iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: V Month of publication: May 2015
DOI:

www.ijraset.com
Call: 08813907089 | E-mail ID: ijraset@gmail.com

www.ijraset.com
IC Value: 13.98

Volume 3 Issue V, May 2015
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Implementation of K-Means Clustering Algorithm in Hadoop Framework

Uday Kumar Sr[1], Naveen D Chandavarkar[2]
[1]PG Scholar, Assistant professor, Dept. of CSE, NMAMIT, Nitte, India

Abstract*— Drastic growth of digital data is an emerging area of concern which has led to concentration of Data Mining technique. The actual data mining task involves programmatic or semi-programmatic analysis of large quantities of data to extract hidden interesting patterns such as groups of data records, which is referred as Cluster Analysis. Clustering is the partitioning of data items into different groups (clusters), so that the data objects of each cluster share common characteristics. Data collected in practical scenarios is more often than not completely random and unstructured or semi-structured. Hence, there is always a need for analysis of such data sets to derive meaningful hidden information. In this kind of scenarios, the unsupervised algorithms come in to picture to process unstructured or even semi structured data sets by resultant. Several clustering algorithms have been proposed in the past few years among which k-means clustering algorithm is one of the simplest and popular unsupervised learning algorithm that will solve the well-known clustering problem. K-means clustering algorithm produces a specific number of disjoint clusters. The k-means algorithm requires k initial cluster centers that must be specified beforehand and are randomly selected. This paper discusses the implementation of K-means algorithm in MapReduce programing model which is run on Hadoop distributive environment.

***Keywords*— Data Mining, Clustering, K-Means Clustering, Distributed Computing, MapReduce, Hadoop**

I. INTRODUCTION

Data mining has been defined as the use of algorithms to extract information and patterns as part of Knowledge Discovery in Databases (KDD). The three important research areas of data mining are Classification, Clustering and Association rule mining. Brief reviews of these areas are discussed as follows. Classification assigns items to appropriate classes by using the attributes of each item. Clustering methods group items, but unlike classification, the groups are not predefined. A distance measure, such as the Euclidean distance between feature vectors of the items, is used to produce the clusters. Association rule mining (ARM) considers shopping-cart or market basket data items, i.e., the items purchased on a particular visit to the supermarket. ARM firstly finds out the frequent sets out of the data, which must have to meet a certain support level. Cluster analysis has been widely used in numerous applications, including data analysis, pattern recognition, market research, and image processing. In many businesses which involves sales and transactions, clustering can help marketing specialists discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns. In biological field, it can be used to derive animal and plant taxonomies, categorize genes with similar functionality, and obtain an insight into structures inherent in populations. There are plenty of clustering algorithms available for use, among which K-Means clustering algorithm is one of the simple and popular algorithm. K-means is one of the simplest and popular unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a very simple and easy way to cluster a given data set through a certain number of clusters (assume k clusters) specified in prior. The main idea is to define k centroids, one for each cluster. In general, distributed computing is any technique of computing that involves multiple computers remote from each other that each has a role in a computation problem or information processing. Apache™ Hadoop is one such open source framework that supports distributed computing. It came into existence from Google's MapReduce and Google File Systems projects. It was actually created by "Doug Cutting", the creator of Apache Lucene project which is actually the widely used text search library. Hadoop finds its origin in Apache Nutch project, an open source web search engine, itself a part of the Lucene project. It is a framework that can be used for distributed computing and large-scale data processing, although it is best known for MapReduce and its distributed filesystem (HDFS). The Hadoop framework takes into account the node failures that can occur in its cluster and is automatically handled by it. This makes Hadoop really flexible and a fault tolerant platform for data intensive applications. This not only reduces the time required for completion of the operation or processing of voluminous data sets but also reduces the individual system requirements for computation of large volumes of data. Since the start of the Google File Systems(GFS) and MapReduce concepts, Hadoop has taken the world of distributed computing to a greater extent of successful track with various versions of Hadoop that are now being in existence and fewer under Research and

©IJRASET 2015: All Rights are Reserved

www.ijraset.com Volume 3 Issue V, May 2015
IC Value: 13.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Development. Few of which include Pig, Hive, Zookeeper, HBase, Sqoop, Oozie. The MapReduce structure gives great flexibility and speed to execute a process over a distributed Framework. Unstructured data analysis is one of the most challenging aspects of data mining that involve implementation of complex algorithms. The Hadoop Framework is designed to give the solution for the storage and computation of voluminous data sets. This can be done by downscaling and consequent integration of data and reducing the configuration demands of systems participating in processing such huge volumes of data. The workload is shared by all the nodes of computers connected on the network (Cluster) and hence increases the efficiency and overall performance of the network and at the same time facilitating the fast processing of voluminous data.

II. K-MEANS CLUSTERING

Numbers of clustering algorithms have been designed, among which k-means, the most is widely used in data mining applications because of its simplicity and fast processing ability of huge data sets. K-means clustering is an algorithm to classify or to group the objects based on attributes into K (Positive integer) number of groups. The clustering is done by minimizing the sum of squares of distances between the data points and the corresponding cluster centroid. The algorithm accepts the count of clusters and the initial set of cluster centroids as input parameters. It computes the distance between each data point and the initial set of cluster centroid. The point which has least distance from a centroid is then added to that cluster. After the assigning all data points to the respective clusters, the centroids of the clusters are recalculated. For distance measure the Euclidean distance method is normally used. Consider there are two points defined as P=(a1(p), a2(p), a3(p)….) and Q=(a1(q), a2(q), a3(q)….). The Euclidean distance between two multi-dimensional data points is defined as:

$$D(P,Q) = \sqrt{(a1(p) - a1(q))^2 + (a2(p) - a2(q))^2 + \ldots}$$
$$= \sqrt{\sum_{j=1}^{p} (aj(p) - aj(q))^2} \quad (1)$$

The main part of this algorithm is to obtain a minimal squared difference between the centroid of the cluster and the data point in the dataset.

$$|X(j)\, i - Cj|2 \quad (2)$$

Where Xi is the value of the data in the dataset and Cj is the value of the centroid of the cluster.

The K-Means algorithm can be implemented as follows:

A. Designate k randomly selected points from n points as the centroids of the clusters;
B. Assign a point to the cluster, whose centroid is closest to it, based on Euclidean or some other distance measure;
C. Recompute the centroids for all clusters based on the items assigned to them;
D. Repeat steps (2 through 3) with the new centroids until there is no change in point membership.

One measure of the quality of clustering is the sum of squared distances (SSD) of points in each cluster with respect to its centroid. The algorithm may be applied several times and the results of the iteration with the smallest SSD selected. The k-means algorithm is relatively efficient and fast. It computes result at O (tkn), where n is the number of objects or points, k is the number of clusters and t is the number of iterations. The computational complexity of the k-means algorithm mainly depends on the number of clusters, number of data elements and the number of iterations.

III.MAPREDUCE PROGRAMING MODEL

MapReduce is a programing model for data processing. It is one of the core components of the Hadoop framework. Hadoop can run MapReduce programs written in various languages i.e Java, Ruby, Python and C++. Most importantly, MapReduce programs are inherently parallel. MapReduce works by breaking the processing into two phases: the map phase and the reduce phase. Each phase takes key-value pairs as input and output, the data types of each key-value pairs may be chosen by the programmer. The programmer must also design a Map function that uses a (key,value) pair for computation. The Map function results in the creation

©IJRASET 2015: All Rights are Reserved

of another set of data in form of (key,value) pair which is known as the intermediate data set. The programmer also designs a Reduce function that combines value elements of the (key,value) paired intermediate data set having the same intermediate key. MapReduce implementation splits the huge data into several smaller chunks that are independently fed into the nodes in the cluster so the number and size of each individual chunk of data is dependent on the number of nodes connected to the network (Cluster). There are major three components in MapReduce e.g. Job Tracker, Task Tracker and Job history Servers. Job tracker is actually the master of the system, it actually manages the jobs that need to be done on the resources in a way that which resource (task tracker) would be assigned what job. Task trackers are the slaves whose major responsibility is to run the jobs of map as well as reduce upon the assignment of job tracker. Job history servers are contains the history logs of all the job trackers; it is just for the backup purposes. The issues like division of data to various nodes , task scheduling, node failures, task failure management, communication of nodes, monitoring the task progress is all taken care by the master node. The data used as input and output data is stored in the HDFS file-system [10].

IV. K-MEANS CLUSTERING USING MAPREDUCE

The primary step in implementing the K-Means clustering algorithm in MapReduce is to state and handle the input and output of the application. Since the MapReduce programing model requires key/value pairs to be submitted to the MapReduce job, the input to K-Means algorithm must be prepared as key/value pairs. Here, the key is selected as cluster center and value be the vector of data set. To implement Map and Reduce routines two files are very essential, one is that stores the initial cluster centroids and the other that stores the vector form of data set to be clustered. Having these two input files ready in our hand, clustering of data can be done by following the detailed steps described in the chapter 1. The Map and Reduce routines are designed using the steps described in the K-Means algorithm which is described as follows. Before getting into the implementation both of the input file must be copied into HDFS from the local filesystem because the hadoop reads input from its own filesystem called HDFS rather than any local filesystem. The Map function takes these two files as the input, the initial cluster center file in HDFS form the key field and the other file consisting of vector form of data set forms the value. As followed from the algorithmic steps the distance calculation from cluster center to each point in the data set is performed in the Map function, the suitable code is written for the distance calculation, simultaneously recording the cluster to which the given vector is nearest. After processing all the vectors, the vectors are assigned to the nearest cluster. As soon as the Mapper finishes its task the computation is recalculated until it reaches a convergent point. This recalculation part goes into the Reduce routine, it also restructures the clusters to avoid the clusters with extreme size that is the clusters with too fewer data vectors or too many data vectors. This newly created clusters are written back to the disk which will be loaded as input to the next iteration.

V. EXPERIMENTAL RESULTS

Here, the 20_newsgroups data set is being used for clustering. This data set consists of 20000 messages taken from 20 newsgroups. Each newsgroup is stored in a subdirectory, with each article stored as a separate file. The data is organized into 20 different newsgroups, each corresponding to a different topic. Firstly, the 20_newsgroups data set is converted into bag of words vectors form and stored in a file called “vectors”. Here, to generate bag of words vectors from these data set first we parse every document of each news group and create a dictionary of words which contains index of each word read. Later again all the documents are parsed, for each word it parse, it records the occurrence and computes the fraction of occurrences. From this bag of words vector form of data set select randomly the k (20) initial centroids and store in a separate file called “clusters”. These two files are fed to the MapReduce implementation of K-means clustering algorithm which is discussed in the previous section. In this implementation the numbers of iterations are restricted to 3; hence it generates 3 cluster files after the processing of vectors and initial clusters. The fig.1 show the comparison of two different implementation results of K-Means clustering algorithm in which the MapReduce implementation gives the improved results over the stand-alone implementation in terms of execution time. By properly modifying the configuration parameters in hadoop framework it still enhances the performance of the MapReduce process. The fig.2 shows the comparison of MapReduce implementation and improved MapReduce implementation which shows the improvement in execution time. The hadoop configuration parameters considered fir our experiment are number of map slots, mapred.io.sort.record.percent, mapred.io.sort.spill.percent. The fig.3 summarizes all the results obtained by which it can be concluded that MapReduce implementation of an algorithm gives good performance results. By setting the hadoop configuration parameters properly it will still enhance the overall performance.

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue V, May 2015
ISSN: 2321-9653

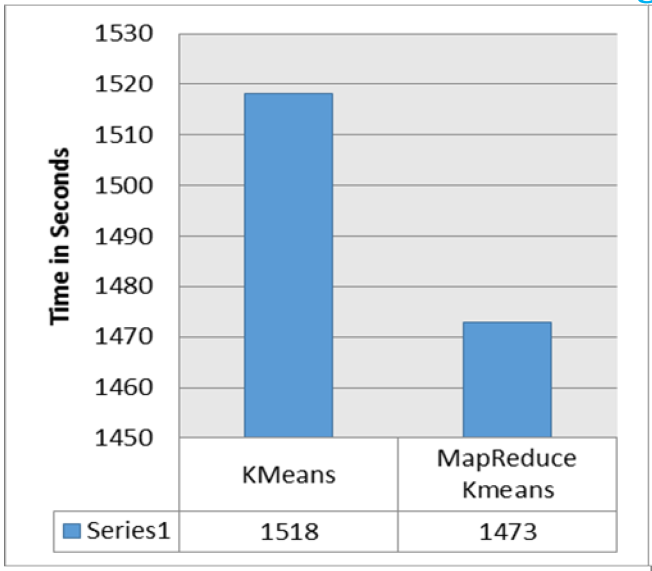


Fig.1. Comparison of execution time between Kmeans and MapReduce Kmeans

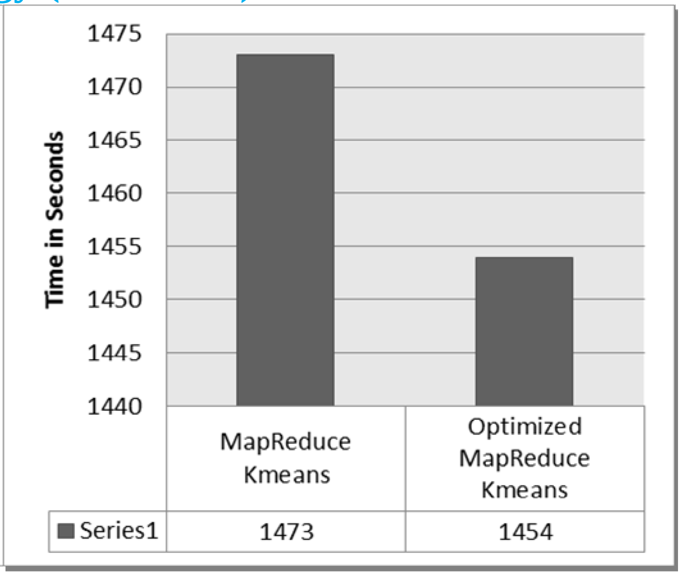


Fig. 2 Comparison of execution time between MapReduce Kmeans and Optimized MapReduce Kmeans

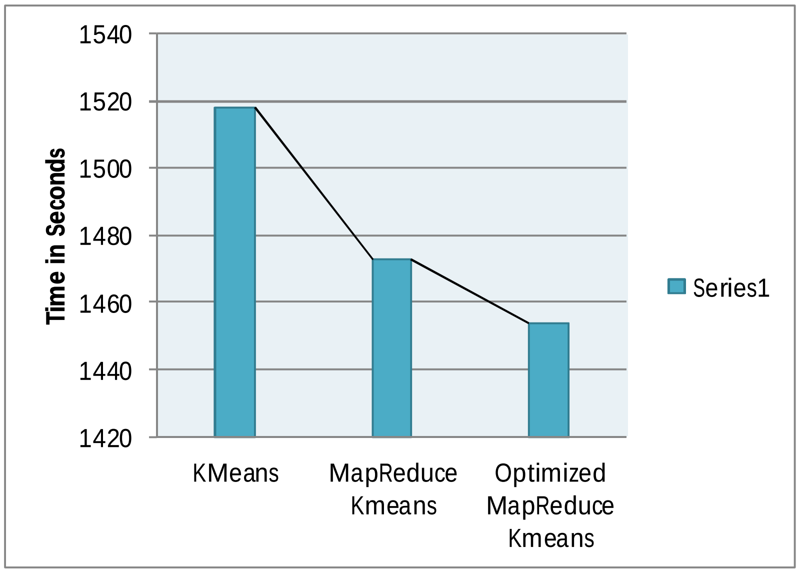


Fig.3 Overall Comparison of execution time between Kmeans, MapReduce Kmeans and Optimized MapReduce Kmeans

VI. CONCLUSIONS

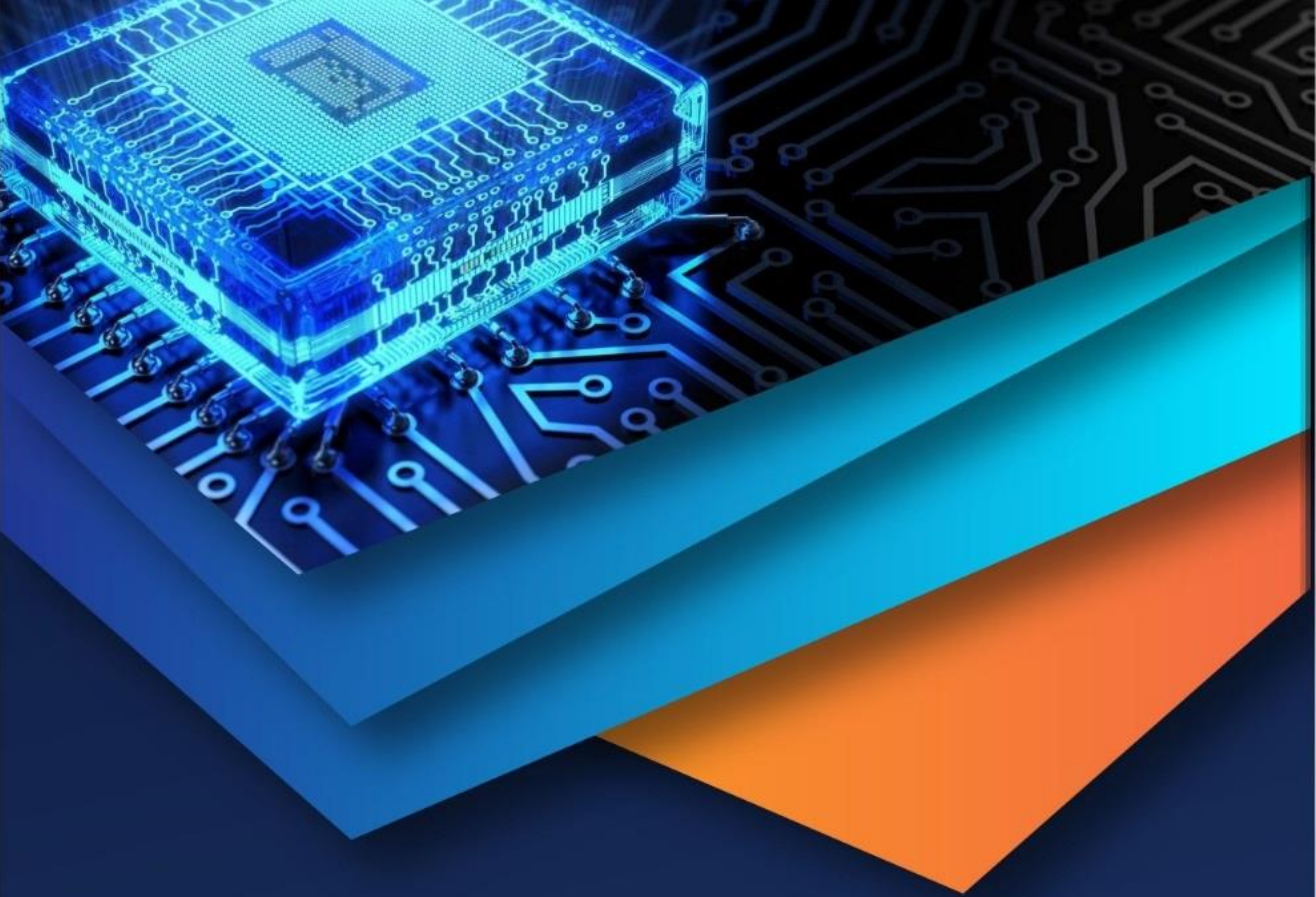
In this current world the data size is growing exponentially from various sources. It is very essential to process these huge data to extract useful information hidden in it. Clustering is one such research attention to extract useful information from voluminous data. Among several clustering algorithm K-Means algorithm is simple and popular. Hadoop is an apache open source product which gives us the distributive environment for processing of data. This paper discussed the implementation of K-Means Clustering Algorithm over a distributed environment in MapReduce programing model and improving the MapReduce process execution time.

©IJRASET 2015: All Rights are Reserved

REFERENCES

[1] Apache Hadoop. http://hadoop.apache.org/
[2] T. White, Hadoop: The Definitive Guide. O'Reilly Media, Yahoo! Press, June 5, 2009.
[3] J. Venner, Pro Hadoop. Apress, June 22, 2009.
[4] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu,P. Wyckoff, R. Murthy, "Hive – A Warehousing Solution Over a Map-Reduce Framework," In Proc. of Very Large Data Bases, vol. 2 no. 2,August 2009, pp. 1626-1629.
[5] S. Ghemawat, H. Gobioff, S. Leung. "The Google file system," In Proc.of ACM Symposium on Operating Systems Principles, Lake George, NY, Oct 2003, pp 29–43.
[6] F. P. Junqueira, B. C. Reed. "The life and times of a zookeeper," In Proc. of the 28th ACM Symposium on Principles of Distributed Computing, Calgary, AB, Canada, August 10–12, 2009.
[7] A. Gates, O. Natkovich, S. Chopra, P. Kamath, S. Narayanam, C.Olston, B. Reed, S. Srinivasan, U. Srivastava. "Building a High-Level Dataflow System on top of MapReduce: The Pig Experience," In Proc. of Very Large Data Bases, vol 2 no. 2, 2009, pp. 1414–1425
[8] Description if Single Node Cluster Setup at: http://www.michael noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-nodecluster/ visited on 21st January, 2012
[9] Description of Multi Node Cluster Setup at: http://www.michael noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/ visited on 21st January, 2012
[10] J. Dean and S. Ghemawat. Mapreduce: Simplied data processing on large clusters.Communications of the ACM, 51(1):107-113, 2008.
[11] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J. Qiu, and G. Fox. Twister: A runtime for iterative mapreduce. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pages 810-818. ACM, 2010.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)