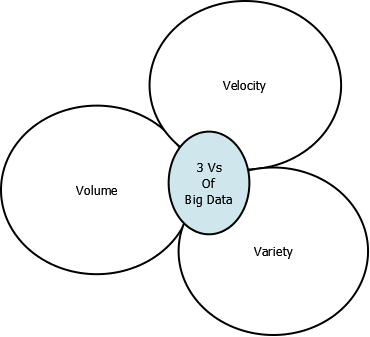
**CHAPTER ONE**

**INTRODUCTION**

**1.1 INTRODUCTION**

Growing technologies like Cloud Computing, Internet of Things, etc, also leads to a tremendous increase in the volume of data over the Internet. This data is termed as Big Data. Big data [1] is a collection of large and complex datasets which is very difficult to process using traditional data processing applications and tools. According to a survey by Science Daily in 2013, 90% of data over the internet is produced in last two years [2]. As shown in **Fig.1.1** Big data [3] have three main attributes called as three Vs of Big data i.e. volume, variety, and velocity. Volume is the primary attribute of Big Data, which refers to the storage space required by data in terabytes or petabytes. Variety in big data means it can be structured, unstructured and semi-structured type of data. Velocity defines that datasets in big data can be real time, near time, batch or stream data.



**Fig 1.1 3 Vs of Big Data**

Storage is not the only problem with big data, but another main problem is how to analyze this much amount of data. For that Big Data Analytics concept is used. In big data analytics, different analytic techniques are applied to big data. Big data enhances the results of analytic tools because it provides gigantic statistical samples.

Nowadays, large data sets are publically available for research purpose. These datasets can be of weather forecasting, health care i.e. medical data, image processing, intrusion detection and business data. We will take one of these datasets to evaluate the performance of proposed Multi-Classifier.

Data mining [4] is an analytical process designed to extract useful patterns from any dataset. These patterns can be further used for building users profile, detecting anomalous data and misuse detection, identifying features from image data, and drawing a conclusion for decision making in business intelligence. Machine learning is one of the best approaches for data mining. Machine learning algorithms [5] are set of algorithms that can learn patterns from data and make the predictions accordingly. They are used to discover valuable information from a large dataset. It brings together computer science and statistics to increase the prediction power. They can be further classified into three categories supervised, unsupervised and semi-supervised algorithms. Supervised machine learning builds classification model by already classified data called training data while in unsupervised learning algorithms data is classified into different classes with high intra-cluster similarity and inter- cluster dissimilarity.

Classification algorithms are the supervised type of mining algorithms which means a set of training data is provided for building the classifier model. The classifier can be further categorized into Single Classifier and Multi-Classifier. Single classifier is the classification model built by one classification algorithm. While Multi-Classifier [6] combines the approximations of two or more different training classifiers to provide better results.

Deep Learning [7] is another type of supervised machine learning algorithms which is used to learn complicated function that can represent high-level abstractions. Deep learning performs many operations on multiple levels such as in complicated propositional formulae re-using many sub-formulae.

Our objective is to design a hybrid Multi-Classification algorithm for big data analytics or analyzing a large amount of data, which will be more robust than existing algorithms in Hadoop map-reduce environment. The proposed algorithm is based on top classification algorithms like SVM, C4.5, K Nearest Neighbor, Naïve Bayes, Deep learning, etc. Feature extraction is the main focus for designing this algorithm.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

Presently, a lot of work is going on machine learning algorithms and big data analytics. In this chapter, we have reviewed some of the major existing work in the areas of classification, multi-classifiers, and big data analytics.

**2.2 CLASSIFICATION**

Classification is a type of supervised machine learning algorithms in which a classifier is built from training data samples i.e. those samples whose class is already known and that trained classifier are further used to predict the classes of unknown class labels. As multi-classifier is built by combining two or three classification algorithms so, we have reviewed following algorithms.

* Decision Tree
* Naïve Bayes
* K-Nearest Neighbor
* SVM

The performance of a classifier can be compared some parameters like Accuracy, Sensitivity, Specificity, Precision, Recall, and ROC Curve. The value of these parameters can be estimated by True Positive, True Negative, False Positive and False Negative.

Accuracy is one of the primary measures for describing the performance of any algorithm. It represents the degree to which an algorithm can correctly predict the positive and negative instances and is calculated by the formula:

(1)

Sensitivity measure the proportion of positives that are correctly identified by a learning method and are calculated by the formula:

(2)

Specificity is the measure of the proportion of negatives that are correctly identified by a learning algorithm and is calculated by the formula:

(3)

Precision is the fact of being accurate and correct. Precision gives the idea of correctly predicted instances. It is measured as proportion of true positive from all positives and is calculated as:

(4)

Recall measure how much relevant data is retrieved from any machine learning algorithm. It focuses on the valuable information.

ROC Curve [8] is a graph that describes the performance of classifiers. It is plotted between False Positive (i.e. 1-specificity) and True Positive (sensitivity). In RoC curve, AuC i.e. area under the curve is directly proportional to its performance. Classifiers having more area under the curve will have high performance and vice-versa.

**2.2.1 DECISION TREE**

A Decision Tree [9] is very simple and widely used classification technique. In decision tree classification, a tree like a classifier is built based on some already classified data samples. In decision tree classifier, all non-leaf node represents attributes and leaf node represents the class labels. For any particular data sets, a lot of decision trees are possible. But the main point of focus is splitting criteria i.e. how to get more informative attributes at the upper part of the tree. So a lot of techniques are applied to fulfill this criterion like Gini index, information gain, entropy, etc. Examples of decision tree based algorithms are Id3, C4.5, etc.

**2.2.2 NAÏVE BAYES**

Naïve Bayes [10] is a very robust classifier that is mainly used for handling noisy data. It is based on Bayes theorem, which uses conditional probability to predict the class label. It assumes that attributes and class labels are independent of each other. It computes posterior probability to classify the test records.

To predict each class C, it uses the following equation.

(5)

Where P(C) is the prior probability, P() is the maximum likelihood and P(X) is the evidence.

**2.2.3 K-NEAREST NEIGHBOR**

KNN [11] is a simple majority voting classifier where k represents the number of classes. It is two-stage classification process wherein the first stage we determine the nearest neighbors of the data element and in second stage class is predicted by voting of these neighbors. KNN requires a proximity measure to select nearest neighbors, for example, distance matrix and a classification function to predict the class like voting.

KNN have many advantages like they don’t require any model building and can have arbitrarily shaped decision boundaries. But their main disadvantage is they predict the classes by local computation, not on global computation.

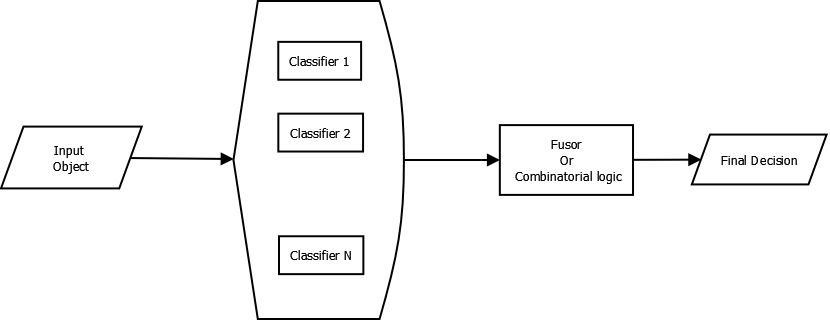
**2.2.4 SVM**

SVM [12] is a supervised type of machine learning algorithms which can be used for regression, classification, and outlier detection. SVM Classifier uses the concept of hyperplanes to classify data. A hyperplane is a subspace of vector space that has one less than the dimensions of vector space. An optimal hyperplane finds maximum margin between two different planes. Support vectors are most important data points in hyperplanes. A hyperplane H in Rn can express as [13]:

where a is an element of Rn, a!=0 and b is an element of R are given. SVM perform well with both linear and nonlinear data sets, but it requires extensive training time to build a classifier[14].

**2.3 MULTI-CLASSIFIERS**

Multi-Classifier [6] is a combination of approximations calculated by different individual classifiers in a proper fashion for extracting useful information from data. The combined result provides, at least equal or better results than a single classifier. However, Multi-Classifier [15] will work better only when combined classifiers are independent of each other and have more than 50 % of accuracy i.e. they are not random classifiers. Multi-Classifier provides a solution to the major problems like mining difficult patterns from large and noisy data sets. According to Romesh et al. [6] multi-classifier is used to get the best classification by increasing accuracy and efficiency of the classification algorithms. Wozniak et al. [15] stated that Multi-Classifier is a type of hybrid intelligent system that focuses on a combination of different homogeneous and heterogeneous classifiers to produce the final model.

****

**Fig 2.1: Overview of Logic of Multi-Classifier Systems**

**Fig.2.1** is describing the general structure of a multi-classifier system. In this structure, input object is provided to a set of the diverse classifier and then a combinatorial function is applied to achieve a final decision.

**DIVERSITY**

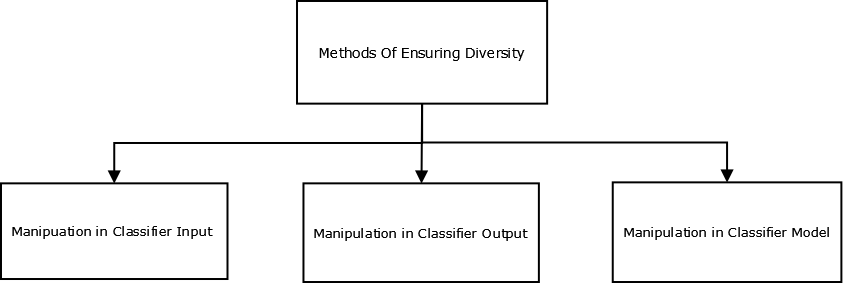
Diversity [16] among classifier is the primary demand of designing a multi-Classifier. Diversity defines the degree of independence among different classifiers. Multi-Classifier [15] requires a set of independent classifiers to the ensemble, for better performance of classification.

**DIVERSITY MEASURING METHODS**

Unfortunately, no standard method is there to calculate diversity of classifiers. It is still an open research topic. Brown et al [17] have formulated following taxonomy for measuring diversity.

* Pairwise diversity measures: It measures the diversity between two of the ensemble classifiers. Some methods of pairwise diversity measures are following.
  + The Q statistic.
  + The correlation coefficient – ρ.
  + The disagreement measure.
  + The double fault measure.
* Non-pairwise diversity measures: It is used to measure the diversity for more than two classifiers. Some of the methods of non-pairwise diversity measures are listed below.
  + The entropy measure (E).
  + The Kohavi-Wolpert measure (KW).
  + The measure of inter-rater agreement (κ).
  + The measure of difficulty (θ).
  + Generalized diversity measure.
  + Coincidence failure diversity measure.

**METHODS TO INCORPORATE DIVERSITY**

****

**Fig 2.2: Taxonomy of Ensuring Diversification**

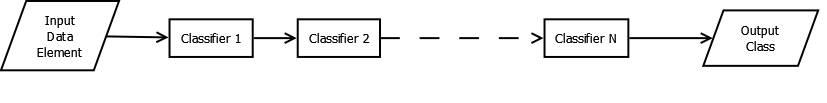
Diversification of the classifiers can be assured by manipulation of classifier’s input, output, and models. **Fig 2.2** depicts the taxonomy of ensuring diversification**.**

**2.3.1 MULTI-CLASSIFIER TOPOLOGIES**

After diversity, the next major factor for designing multi-classifier is the combination logic of different classifiers. Some of the previous research show that a proper fusion strategy can further improve the performance of a multi-classifier. Lu et al. [18] have classified MCS topologies into following three categories.

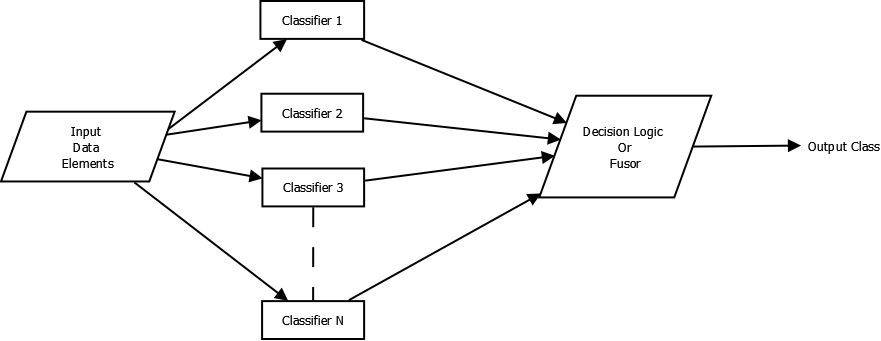
* Serial or Cascading Multi-Classifier.
* Parallel Multi-Classifier.
* Hybrid Multi-Classifier.

**In Serial or Cascading Multi-Classifier,** different classification algorithms are applied to an input data set in a sequence one after another. All these classifiers imply some rank or order by which they are arranged in a sequence. If the results of the primary classifier are not trusted due to low support or confidence, then this input is provided to next classifier. This approach is used to design cost-effective multi-classifier. The major disadvantage of this approach is that later classifiers are not able to correct the mistakes made by the earlier classifiers. **Fig.2.3** is depicting the general approach of cascading multi-classifier topology.



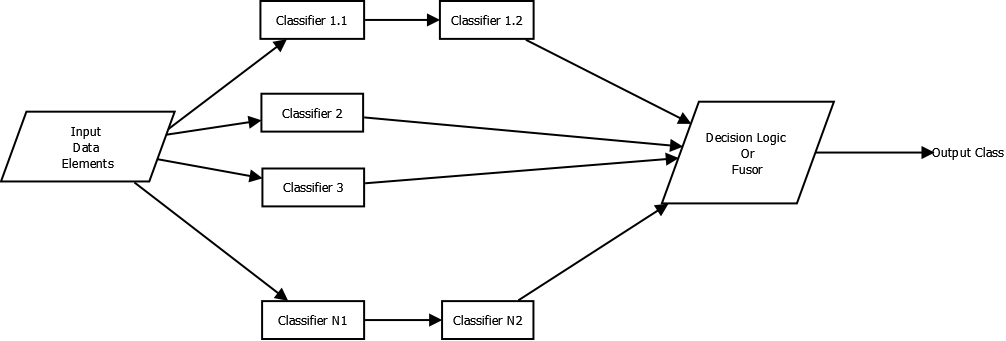
**Fig.2.3 Classifiers in Cascading Fashion**

**In Parallel Multi-Classifiers,** same input is provided to different classifiers and then a combinatorial function is applied to give the final decision. The combinatorial function is the main point to focus on this topology because the proper selection of combinatorial or decision logic can lead a multi-classifier to its peak performance while the improper selection of combinatorial strategy can adversely affect its overall performance. Multi-Classifiers are mostly implemented in a parallel fashion. Some of the famous examples of this topology are Voting, Stacking, and belief integration. Nowadays, these features are also provided in a very famous mining tool called WEKA [19]. **Fig.2.4** is showing the parallel topology of multi-classifier.



**Fig.2.4 Parallel Multi-Classifier Topology**

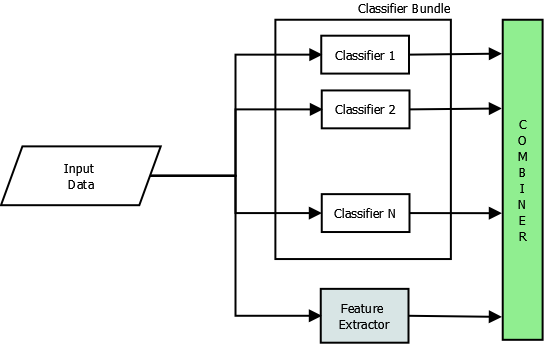
**Hybrid Approach** combines [1] both serial and parallel topology to get optimal performance as shown in **Fig 2.5**. This methodology can be used as an error-checking approach that will remove the disadvantages of both of the topologies mentioned above. Hybrid topology will choose the best classification method for the given input.



**Fig.2.5 Hybrid Topology for Multi-Classifier Systems**

**FEATURE-BASED MULTI-CLASSIFIER ARCHITECTURE**

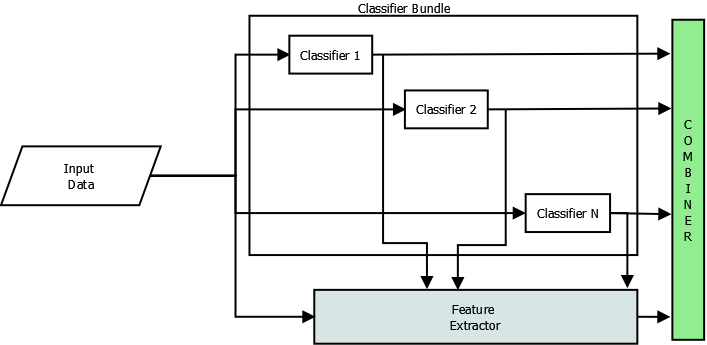
Feature**-**based multi-classifiers are dynamically adaptive in nature. In these architectures, input pattern can be used to select the combinational function, classifiers or both. Wanas et al. [20] have proposed two feature based architectures. In first architecture, features are extracted only from input values while the second architecture is based on both input and output values of the individual classifiers. **Fig.2.6** is showing first feature based architecture, in which different classifiers *Cj* used to predict the class *wi*of input data *x*. Another classifier is used as a feature extractor to extract the meaningful features from the input pattern *x*. This feature extractor is used to provide the probability *ei(x)*, which denotes the confidence that input *x* belongs to class *wi.* At last, the combiner is used to combine the results of individual classifiers and output of feature extractor. The combine can be any of the combinatorial function such as MAX, SUM, PROD, etc.



**Fig.2.6 Feature-based classification by input values**

The output of this feature based multi-classifier is represented by Eq. (1).

(6)



**Fig.2.7 Feature-based multi-classifier by both input and output of individual classifier**

Second feature base architecture is depicted in **Fig.2**.7 In this architecture output *wi*of the individual classifier is used for extracting meaningful information to predict confidence in each classifier. The feature extractor uses weighted matrix *wij*. The output of this feature based multi-classifier can be represented by Eq. (7).

(7)

**2.3.2 EXISTING WORK**

Borji et al. [21] have provided a comparative analysis of different multi-classifier with same classification algorithms but different combinatorial logics. They have proposed these algorithms for designing network intrusion detection system. They have used DARPA data set for both training and testing of classifiers. They have used SVM, Decision Tree, Neural Network and Nearest Neighbor Classifier for building these multi-classifiers. They have used three different combinatorial functions for combining the results of multi-classifier namely majority voting, Bayesian average and belief networks. Then they have provided the results for different combinatorial approaches.

Dong et al. [22] have introduced the concept of multi-classifier to network traffic classification. They have used three single classifier SVM, Naïve Bayes, C4.5 and one parallel multi-classifier based on these algorithms. In the experimental setup, they have locally collected the network traffic. Then they have applied these entire algorithms on same datasets and compared the performance of all these algorithms. Their final results are showing that multi-classifier is providing better results than single classifiers.

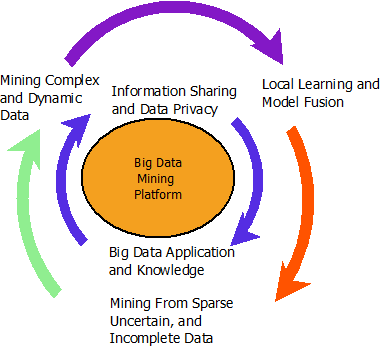
Neelam et al. [23] have used a combination of Naïve Bayes and Naïve Bayes tree to improve the detection rate of network Intrusion Detection System. They have used NSL-KDD’99 [24] data set for both training and testing purpose. They initially extracted features according to a different attack and then they have used WEKA to perform the experiments of building and testing classifiers.

**2.4 BIG DATA ANALYTICS**

Mining useful pattern from large structured, semi-structured and unstructured datasets that are very complex to handle is termed as Big Data Analytics. Minelli et al [3] defined that “ *Big Data* process is not only referred to storing a large amount of different data sets in petabytes or terabytes but also it also adds the ability to make a better decision and take a better decision on time”. Big data means broad sample to train a classifier that ultimately results in enhanced performance of that classifier. Wu et al. [4] have defined big data characteristics as HACE theorem that describes that “Big Data starts with large volume, Heterogeneous, Autonomous source with distributed and decentralized control, and seeks to explore Complex and Evolving relationship". And due to these challenges, it is very difficult to mine useful information from big data. So from above we can conclude that big data enhances the performance of different classifiers but major challenges are how to implement analysis algorithms for a huge amount of heterogeneous data sets with complex and evolving relationships.

**PROPOSED FRAMEWORKS/ARCHITECTURES OF BIG DATA ANALYTICS**

As shown in **Fig.2.8** Wu et al. [25] have proposed a Big Data Processing Framework. They have defined three-tier architecture that center around “Big Data Mining Platform”. In this architecture, Tier I is describing low-level data accessing and computing, Big Data application domain, and challenges on information sharing and privacy. Tier II, focuses on high-level semantics, application domain knowledge, and user privacy issues. And Tier III is describing the true challenges of implementing mining or machine learning algorithms for Big Data Analytics.

****

**Fig.2.8 Big Data Processing Framework Proposed By Wu et al [25]**

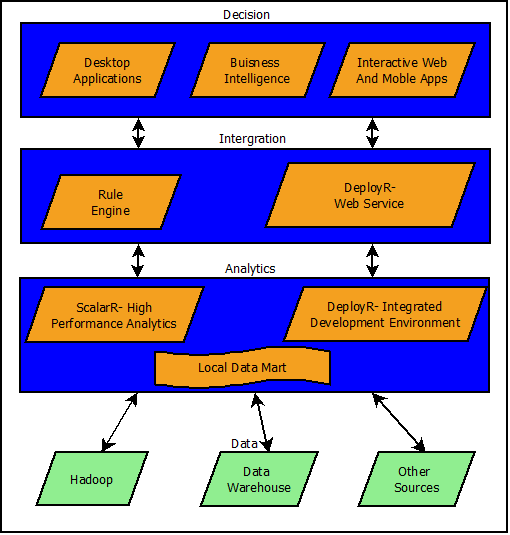
David Smith has also proposed architecture for Real-Time Big Data Analytics. **Fig.2.9**. is depicting that architecture [26]. This architecture has classified big data analytics process into four different phases.

**Phase1: Data,** which is describing various storage sources of big data such as Hadoop, Spark, Data Warehouses and other data sources.

**Phase2: Analytics,** which is describing how different algorithms are implemented on Big Data. Smith has used R language for scaling and development environment. This phase is genuinely describing various operations of analytics in R.

**Phase3: Integration** is focusing mainly on the ways to integrate various patterns and rules extracted from big data in analytics phase. This phase is also deploying R to provide several web services.

**Phase4: Decision** phase is concentrating on the real application of extracting meaningful information from large datasets. These applications are like Health Care Decisions, Business Intelligence, etc.



**Fig.2.9. Big Data Analytics Architecture by Smith [27]**

**2.5 INFERENCES DRAWN FROM LITERATURE REVIEW**

* Most of the existing algorithms are not able to handle a large volume of data.
* Most of the combined multi-classifiers are implement using WEKA for small datasets.
* Existing classifiers are not able to handle complex data sets like an unstructured and semi-structured dataset.
* Hence, our primary focus is to compare the performance of existing classifiers and combining them in such a way that they will be more robust and give better performance than existing algorithms.

**2.6 SCOPE OF THE WORK**

We want to design a hybrid multi-classification algorithm which is more robust than existing algorithms. This algorithm will quickly handle large and complex data sets. This can be applied to large and complex datasets like of network data, medical data, etc.

**CHAPTER 3**

**PROPOSED WORK**

**3.1 PROBLEM STATEMENT**

The Problem statement of our proposed work can be stated as follows:

*“Big Data Analytics Using Multi-Classifier for Large Datasets”*

* + 1. **OBJECTIVES OF PROPOSED WORK**
* To implement already existing classification approaches in Hadoop/ Spark environments.
* To analyze the performance of existing classification algorithm while handling large datasets.
* To design a more robust algorithm using multi-classification approach.
* To provide a better approach for classifying large datasets and complex datasets.
  1. **PROPOSED WORK**

**MULTI-CLASSIFIER: BASED ON DIFFERENT TOPOLOGIES**

We will design a multi-classifier on various topologies like serial or cascading fashion and parallel fashion depicted in **Fig.4** and **Fig.5**. We will use existing algorithms like Decision Tree, Naïve Bayes, K-Nearest Neighbor and SVM. Then we will compare their performance with parameters like accuracy, precision, recall, etc., mentioned in **Eq(1-4).**

**MULTI-CLASSIFIER: FEATURE EXTRACTION BASED**

After analyzing multi-classifier based on topologies, we will try to add the concept of feature extraction into it. **Fig.7** and **Fig.8** are describing the feature based architectures of multi-classification then again we will compare this implementation with existing approaches.

**MULTI-CLASSIFIER: COMBINING FEATURES OF ALGORITHMS**

In this architecture, best features of algorithms are combined to provide a better approach for the analysis of a large amount of heterogeneous data.

**CHAPTER 4**

**METHODOLOGY**

**4.1 HADOOP MAP-REDUCE ENVIRONMENT**

Hadoop [28] is open-source software developed by Apache to for reliable, scalable and distributed computing. It provides a framework for distributed processing of large datasets over clusters of the computer using simple programming model. Hadoop provides highly scalable and available service on the top of a cluster of computers. Hadoop includes some following modules.

**Hadoop Common:** This includes the common utilities and libraries used by other Hadoop modules.

**Hadoop Distributed File System:** This module defines the distributed file system for Hadoop, which provides high-throughput access to data.

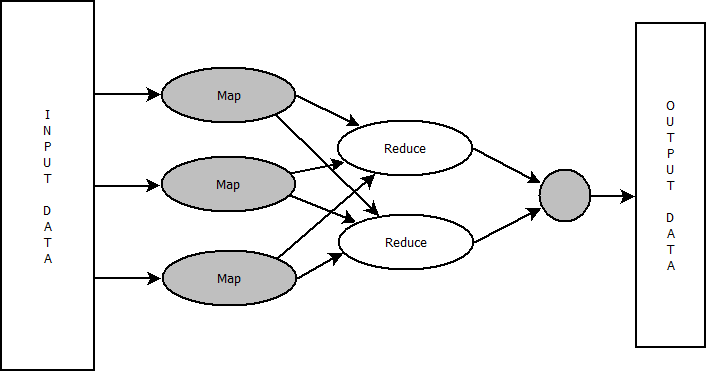
**Hadoop Yarn:** Yarn Stands for Yet Another Resource Negotiator. This module provides resource management framework for handling resource request.

**Hadoop MapReduce:** This module describes a method for processing large sets of data in parallel. Map-Reduce [30] is a programming model used for processing large datasets. As shown in fig 1, it is a 2-step process in first step mapper function converts all input into key/value pairs and passes them to reducer function. Then in second step reducer function merges the all intermediate values of associated with the same key to form a smaller set of values.

Map and Reduce functions can be depicted from Eq.1 and 2. The major advantage of Map-Reduce function is that it provides high scalability over multiple computing nodes and secondly it provides a good framework for distributed computing.

Map (k, v) → list(k1, v1)

Reduce (k1, list(v1)) → list(v1)



**Fig 4.1 Map Reduce Algorithm for Large Datasets**

**HADOOP ECOSYSTEM**

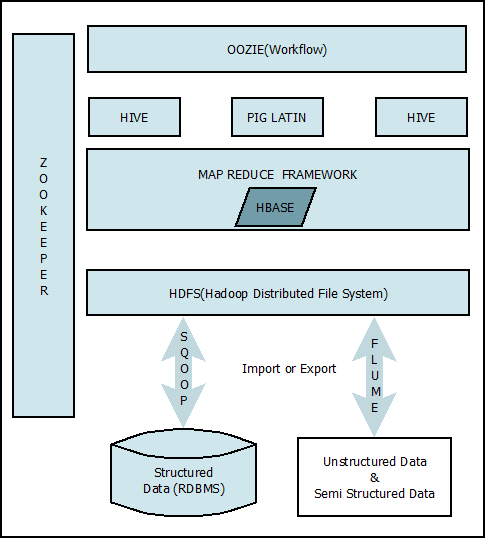
Hadoop ecosystem is a collection of additional software packages that make it easier to load and process data on Hadoop cluster. As shown in **Fig. 10** Hadoop ecosystem is classified into few components. Each component is used for some particular function as follows.

**HBase**

HBase is a Hadoop database that runs on HDFS. It is an NOSQL database that is created for storing large tables with billions of records. Hbase is high scalable and fault tolerable. It only performs batch processing.

**Hive**

Facebook created the hive for dealing structured data by SQL-like queries. The query language used is called as HQL i.e. Hive Query Language which similar to SQL. It also runs map-reduce programs in the backend to process data in HDFS.



**Fig. 10. Hadoop Ecosystem**

**Pig**

Pig is similar to Hive used for handling structured data. It was developed by Yahoo for those who loves scripting but don’t want to use Python/Java. A single PIG LATIN program involves series of operation and transformations to be performed on input data which runs map-reduce programs in the backend.

**Mahout**

Mahout is an open-source machine learning library written by Apache. Mahout is one of the most widely used machine learning tool for handling large datasets. It is primarily designed for classification, clustering and collaborative filtering i.e. recommender systems. Three major components of mahout are following.

* It provides an environment for building scalable algorithms.
* It provides a platform for implementing many new Scala + Spark and H2O algorithms.
* Most importantly, Mahout’s mature Hadoop map/reduce based implementation of existing algorithms.

**Oozoie**

It describes the workflow for managing Hadoop jobs. It is a web-based Java application implemented in Java-Servlet Container.

**Zookeeper**

It is an open-source server used for writing highly reliable distributed applications. It is a centralized mainly used for maintaining configuration information, naming, providing distributed synchronization and group services.

**Spark**

Spark [29] is a software of Apache that can also be used for making clusters handle large datasets. This provides 100 percent faster processing power than Hadoop Map-Reduce in memory. That's why it is also called as Light-Fast cluster computing. Spark is a research project developed by AMPLab in UCBerkley. Spark is mainly used for big data analytics. Their main focus for this project is to design a programming model which supports a much wider class than Hadoop map-reduce.

Spark has a lot of interesting features from which some of them are listed below.

* It can run on Hadoop, Mesos Standalone, or in a cloud.
* It can be used to write applications in Python, Java and Scala.
* It provides a lot of inbuilt libraries like [SQL and DataFrames](http://spark.apache.org/sql/), [MLlib](http://spark.apache.org/mllib/) for machine learning, [GraphX](http://spark.apache.org/graphx/), and [Spark Streaming](http://spark.apache.org/streaming/).

**4.2 MAP REDUCE / SPARK BASED MULTI-CLASSIFIER FOR LARGE DATA SETS**

We will make a Hadoop/Spark Cluster of 6 computers and with one master and five slave machines. And we will compare the performance of different classifiers for same datasets. After that, we will perform a different combination of algorithms to get better results.

**REFERENCES AND BIBLIOGRAPHY**

[1] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications,* vol. 19, pp. 171-209, 2014.

[2] ScienceDaily. (2013, 26 Nov). *Big Data, for better or worse: 90% of world's data generated over last two years*.

[3] M. Minelli, M. Chambers, and A. Dhiraj, *Big data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*: John Wiley & Sons, 2012.

[4] J. Han, M. Kamber, and J. Pei, *Data mining: Concepts and Techniques: Concepts and Techniques*: Elsevier, 2011.

[5] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*: Springer Science & Business Media, 2013.

[6] R. Ranawana and V. Palade, "Multi-classifier systems: Review and a Roadmap for Developers," *International Journal of Hybrid Intelligent System,* vol. 3, pp. 35-61, 2006.

[7] Y. Bengio, "Learning deep architectures for AI," *Foundations And Rrends in Machine Learning,* vol. 2, pp. 1-127, 2009.

[8] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, pp. 861-874, 2006.

[9] S. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE transactions on systems, man, and cybernetics,* vol. 21, pp. 660-674, 1991.

[10] I. Rish, "An Empirical Study of the Naive Bayes Classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, pp. 41-46.

[11] P. Cunningham and S. J. Delany, "k-Nearest Neighbour Classifiers," *Multiple Classifier Systems,* pp. 1-17, 2007.

[12] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine learning,* vol. 20, pp. 273-297, 1995.

[13] O. Ivanciuc. (2005, 2 Dec 2015). *SVM - Support Vector Machines,* . Available: <http://www.support-vector-machines.org/>

[14] Y. B. Bhavsar and K. C. Waghmare, "Intrusion Detection System Using Data Mining Technique: Support Vector Machine," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, pp. 581-586, 2013.

[15] M. Woźniak, M. Graña, and E. Corchado, "A Survey of Multiple Classifier Systems as Hybrid Systems," *Information Fusion,* vol. 16, pp. 3-17, 2014.

[16] L. I. Kuncheva and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and their Relationship With the Ensemble Accuracy," *Machine learning,* vol. 51, pp. 181-207, 2003.

[17] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity Creation Methods: A Survey and Categorisation," *Information Fusion,* vol. 6, pp. 5-20, 2005.

[18] Y. Lu, "Knowledge Integration in a Multiple Classifier System," *Applied Intelligence,* vol. 6, pp. 75-86, 1996.

[19] U. o. Waikato. (25 Nov ). *WEKA*. Available: <http://www.cs.waikato.ac.nz/ml/weka/>

[20] N. Wanas, "Feature based Architecture for Decision Fusion," University of Waterloo, 2003.

[21] A. Borji, "Combining Heterogeneous Classifiers for Network Intrusion Detection," in *Advances in Computer Science–ASIAN 2007. Computer and Network Security*, ed: Springer, 2007, pp. 254-260.

[22] S. Dong, D. Zhou, and W. Ding, "Traffic Classification Model Based on Integration of Multiple Classifiers," *Journal of Computational Information Systems,* vol. 8, pp. 10429-10437, 2012.

[23] N. Sharma and S. Mukherjee, "A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS," *Procedia Technology,* vol. 6, pp. 913-921, 2012.

[24] (1999). *NSL KDD*. Available: <http://kdd.ics.uci.edu/databases/kddcup1999.html/>

[25] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with Big Data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, pp. 97-107, 2014.

[26] M. Barlow, *Real-Time Big Data Analytics: Emerging Architecture*: " O'Reilly Media, Inc.", 2013.

[27] D. Smith. (2012, 2 Dec 2015). *Real-Time Predictive Analytics with Big Data, and R*. Available: <http://www.r-bloggers.com/real-time-predictive-analytics-with-big-data-and-r/>

[28] (20 Nov 2015). *Hadoop*. Available: <https://hadoop.apache.org/>

[29] Apache. (2014, 2 Dec 2015). *Apache Spark Lightning-fast cluster computing*. Available: <http://spark.apache.org/>

[30] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM,* vol. 51, pp. 107-113, 2008.