

## High Performance Scalable Big Data and Machine Learning using Apache Mahout

L.A ABDUL RASUL AL WAILI

College of Education – University Of Wasit

### **Abstract**

Big Data Analytics and Machine learning refers to the intelligent and dynamic response by the software or embedded hardware programs depending upon the input data. Machine learning is the specialized domain that operates in association with the artificial intelligence to have strong predictions and analysis. Using this approach, there is no need to explicitly program the computers for specific applications rather the computing modules evaluates the dataset with its inherent behavior so that real time fuzzy based analysis can be done. The programs developed with machine learning paradigms focuses on the dynamic input and dataset so that the custom and related output can be presented to the end user. This manuscript underlines the high performance big data based execution and machine learning using effectual approach of Apache Mahout.

*Keywords: Big Data, High Performance Computing, Machine Learning*

### **Introduction**

A number of application domains exist where big data processing [1] and machine learning [2] approaches are widely used including fingerprint analysis, multidimensional biometric evaluation, image forensic, pattern recognition, criminal investigation, bioinformatics, Biomedical informatics, Computer vision, Customer relationship management, Data mining, Email filtering, Natural language processing, Automatic

summarization, Automatic taxonomy construction, Robotics, Dialog system, Grammar checker, Language recognition, Handwriting recognition, Optical character recognition, Speech recognition, Machine translation, Question answering, Speech synthesis, Text simplification, Pattern recognition, Facial recognition system, Handwriting recognition, Image recognition, Search engine analytics, Recommendation system and many others.

A number of approaches are implemented to machine learning but in traditional integrations the Supervised and Unsupervised Learning [3] is widely used. In supervised learning, the program is trained with a specific type of dataset with the target value. After learning and deep evaluation of the input data and corresponding target, it starts giving prediction. The common examples of supervised learning algorithms include artificial neural networks, support vector machines and the classifiers. In case of unsupervised learning, the target is not assigned with the input data. In this approach, the dynamic evaluation of data is done with the high performance algorithms including k-means, self-organizing maps (SOM) and clustering techniques. Other prominent approaches and algorithms associated with Machine Learning includes Dimensionality reduction, Decision tree algorithm, Ensemble learning, Regularization algorithm, Supervised learning, Artificial neural network, Deep learning, Instance-based algorithm, Regression analysis, Classifiers, Bayesian statistics, Linear classifier, Unsupervised learning, Artificial neural network, Association rule learning, Hierarchical clustering, deep cluster evaluation, Anomaly detection, Semi-supervised learning, Reinforcement learning and many others.

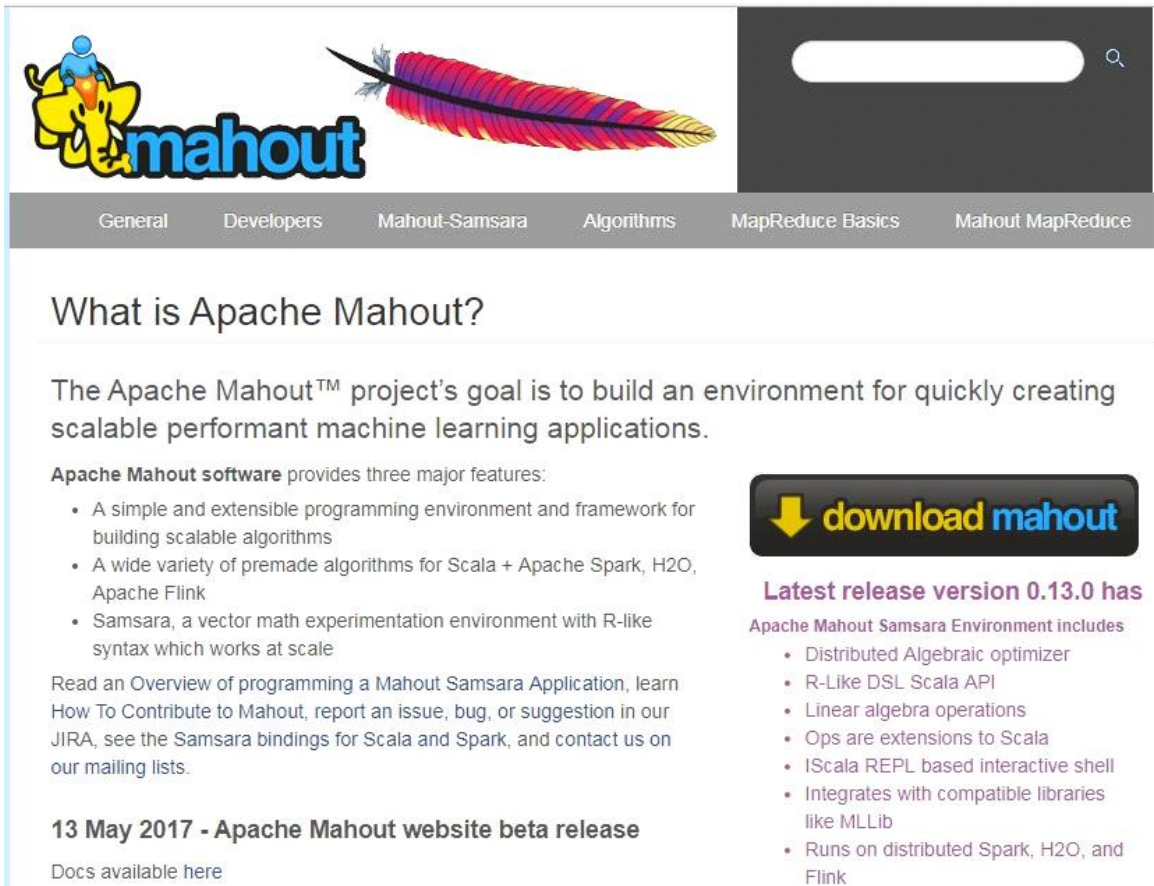
### **Free and Open Source Tools for Machine Learning**

- Apache Mahout
- Scikit-Learn
- OpenAI

- TensorFlow
- Char-RNN
- PaddlePaddle
- CNTX
- Apache Singa
- DeepLearning4J
- H2O
- GNU Octave
- R
- Orange
- WEKA
- Torch
- Yooreeka
- Shogun
- Massive Online Analysis (MOA)
- Mallet
- ELKI

**Apache Mahout: The Scalable High Performance Machine Learning Framework**

URL: [mahout.apache.org](http://mahout.apache.org)



**Figure 1: Official Portal of Apache Mahout**

Apache Mahout [4] is the powerful and high performance machine learning framework for the implementation of machine learning algorithms. Apache Mahout is traditionally used for the integration of supervised machine learning algorithms with the target value assigned to each input data set. Apache Mahout can be used for assorted research based applications including Social Media Extraction and Sentiment Mining, User Belief Analytics, YouTube Analytics and many related real time applications.

In Apache Mahout, a Mahout refers to the object which drives or operates the elephant. The mahout act as the master of elephant in association with Apache Hadoop and it is presented in the logo of elephant. Apache Mahout runs with the base installation of Apache Hadoop and then the machine learning algorithms are implemented with the features to develop and deploy the scalable machine learning

algorithms. The prime approaches like recommender engines, classification problems and clustering can be effectively solved using mahout.

Corporate Users of Mahout includes the following

- Adobe
- Facebook
- LinkedIn
- FourSquare
- Twitter
- Yahoo

### **Installation of Apache Mahout**

To start with the Mahout installation, first of all Apache Hadoop is required to be setup on the Linux Distribution. To get ready with Hadoop, the installation is required to be updated as follows in the Ubuntu Linux.

```
$ sudo apt-get update
$ sudo addgroup hadoop
$ sudo adduser --ingroup hadoop hadoopuser1
$ sudo adduser hadoopuser1 sudo
$ sudo apt-get install ssh
$ su hadoopuser1
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
$ ssh localhost
```

### **Installing the Latest Version of Hadoop**

```
$ wget http://www-us.apache.org/dist/hadoop/common/hadoop-HadoopVersion/hadoop-HadoopVersion.tar.gz
$ tar xvzf hadoop-HadoopVersion.tar.gz
$ sudo mkdir -p /usr/local/hadoop
$ cd hadoop-HadoopVersion/
$ sudo mv * /usr/local/hadoop
$ sudo chown -R hadoopuser1:hadoop /usr/local/hadoop
```

The following files are required to be updated next

- ~/.bashrc
- core-site.xml
- hadoop-env.sh
- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml

```
$ hadoop namenode -format
$ cd /usr/local/hadoop/sbin
$ start-all.sh
```

### Web Interfaces of Hadoop

MapReduce: <http://localhost:8042/>  
NameNode daemon: <http://localhost:50070/>  
Resource Manager: <http://localhost:8088/>  
SecondaryNameNode:: <http://localhost:50090/status.html>

The default port to access Hadoop is 50070 and using <http://localhost:50070/> on Web Browser

After installation of Hadoop, the setup of Mahout is required as follows.

```
$ wget http://mirror.nexcess.net/apache/mahout/0.9/mahout-Distribution.tar.gz  
$ tar zxvf mahout-Distribution.tar.gz
```

### **Implementation of Recommender Engine Algorithm**

Now days, we shop on the online shopping platforms like Amazon, E-Bay, SnapDeal, FlipKart and many others. We generally see that most of these online shopping platforms give us suggestions or recommendations [5, 6] about the products which we like or earlier purchased. This type of implementation or suggestive modeling is known as recommender engine or recommendation system. Even in YouTube, we see the number of suggestions regarding related videos which we view. Such online platforms integrate the approaches of recommendation engines by which the related best fit or most viewed items are presented to the user as recommendations.

Apache Mahout provides the platform to program and implement the recommender systems. For example, the Twitter HashTag Popularity can be evaluated and ranking can be done based on the visitor count or popularity or simply hits by the users. In YouTube, the number of viewers is the key value which determines the actual popularity of that particular video. Using Apache Mahout, such algorithms can be implemented which are covered under high performance real time machine learning.

For example, a data table which presents the popularity of products after online shopping by the users is recorded by the companies so that the overall analysis of popularity of products can be done. The rating from 0–5 is logged from the users so that the overall prominence of the product can be evaluated. This dataset can be evaluated using Apache Mahout in Eclipse IDE.

For integration of Java Code with Apache Mahout Libraries on Eclipse IDE, there are specific JAR files which are required to be added from Simple Logging Facade for Java (SLF4J).


Following is the Java Code Module with the methods which can be executed using Eclipse IDE with the JAR files of Mahout to implement Recommender Algorithm

```
DataModel dm = new FileDataModel(new File("inputdata"));
UserSimilarity us = new PearsonCorrelationSimilarity(dm);
UserNeighborhood un = new ThresholdUserNeighborhood(ThresholdValue), us, dm);
UserBasedRecommender r=new GenericUserBasedRecommender(dm, un, us);
List<RecommendedItem> rs=recommender.recommend(UserID, Recommendations);
for (RecommendedItem rc : rs) {
    System.out.println(rc);
}
```



Figure 2: Simple Logging Facade for Java





SLF4J Project  
[Introduction](#)  
[Download](#)  
[Documentation](#)  
[License](#)  
[News](#)  
Support  
[Mailing Lists](#)  
[Bug Reporting](#)  
[Source Repository](#)  
[Support offerings](#)  
Native implementations  
[Logback](#)  
[Wrapped implementations](#)  
[JDK14](#)  
[Log4j](#)  
[Simple](#)

## Latest STABLE version

Download version 1.7.25 including *full source code*, class files and documentation in ZIP or TAR.GZ format:

- [slf4j-1.7.25.tar.gz](#)
- [slf4j-1.7.25.zip](#)

## Java 9 Modularized EXPERIMENTAL version

Download version 1.8.0-alpha2 including *full source code*, class files and documentation in ZIP or TAR.GZ format:

- [slf4j-1.8.0-alpha2.tar.gz](#)
- [slf4j-1.8.0-alpha2.zip](#)

### Previous versions

Previous versions of SLF4J can be downloaded from the [main repository](#).

**Figure 3: Stable JAR Files from SLF54J Portal**

## Implementation Scenario of Recommendation Engine

### Phase – 1 : Products Table

Table 1 – Product Table of User Purchase

Price	Product
100	Product-1
100	Product-1
80	Product-2
80	Product-2
40	Product-3
30	Product-4
20	Product-5

20	Product-5
----	-----------

## Phase – 2 : Products Occurrences Count

Table 2 – Product Occurrences

Price	Occurrences	Product
100	2	Product-1
80	2	Product-2
40	1	Product-3
30	1	Product-4
20	2	Product-5

## Phase – 3 : Sorting

Table 3 – Sorted Product Occurrences

Price	Occurrences	Product
100	2	Product-1
80	2	Product-2
20	2	Product-5
40	1	Product-3
30	1	Product-4

Phase – 4 : New Arrivals Products Table

Table 4 – New Arrival of Products

Price	Product
120	Product-3
90	Product-6
150	Product-7
90	Product-7
190	Product-3

Phase – 5 : Recommendations (Classical Approach)

New Arrival Products – Array ( => Product-3 => Product-6 => american Product-7  
=> Product-7 => Product-3 )

Table 5 – Recommendations in Classical Approach

Price	Recommended Puchase Item	Price	Earlier Similar Puchased Item
190	Product-3	40	Product-3

Execution Time -> 1.0321700572968 MicroSeconds

Phase – 1 : Products Table

Table 6 – Product Table

Price	Product
100	Product-1
100	Product-1
80	Product-2
80	Product-2
40	Product-3
30	Product-4
20	Product-5
20	Product-5

Phase – 2 : Products Occurrences Count

Table 7 – Product occurrences

Price	Occurrences	Product
100	2	Product-1
80	2	Product-2
40	1	Product-3
30	1	Product-4
20	2	Product-5

Phase – 3 : Sorting

Table 8 – Sorted Product occurrences

Price	Occurrences	Product
100	2	Product-1
80	2	Product-2
20	2	Product-5
40	1	Product-3
30	1	Product-4

Phase – 4 : New Arrivals Products Table

Table 9 – New Arrival of Products

Price	Product
120	Product-3
90	Product-6
150	Product-7
90	Product-7
190	Product-3

Phase – 5 : Recommendations (Proposed Approach)

Table 10 – Recommendations in Proposed Approach

Price	Product
90	Product-6
90	Product-7

Execution Time -> 0.042825937271118 MicroSeconds

### Comparative Analysis

Table 11 – Comparison of Execution Time

Proposed Approach	Classical Approach
0.29401683807373	1.0620608329773
0.22601199150085	1.0590600967407
0.059003114700317	1.0650610923767
0.057003021240234	1.0620610713959
0.032001972198486	1.0740621089935
0.03800201416	1.08206200599

0156	67
0.02600193023 6816	1.04006004333 5
0.02900195121 7651	1.02605891227 72
0.07700395584 1064	1.02605915069 58
0.03100085258 4839	1.04129910469 06
0.03300213813 7817	1.04038310050 96
0.03700208663 9404	1.03823590278 63
0.03400206565 8569	1.03217005729 68



Figure 4 – Comparison of Execution Time

### Pragmatic Comparative Analysis

Proposed Approach	Classical Approach
91	83
93	85
97	87
91	84
92	89
91	80
91	81
97	90



97	83
94	90
97	89

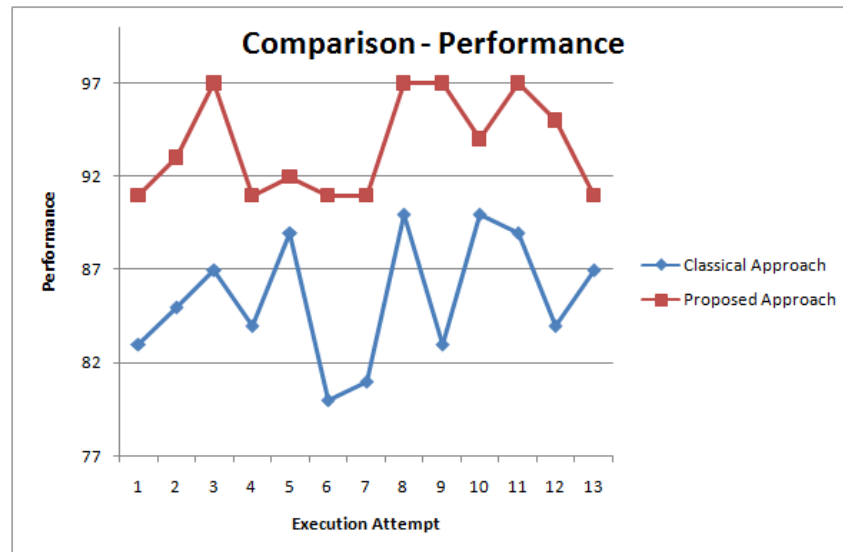


Figure 5: Comparison of Performance

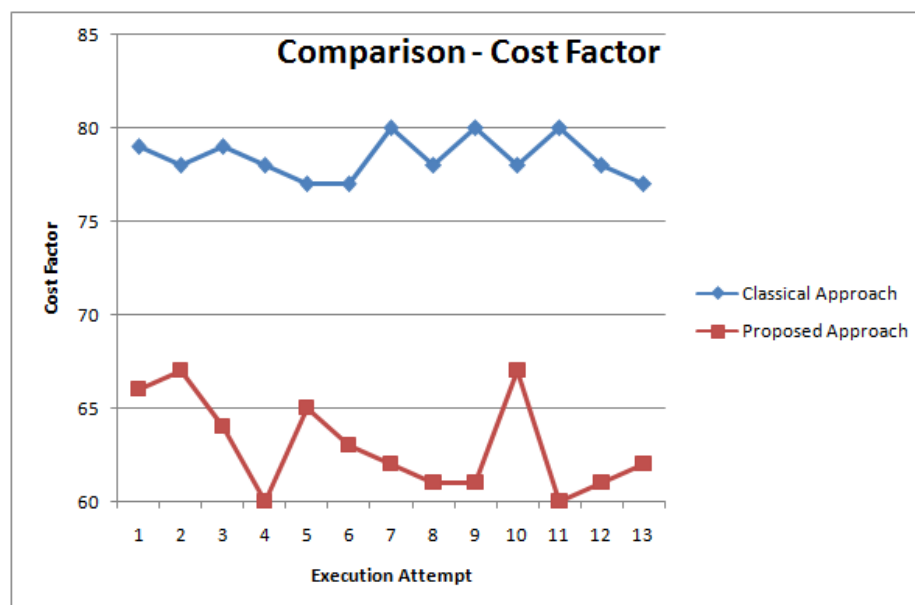


Figure 6: Comparison of Cost Factor

## **Conclusion**

The research problems can be solved effectively using Apache Mahout with the customized algorithms in multiple applications including Malware Predictive Analytics, User Sentiment Mining, Rainfall Predictions, Network Forensic and Network Routing with deep analytics. Now days, the integration of deep learning approaches can be embedded in the existing algorithms so that higher degree of accuracy and optimization in the results can be achieved.

## **References**

- Ji C, Li Y, Qiu W, Awada U, Li K. Big data processing in cloud computing environments. InPervasive Systems, Algorithms and Networks (ISPAN), 2012 12th International Symposium on 2012 Dec 13 (pp. 17–23). IEEE.
- Madden S. From databases to big data. IEEE Internet Computing. 2012 May;16(3):4–6.
- Schlesinger MI, Hlavác V. Supervised and unsupervised learning. Artificial Intelligence.;1:48.
- Schelter S, Owen S. Collaborative filtering with apache mahout. Proc. of ACM RecSys Challenge. 2012.
- Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. InProceedings of the 10th international conference on World Wide Web 2001 Apr 1 (pp. 285–295). ACM.
- Gori M, Pucci A, Roma V, Siena I. ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines. In IJCAI 2007 Jan 6 (Vol. 7, pp. 2766–2771).
- John Walker S. Big data: A revolution that will transform how we live, work, and think.
- Swan M. The quantified self: Fundamental disruption in big data science and biological discovery. Big Data. 2013 Jun 1;1(2):85–99.

- Provost F, Fawcett T. Data science and its relationship to big data and data-driven decision making. Big Data. 2013 Mar 1;1(1):51-9.
- Boyd D, Crawford K. Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. Information, communication & society. 2012 Jun 1;15(5):662-79.
- Lohr S. The age of big data. New York Times. 2012 Feb 11;11(2012).