzed is fed into the algorithm which segregates the data based on input query and is sent to the storage disk for ready retrieval. The adhoc query raised by the user is further refined by additional queries. This process happens in memory. Experimentally, it is shown that this is a better approach in terms of error correction and yields better output than the previous algorithms.

The design function and merits of DRDT algorithm is discussed in chapter three. The DTEL algorithm is developed to mine high-speed data streams the results are presented in chapter four. Function and merits of Efficient Hierarchical Clustering (EHC) algorithm is discussed in chapter five. Evaluation of new ICT algorithm to mine high-speed data is presented in chapter six. Finally, the ability of DEGC algorithm to mine high-speed data streams is critically assessed and results are presented in chapter seven.

1.13 Limitations of the study

Though the novel algorithms reported in this thesis work showed better performance as compared to earlier algorithms, they all still have few limitations and there is scope to improve them further. All the algorithms reported here are one pass algorithms for streaming environment, which have deterministic bounds on the accuracy. As a result, datasets with very large average length of an item set could not be processed efficiently. Hence, additional knowledge about maximal recurrent item sets is required in such cases. The memory requirements of these algorithms are significantly lower than those of Frequent Pattern (FP) tree algorithm However, time-sensitive queries have not been considered. Due to the limitation of the system is the
extensive experimental result based on the actual network traffic is consistent with the results of the lower per-flow analysis. Due to the limitation of these applications, it is not feasible to load the arriving data into a traditional database management system that increases memory requirements and algorithm processing time.

1.14 Organization of the Thesis

Chapter 1: Introduction

This chapter gives a preface to the proposed research work on algorithms and needs that led to the research, problem statement and a brief description of the basic concepts underlying this research work.

Chapter 2: Review of Literature

This chapter depicts the review of the state of the artworks related to mining high speed data streams which are mainly based on classification methods. This chapter also presents the comparative analysis of various code clone detection and clone management.

Chapter 3: A Dynamic Random Decision Tree Algorithm

The chapter presents the detailed description of the various approaches towards the mining process of high-speed data using Dynamic Random Decision Tree (DRDT) classification. Also, it presents a detailed description of the different results is proposed.

Chapter 4: Decision Tree and Ensemble Learning Algorithm

The chapter presents the decision tree and ensemble classification of learning which has a great impact on the classification of high-speed data. With this notation, the DTEL (Decision Tree and Ensemble Learning) has been developed.
Chapter 5: An Efficient Hierarchical Clustering Algorithm

This chapter represents a detailed description of various approaches in an Efficient Hierarchical Clustering (EHC) algorithm and it is the improvement part of classification in the area of mining high-speed data streams. This chapter also presents a detailed description of the results being produced by different methods.

Chapter 6: Evaluation of a New Incremental Classification Tree Algorithm

Here the chapter presents the detailed description of the various approaches towards an evaluation of a new Incremental Classification Tree (ICT) algorithm for mining high-speed data streams. It also presents the results being produced by different methods proposed.

Chapter 7: Dynamic Ensemble Generation and Classification Algorithm

The detailed discussion about the various approaches towards an evaluation of a Dynamic Ensemble Generation and Classification (DEGC) of high-speed data streams using impact measures is presented in this chapter. It also presents a detailed discussion about the results being produced by different methods proposed.

Chapter 8: Conclusions and Scope for Further Study

The chapter highlights the conclusion of the research work and scope for further study and its improvements is specified.