Chapter – 1: Machine Learning Approach in Predicting Breast Cancer Recurrence

1.1 Introduction

The detection and treatment in the initial stages of breast cancer in women are key to the success of patient survival. The challenges of characterizing early breast cancer, a clinical oncologist need to concentrate on detection and its treatment. In the last 2-3 decades, there are remarkable technological innovations with respect to initial stage diagnosis and also detection of recurrence phenomena of breast cancer. There is a need for new research avenues such as knowledge discovery in databases (KDD) which incorporates data mining techniques as a popular research tool. This tool can be used to identify and exploit knowledge & relationships among a large number of expected variables to predict recurrence of breast cancer using the historical cases stored within datasets [Lavrac N, 1999; Richards et al., 2001].

1.2 Breast Cancer

Breast cancer is a malignant tumor that develops when cells in the breast tissue divide and grow without the normal control on cell death and cell division. Even though a definite cause of breast cancer still remains a mystery but to provide some perception to its possibility of occurrence in women some common risk factors are extracted. These factors include attributes such as age, hereditary risk and family history.

After a breast lump or mammographic abnormality is discovered and cancer is suspected, the only way to confirm the diagnosis is to take a biopsy. There are a number of procedures to obtain cells prior to treatment, such as fine-needle aspiration (FNA), nipple aspirates, ductal lavage, core needle biopsy, and local surgical biopsy [Gianni et al., 1997]. The biopsy is examined by a pathologist, and radiology is used to detect distant involvement in other organs by cancerous cells (metastasis) and includes chest x-ray, bone scan, computerized tomography (CT), Magnetic resonance imaging (MRI), and positron-emission tomography (PET) scanning.
If the tumor is localized, the main treatment is surgery. Other possible treatment includes radiation therapy, chemotherapy, immune therapy and endocrine therapy. To prevent the recurrence of breast cancer in the chest wall and improve overall survivability, a specific treatment, post-mastectomy radiotherapy (PMRT) is recommended after mastectomy i.e. removal of the breast and surrounding tissues [Lee et al., 2005, Overgaard et al., 1997]. Striking an equilibrium in the benefits and adverse effects, a data mining model with a focus on benefits to patient treatment has been proposed.

The prognostic indicator variables for more hostile and higher incidence of breast cancer recurrence are lymph node involvement, large tumors, low levels of estrogen and progesterone receptors and higher histologic grade. Studying these prognostic variables and finding those of most importance can give oncologist better perception regarding the prognosis of breast cancer recurrence. After the most important risk variables are discovered, they can be used for identifying high-risk patients.

The internationally reputed GLOBOCAN Institute has listed global statistics of breast cancer in women in the literature only up to the year 2012 with 1.7 million new cancer cases identified for the year 2012 [Ferlay et al., 2015]. It is reported that about 25% of new cancer cases in women and in particular breast cancer cases of about 12%. Breast cancer is the most common type of cancer in women both in developed and developing countries with 8,83,000 and 7,94,000 cases reported respectively.

The authoritative source of statistical evidence on cancer incidence and survival in the United States shows that the Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI) publishes cancer incidence and survival statistical data which covers approximately 28% of the US population [Howlader et al., 2015]. Also, the SEER coverage has published population-based data which includes 26% of African Americans, 38% of Hispanics, 44% of American Indians & Alaska Natives, 50% of Asians and 67% of Hawaiian/Pacific Islanders. As per the statistics from 1975 – 2013, it is expected that 2,46,660 women might be identified with breast cancer and 40,450 women will die of breast cancer in the year 2016, however, the latest bulletin from SEER is yet to be published. The statistical evidence based on 2009 – 2013 estimated that the median age of diagnosis for breast cancer is 61 years of age and the incidence rate is 125 per 100,000 women per year. The same statistical data source
shows that the median age of losing their life from breast cancer is 68 years of age with a death rate of 21 per 100,000 women per year. The overall five-year survival rate for 2006 - 2012 was as high as 89.7%; and approximately 12.4% of women were identified with breast cancer at some point in time during their lifetime based on 2011 - 2013 data. In other words, one in eight women will be identified with breast cancer during their lifetime.

In India, National Cancer Registry Programme has published a progressive number of patients identified with breast cancer is shown in Figure 1.1. It is observed that there is a drastic increase in breast cancer detection in the age group of 30 to 40 and also shows a short decline of cases in the age group of 50 to 60 & above.

Figure 1.1: Breast Cancer in India

Figure 1.2 shows the published data of different types of cancer detected in women for Bengaluru during the year 2012-2014 by National Cancer Registry Programme which indicates that breast cancer accounts for 27.5% from all types of cancers detected in women and age-adjusted incidence rates are reported in parenthesis.
1.2.1. Breast Cancer Recurrence

Breast cancer can recur at any time or not at all, but most recurrences happen mainly in the first 5 years after the primary treatment. The recurrence of breast cancer is indicated by the prognostic variables such as age, the involvement of lymph node, large tumors, low levels of estrogen and progesterone receptors, a high synthesis-phase fraction, higher histologic grade and cancer cells with a higher nuclear grade [Amir et al., 2007]. Local recurrence and metastasis or distant recurrence are the two types of breast cancer recurrence.

Local recurrence can recur at or near the same parts of the breast skin and surrounding tissues of the original or primary tumor, usually after a period of treatment. Local recurrence treatment after lumpectomy (breast-conserving surgery) most often can be treated successfully by surgery, usually a mastectomy. Even though the initial treatment was treated by mastectomy, breast cancer can still return to the chest area. The more lymph nodes with cancer at the time of the mastectomy, the higher the chances of breast cancer recurrence. Local recurrence after a mastectomy is usually treated with surgery followed by radiation therapy (if radiation therapy was not part of initial treatment). Treatment may also include chemotherapy and hormone therapy [Amir et al., 2007].

![Figure 1.2: Incidence of Breast Cancer in Bengaluru.](image-url)
Metastasis or distant recurrence of breast cancer most often spreads to the other parts of the body like bones, lungs, liver or brain, but still, it is considered and treated as breast cancer. When metastases are present, tests are done to determine which parts of the body are involved and also check the status of hormone receptor & HER2 status of the tumor. Metastasis is usually detected by bone scan, CAT scan, MRI scan, or liver blood tests [Amir et al., 2007]. Metastasis usually includes the treatment options like chemotherapy, radiation therapy, and hormonal therapies.

1.3 Public Data Source for Research

Wisconsin Breast Cancer dataset (WBCD) and Surveillance, Epidemiology and End Results (SEER) breast cancer dataset are the two most popular data mining dataset which can be used for research work. In the present work SEER breast cancer data statistics available during the years 1975 – 2013 has been used [Howlader et al., 2015]. These statistics provide information pertaining to cancer patients about patient demographics, primary tumor site, tumor morphology and stage at diagnosis, the first course of treatment, type of operation, complications, relapses, follow-up for vital status etc. The SEER Program is the only comprehensive source of population-based information which provides a stage of cancer at the time of diagnosis and complete patient survival data. The strength of the dataset is their high coverage, which is due to the close collaboration among clinical guideline groups and specific routines for reporting the same.

The data can be used for different purposes like epidemiological studies, monitoring & evaluating medical interventions and finding risk factors for specific types of cancer [Bull et al., 1989 and Berg et al., 2004]. Most of the research in oncology is focused on analyzing death events by obtaining statistics about survival. By analyzing death events, it is possible to discover relationships between tumor specifications and the severity of the disease. These datasets can also be used as a good source for information and knowledge extraction for research. Rules for predicting the prognosis of new patients or explaining specific outcomes can be extracted by data mining methods with extensive capability to support physicians in their decision-making [Windle et al., 2004].
1.4 Data Mining

The practice of learning from experience utilizing the outcomes is known as knowledge discovery in databases (KDD). It is the technique that determines new patterns and knowledge embedded in massive volumes of record sets [Fayyad et al., 1996]. In general, machine learning may be categorized as an investigational domain that aims to describe and develop algorithms which permit computers to learn and behave to resolve real-time problems based on statistical data. There are many machine learning techniques among them most preferred technique is data mining.

In medicine, automation is an important tool to analyze the enormous and complex volumes of data that are generated by healthcare activities. The data mining technique can automate to give valuable information to all stakeholders in health care, inclusive of patients through figuring out effective treatments and best practices.

Particular treatment for any patient is an end result of the doctor's learning, which is acquired during their medical education and from years of clinical practice. This achieved expertise can provide the relevant treatment or medication for any new patient. It is worth for the specialist to make use of their experiences and also the relevant data stored in electronic formats as a second opinion.

Data mining is a technology to discover knowledge from databases which incorporate a collection of instances or records [Rokach et al., 2007]. Each instance used by data mining algorithms is formatted using the identical set of attributes or variables. When the instances incorporate the class label then the learning process is known as the supervised learning or else is referred to as unsupervised learning. Clustering is a technique to depict data which can be normally adopted for unsupervised learning and rarely for supervised learning.

1.4.1 Classification

The classification and regression are predictive models normally adopted in supervised learning. Regression algorithms try to map input to domain values that can estimate certain items by considering items features. In the meantime, classifiers can map the input space into pre-characterized classes that can predict a new case of the patient whether benign (harmless) or malignant (harmful) [Rokach et al., 2010].
A Knowledge Based Approach for the Prediction of Recurrence of Breast Cancer

Classification is the technique of gaining knowledge from the target function ‘\( f \)’ that maps each attribute set ‘\( x \)’ to one of the predefined class labels ‘\( y \)’. The target function ‘\( f \)’ is also acknowledged informally as a classification model. Each instance is a record of data in the form of \((x, y)\) where ‘\( x \)’ is the features set and ‘\( y \)’ is the target variable (class label). Classification model is a tool used to depict data (Descriptive Model) or to predict the target variable for new instance (Predictive Model). Instances of classification prototypes are decision tree based, rule based methods, memory based reasoning, artificial neural network, naive bayes and nearest neighbors’ classifier [Tan et al., 2006].

The general procedure for explaining classification problem is presented in Figure 1.3. To build a classification prototype, the categorized data set is initially partitioned into two disjoint sets, indicated as training set and test set. Then a classification technique is applied on the training dataset to produce a classification model. Each classification technique applies a learning algorithm to build a model that has good generalization capability, which not only fits the training dataset well but also predicts correctly the class labels of many records which it can be seen on the test dataset [Tan et al., 2006].

![Figure 1.3: General approach for building a classification model](image)

Assume the objective is to classify some objects \( i = 1 \ldots n \) into ‘\( k \)’ predefined classes, where ‘\( k \)’ represents the number of classes. For instance, if the aim of classification is to investigate a patient whether breast will recur or non-recur, then the value of ‘\( k \)’ could be 2. Data set can be organized as \( n \times p \) grid ‘\( X \)’, where ‘\( x_{ij} \)’ constitute the feature value ‘\( j \)’ in the record ‘\( i \)’. Each row within the grid ‘\( X \)’ is represented by a vector ‘\( x_i \)’ with ‘\( p \)’ features & a class label ‘\( y_i \)’ and the classifier can be denoted as ‘\( x \)’. One technique to assess the classifier is by calculating the error estimation based on the confusion matrix.
1.4.2 Decision Tree Analysis

The decision tree analysis is used in various applications like finance, health & risk assessment, supply chain management, engineering fields, biotechnology and medical applications etc. The decision tree analysis can be applied for cancer studies to analyze and model based on the clinical data. The principles behind decision tree analysis are to illustrate its usefulness by statistical software program development for various applications in decision making.

1.4.2.1 Introduction to Decision Tree

A decision tree is a visual and analytical decision support tool in a hierarchical tree structure. Inductive machine learning algorithms is used to examine the decision characteristics stored in the data in the form \((X, Y) = (X_1, X_2, X_3 ... X_k, Y)\) that maps some sets of attributes \((X_1, X_2, X_3 ... X_k, Y)\) to the decision about target variable ‘\(Y\)’, after which the target variable ‘\(Y\)’ may be classified or anticipated as essential. The attributes can be any form of variables and based totally on the type of outcomes that we are interested in, a decision tree may be referred to as classification tree in a descriptive manner if the final outcome is discrete or regression tree in a predictive manner if there are continuous outcomes.

The concept of a decision tree has the following primary elements: a “root” node is the starting point of the tree, branches join nodes displaying the flow from question to answer. Nodes that have child nodes are called “interior” nodes. “Leaf” or “terminal” nodes are nodes that do not have child nodes and constitute a possible value of target variable given the variables represented by the path from the root. The subsequent section explains the mathematical algorithms of how to assemble a decision tree, including how an attribute and the value of the attribute are selected to split up a given node.

There are numerous benefits of decision tree over different classification principle, tools that make a decision tree popular except its easiness and interpretability. The technique is supervised learning that given a training data that consist of input and output, we can induce a decision tree in spite of little difficult data; it performs properly with massive data in a quick time, and other statistical or mathematical strategies can be easily incorporated in it.
1.4.2.2 Theory behind Decision Tree Analysis

The straightforward concept of decision tree analysis is to split the given source data set into subsets through recursive dividing of the parent node into child nodes primarily based on the homogeneity of within node instances or separation of inter-node instances with respect to target variables. For each node, attributes are examined and the splitter is selected to be the attribute such that after dividing the nodes into two child nodes according to the value of the attribute variable, the target variable is differentiated to the best algorithm. Because of this, we need to be able to distinguish between essential attributes and attributes which contribute little to the overall decision procedure. This process is repeated for each child node in a recursive manner until splitting is either non-feasible or all certain pre-specified stopping guidelines are satisfied.

Classification-Regression tree is a decision tree algorithm is a non-parametric probability distribution free technique to construct binary classification or regression trees. Splitting points attribute variables and values of chosen variables are chosen based on Gini impurity and Gini gain are given by:

\[
i(t) = 1 - \sum_{i=1}^{m} f(t, i)^2 = \sum_{i \neq 1} f(t, i) f(t, j)
\]

\[
\Delta i(s, t) = i(t) - P_L \cdot i(t_L) - P_R \cdot i(t_R),
\]

where \( f(t, i) \) is the probability of having ‘\( i \)’ in the node, and the target variable takes values in \( \{1, 2, 3 \ldots m\} \). \( P_L \) is the proportion of cases in node ‘\( t \)’ divided to the left child node and \( P_L \) is the proportion of cases in ‘\( t \)’ sent to the right child node. If the target variable is continuous, the split criterion is used with the Least Squares Deviation (LSD) as impurity measure. If there is no Gini gain or the preset stopping rule are satisfied, the splitting process stops.

Chi-Squared Automatic Interaction Detection (CHAID) classification approach brought with the aid of [Kass et al., 1980] for nominal predictors and extended by Magidson in 1993 to ordinal predictors is another powerful approach for nominal or ordinal target variable. Chi-Squared Automatic Interaction Detection exhausts all viable pairs of categories of the target variable and merges every pair till there is no statistically great distinction in the pair using Chi-square test.
Iterative Dichotomiser 3 (ID.3) developed by [Quinlan et al., 1986]. It is a classification tree used the concept of information entropy first added in a publication by Claude Shannon and Warren Weaver in 1949. This gives a way to measure the number of bits every attribute can offer and the attribute that yields the most information gain turns into an essential attribute and it should go on the top of the tree. This process is repeated until all instances in the node are in the identical categorized.

Suppose there are only two effective consequences “Yes” and “No” within the root node ‘T’ of the target variable. Let ‘p’ and ‘n’ denote the variety of “positive records and negative records, respectively. The initial information entropy is given by using:

$$I(p, n) = -\frac{p}{p + n} \log_2 \frac{p}{p + n} - \frac{n}{p + n} \log_2 \frac{n}{p + n}$$

If attribute ‘X’ with values \( \{x_1, x_2, x_3, \ldots x_N\} \) is chosen to be the split up predictor and partition the preliminary node into \( \{T_1, T_2, T_3, \ldots T_N\} \), and ‘p_i’ and ‘n_i’ denote the number of positive records and negative records in the child node ‘i’, then the expected information \( EI(X) \) and information gain \( G(X) \) are given by using:

$$EI(X) = \sum_{i=1}^{N} \frac{p_i + n_i}{p + n} \cdot I(p_i, n_i)$$

Information Gain, \( G(X) = I(p, n) - EI(X) \)

Ross Quinlan made several updates to ID.3 in 1993 and extended it to C4.5. Unlike ID.3 which offers with discrete attributes, C4.5 handles both continuous and discrete attributes by creating a threshold to split up the attribute into two groups, the ones above the threshold and those that are up to and including the threshold. Additionally, C4.5 deals with records that have unknown attribute values. The C4.5 algorithm uses normalized information gain or gain ratio as a changed splitting criterion of information gain that is the ratio of information gain divided by using the information because of the split up of a node on the basis of the cost of a specific attribute. The motive of this change is that the information gain tends to favor attributes that have a large range of values.
The best technique for deciding on the attribute for a selected node is to select the one that maximizes the given ratio. Stopping rule of C4.5 desires to be prespecified and it initiates a pruning process with the aid of changing branches that do not help with leaf nodes after they are created to decrease standard tree size and the estimated error of the tree. A rule set can be derived from the decision tree constructed by writing a rule for each path from the root node to the leaf node. [Quinlan et al, 2007] created C5.0 as an extended commercial version of C4.5 proving a number of enhancement such as smaller decision trees, misclassification types and weighing different attributes, reducing noise, time and memory efficiency, support for boosting which offers the trees more accuracy.

As a binary-split algorithm, like CART, Quick, Unbiased, Efficient, Statistical Tree (QUEST) proposed by [Loh et al., 1997]. It is a classification algorithm dealing with either categorical or continuous predictor ‘X’. Pearson’s chi-square test is applied to target variable~ ‘Y’ and predictor X’s independence if ‘X’ is a categorical predictor. Otherwise, if X is continuous, ANOVA F-test is performed to check if all the difference classes of Y have the identical mean of X. In both cases, values are calculated and compared to a Bonferroni-adjusted threshold to decide if, in addition, Levene’s F-statistics test desires to be achieved to determine if the predictor has to be chosen as the split up predictor for the node.

Overfitting happens in large tree prototypes in which the model fits noise in the data, such as including some attributes which might be inappropriate to the decision making process. If this type of model is applied to data apart from than the training set, the model might not carry out well. There are usually two ways to reduce overfitting: stop growing when data is split not extensive, or grow a complete tree, and then post-prune. For instance, if Gain of the best attribute at a node is under a threshold, stop and make this node a leaf rather than producing child nodes.

1.5 Apache Hadoop

Apache Hadoop is an open source software platform for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Hadoop services provide for data storage, data processing, data access, data governance, security, and operations. The benefits of using Hadoop is its’
ability to store, manage and analyze vast amounts of structured and unstructured data quickly, reliably, flexibly and at low cost.

- **Scalability and Performance** – distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyze data at petabyte scale.

- **Reliability** – large computing clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resilient – when a node fails processing is re-directed to the remaining nodes in the cluster and data is automatically re-replicated in preparation for future node failures.

- **Flexibility** – unlike traditional relational database management systems, you don’t have to create structured schemas before storing data. You can store data in any format, including semi-structured or unstructured formats, and then parse and apply the schema to the data when read.

- **Low Cost** – unlike proprietary software, Hadoop is open source and runs on low-cost commodity hardware.

### 1.5.1 Hadoop MapReduce

MapReduce is a processing technique and a program model for distributed computing. The MapReduce algorithm contains two important tasks, namely Map and Reduce. The map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key, value pairs). Second, the reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration
change. This simple scalability is what has attracted many programmers to use the MapReduce model. MapReduce involves the following operations and is depicted in Figure 1.4:

- Generally, MapReduce paradigm is based on sending the computer to where the data resides.

- MapReduce program executes in three stages, namely map stage, shuffle stage and reduce stage.
  
  - **Map stage**: The map or mapper’s job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

  - **Reduce stage**: This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.
The MapReduce framework operates on \( \langle \text{key, value} \rangle \) pairs, that is, the framework views the input to the job as a set of \( \langle \text{key, value} \rangle \) pairs and produces a set of \( \langle \text{key, value} \rangle \) pairs as the output of the job, conceivably of different types.

The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework.

Input and Output types of a MapReduce job: (Input) \( \langle k_1, v_1 \rangle \rightarrow \text{map} \rightarrow \langle k_2, v_2 \rangle \rightarrow \text{reduce} \rightarrow \langle k_3, v_3 \rangle \) (Output).

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map ( \langle k_1, v_1 \rangle )</td>
<td>list ( \langle k_2, v_2 \rangle )</td>
</tr>
<tr>
<td>Reduce ( \langle k_2, \text{list}(v_2) \rangle )</td>
<td>list ( \langle k_3, v_3 \rangle )</td>
</tr>
</tbody>
</table>

1.5.2 Hadoop Distributed File System

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault-tolerant and designed using low-cost hardware.

HDFS holds the very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in a redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.
Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of the cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

HDFS Architecture

The architecture of a Hadoop File System is depicted in Figure 1.5. It follows the master-slave architecture and it has the following elements.

Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks:

- Manages the file system namespace.
- Regulates client’s access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.
Datanode

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node in a cluster, there will be a datanode. These nodes manage the data storage of their system.

- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

Block

Generally, the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.
Goals of HDFS

- **Fault detection and recovery**: Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

- **Huge datasets**: HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

- **Hardware at data**: A requested task can be done efficiently when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

1.5.2.1 Block Placement in HDFS

It may not be possible to connect all the nodes in a flat topology in a large cluster. A common practice is to spread the nodes across multiple racks. Communication between two nodes on different racks has to intermediate through multiple switches as shown in Figure 1.6.

![Figure 1.6: Example of Cluster Topology](image)

HDFS estimates the network bandwidth between two nodes by their distance. The distance from a node to its parent node is assumed to be one. A shorter distance between two nodes means that the greater bandwidth they can utilize to transfer data. The placement of replica is critical to HDFS data reliability and read/write performance. A good replica placement policy should improve data reliability, availability and network bandwidth utilization. Currently, HDFS provides a configurable block placement
policy interface so that the users and researchers can experiment and test any policy that’s optimal for applications.

The default HDFS block placement policy tries to maintain a tradeoff between minimizing the write cost and maximizing data reliability, availability and aggregate read bandwidth. Upon the creation of a new block, the first replica is placed on the node where the writer is located, the second and the third replicas on two different nodes in a rack different from first’s replica’s rack. The rest of the replicas are placed with the restriction that no more than one replica is placed at one node and no more than two replicas are placed in the same rack when the number of replicas is less than twice the number of racks.

After the selection of all target nodes (DataNodes), the nodes are organized as a pipeline in order their proximity to the first replica. Data are pushed to nodes in this order of proximity. While reading a file, the NameNode first checks if the client’s host are located in the cluster itself. If yes, block locations are returned to the client in the order of its closeness to the reader. The block is read from DataNodes in this preference order.

This policy explained above reduces the inter-rack and inter-node write traffic and generally improves write performance. As the chances of a risk failure are far less than that of a node failure, this policy does not impact data reliability and availability guarantees.

The default HDFS replica placement policy can be summarized as follows:

1. No DataNode contains more than one replica of any block

2. No rack contains more than two replicas of the same block, provided there are sufficient racks in the cluster.

1.5.2.2 Balancer

In order to avoid placing new, more likely to be referenced data, HDFS Block placement policy does not take into account DataNode disk space utilization. Hence, this data might cause non-uniform data placement across the cluster. Also, imbalance might occur when new nodes are added to the cluster.
The Hadoop balancer is a tool that balances disk space usage on the cluster. It takes a threshold value as an input parameter, which is a fuzzy value in the range (0, 1). A cluster is balanced if, for each DataNode the utilization of the node differs from the utilizations of the whole cluster by no more than the threshold value. This tool can be used as an application program that can be run by the cluster administrator. It iteratively moves replicas from DataNodes with higher utilization to DataNodes with lower utilization. One key requirement for the balancer is to maintain data availability. When choosing a replica to move, and deciding its destination, the balancer guarantees that the decision does not reduce either the number of replicas or the number of racks.

The balancing tool optimizes the balancing process by minimizing the inter-rack data copying. As an example, if the balancer decides that a replica X needs to be shifted to a different rack and the destination rack happens to have a replica Y of the same block, then data will be copied from replica Y instead of replica X.

1.6 Comprehensive Review of the Current Literature

Several attempts have been made by the researchers to provide a more robust and accurate breast cancer recurrence system from the past two decades have been studied by Data mining and big data analytics approaches

1.6.1 Data mining Approach

[Delen et al., 2005] has done the comparative study using existing machine learning prototypes like C5.0 decision tree and ANN for breast cancer survivability. In this work, they reported that C5.0 and ANN gave a classification accuracy of 93.6% and 91.2% respectively. However, the pre-classification method was not precise enough to consider the cause of death and survivability rate was not considered.

[Bellaachia et al., 2006] have adopted Naive Bayes, Back-propagated Neural Network and C4.5 decision tree algorithms for the breast cancer recurrence survivability and reported that the best execution accuracy of 86.7% is given by C4.5 algorithm.

[Santi Wulan Purnami et al., 2008] have used 1-norm SVM for feature selection and Smooth SVM for classification of breast cancer. The diagnosis was done by using WBCD with a nine chosen attributes for feature selection.
[Lambrou et al., 2009] in their work have used rule-based Genetic Algorithms (GAs) for developing conformal prediction for the breast cancer survivability. The rules were generated by using WBCD training dataset of 683 records without missing values.

[Liu Ya-Qin et al., 2009] in their work have used C5.0 algorithm for breast cancer survivability on an imbalanced SEER data. Then to build the characterization execution for an imbalance dataset the bagging algorithm were used to yield an accuracy of 76.78%.

[Farzaneh Keivanfard et al., 2010] presented a forward selection method to extract the features to classify breast cancer on dynamic MRI. The extracted features were then applied to different ANN techniques like Multilayer Perceptron (MLP) neural network, Probabilistic Neural Network (PNN) and Generalized Regression Neural Network (GRNN) to classify breast cancer as benign and malignant. A good amount of accuracy was achieved using GRNN and PNN on a limited number of dataset records.

[Ankit Agrawal et al., 2011] have used data mining technique for a lung cancer SEER dataset to predict the survivability of the patients after a certain period of primary treatment. To choose a set of attributes an appropriate feature extraction techniques were used to model a recurrence of lung cancer.

[Souad Demigha et al., 2015] in their work have analyzed image mining technique for breast cancer screening. The work was carried particularly to show that data mining technique can be applied in medical field imaging. The main limitation of this work is that it requires longer training time with a minimal classification for training dataset.

[Smita Jhajharia et al., 2016] predictive model for the prognosis of breast cancer incorporates PCA along with ANN. It adopts Lavenberg-Marquardt optimization learning on a WBCD with 699 records and less number of attributes to classify cancer as benign or malignant with a high level of accuracy.

[Abeer Alzubaidi et al., 2016] proposed a hybrid framework using GA and MI to extract the features of breast cancer prediction. Further k-Nearest Neighbor (k-NN) and SVM are used as a classifier to diagnose WBCD dataset with a few number of attributes.
[Muthuselvan et al., 2016] proposed a prediction model of breast cancer by using a well-known classification rule mining algorithms and reported that J4.8 has a high classification accuracy of 86.36%.

1.6.2 Big data analytics approach

[Freund et al., 1997] proposed well known adaptive boosting algorithm. The algorithm combined with various machine learning algorithms will improve their performance in converting a strong classifier. However, this algorithm is sensitive to noisy data & outlier and when compared to another learning algorithm, it is less susceptible to the overfitting problem.

[Friedman et al., 2000] have proposed a LogitBoost algorithm using additive modeling. Generally, an additive model along with the cost function of logistic regression will yield a powerful execution in the presence of noisy data.

[Escudero et al., 2000] have proposed LAZYBOOST for accelerating ADABOOST by utilizing several feature selection on a fixed-size arbitrary subset of features from each boosting iteration and the base learner is prepared on this subset.

[Lazarevic et al., 2002] have proposed the parallelized boosting for tightly coupled shared memory framework and hence it is not appropriate in a distributed cloud environment. [Fan et al., 1999] proposed a boosting algorithm to achieve a parallelization in space but not with time. Hence, this algorithm works well with a large amount of data to accomplish space but not with processing execution time.

[Busa-Fekete et al., 2009] is a fast boosting algorithm which uses multiple-armed bandits (MAB). In this work, every arm communicates to a subset of a base classifier and one of them is selected in every iteration. The algorithm then utilizes a subset to optimize a classifier over the complete dataset. However, this algorithm does not work in a parallel environment to boost the execution through multiple machines.

[Bradley et al., 2007] proposed a boosting algorithm to minimize the logistic loss by modifying the AdaBoost algorithm. It delivers a boundless number of labeled samples in each iteration and creates a small subset of samples to prepare a base learner.
[Gambs et al., 2007] proposed an MULTBOOST algorithm that accomplishes parallelism in both space and time. This algorithm can fit in a parallel setting where two or more hubs participate in developing a boosting classifier.

[Wu et al., 2009] proposed a hybrid model MReC4.5 by combining C4.5 classifier along with MapReduce. The model incorporates steps of serialization operation to operate in a cloud computing environment.

[Panda et al., 2009] proposed a massively parallel learning system known as PLANET for learning classification and regression on enormous data sets utilizing MapReduce. This methodology is needed for weak learners but ensemble techniques such as boosting are not achieved.

[Malewicz et al., 2010] proposed a MapReduce like concept known as pregal which can process large scale graphs. An API is used exclusively to achieve distributed programming system. It adopts the property of MapReduce to achieve iterative computations over the graph. In this methodology, the sequence of iterations are adopted by super steps and from these a vertex can get the data from the past iteration.

[Rostom Mennour et al., 2015] have used well-known big data processing techniques such as MapReduce and Mahout. It has explored five machine learning algorithms designed for big data analysis to do the classification of ligands into dockable and undockable ones. After that, three algorithms have been selected for breast cancer receptor which is the best ones in terms of precision have made an ensemble classifiers. The obtained results have presented that it can reach about 80% precision.

[John Lewis et al., 2015] aims to realize the potential by identifying the most effective ways to utilize the growing amount of data generated by cancer care information systems through improved data linkage and application of big data and emerging business intelligence applications. This leads to a new set of research challenges. The first is essentially a knowledge management challenge about how rapidly growing sophisticated big data in healthcare research can be easily handled by oncologists. The second challenge is essentially about streaming of online data.

[Veeresh Patel et al., 2015] proposed a knowledge base model which is capable of determining various types of cancer, their genetic cause and drug reclamation. This
model creates a knowledge base in the form of a graph. After pre-processing, sampling and training of dataset, the learning model is capable of predicting the required information. This model uses a small number of genomic attributes for training and predicting.

### 1.6.3 Big Data Performance Enhancement Approach

[Dean et al., 2004] have proposed google MapReduce to handle large datasets and perform parallel programming which utilizes local disk bandwidth. The algorithm doesn’t optimize for analytical applications to provide a control on data storage and movement. Also, it lacks many functionalities of analytic application like join, Cartesian and neighbor operation.

[Yahoo Pig] and [Pike et al., 2005] proposed a concept for handling analytic application as database system by providing the environment of a high-level language. In fact, the users are shielded in using the system to a maximum extent to improve the performance. Moreover, it also lacks with few operations of the analytic application as it is built on top of Google MapReduce.

[Isard et al., 2007] proposed a powerful distributed data parallel concept that can be expressed as dependency graph which is similar to Google MapReduce. The algorithm performs computation as a dependency graph, where vertices represent the sequential programs and edges represents the computation between programs. But, they hard to use because they are built on top of DRYAD.

[Chih yang et al., 2007] extended the concept of MapReduce to merge datasets, which is the concept of Join operator. However, as MapReduce, it doesn’t give the full control on distributed data and also excludes the operations like Cartesian and neighbor operator.

[Abouzeid et al., 2009] proposed a hybrid architectural model known as HadoopDB which is a combination of MapReduce & DBMS technologies. It stores the information in a local database and lacks fault tolerance and dynamic scheduling.

[Dittrich et al., 2010] proposed Hadoop++ which makes a yellow elephant run like a cheetah. In spite of the fact that this methodology doesn’t require a change in Hadoop
technologies. As it adopts static concept, hence it obliges client to reorganize their input data. It can just co-locate 2 records that requested by the same job and requires restoring as a new record that injects into the framework.

[Afrati et al., 2010] proposed an optimizing joins in MapReduce that can enormously enhance Hadoop execution. The limitation of this work is that it cannot co-place the divided data parts and change in the physical design.

[Liu et al., 2008] proposed a Gridbatch which can support large scale data intensive application in cloud computing environment. It incorporates the concept of dividing and co-location of the records at the framework level would decouple them so that diverse programs can utilize distinctive techniques to characterize related documents.

In more propelled partitioning elements of parallel database frameworks [Terasort], for instance, TeraData, IBM DB2, Aster Data, tables are co-partitioned and the query optimizer lacks the reality to produce effective query plans. This methodology adjusts to the MapReduce framework by retaining Hadoop's dynamicity and adaptability.