Feature Selection and Map Reduce-based Neural Network Classification for Big Data

Chit Thu Shine, Thi Thi Soe Nyunt
University of Computer Studies, Yangon
chitthushine@ucsy.edu.mm, thithi@ucsy.edu.mm

Abstract

Nowadays, a large amount of digital data is generated from everywhere, every second of the day. One of the challenges is the volume of generated data with high dimensionality. Most of traditional machine learning algorithms are not good in training time and classification result to find hidden insights from these high dimensional data. Back-propagation Neural Network, one of the most popular Artificial Neural Networks, is widely used in many classification applications. To reduce the data dimension, feature selection is needed to consider. MapReduce is a software framework for writing applications which are run on Hadoop that supports rapid computation and processing of Big Data. In this paper, first the dimension of data is reduced using Chi-square method. Then, Backpropagation Neural Network with MapReduce paradigm is used for classification. MapReduce-based Neural Network classifier is constructed using one and two hidden layers. Six different datasets are used as case study and the performance measures involve the training time, accuracy and number of selected features. The results of MapReduce-based Neural Network algorithm training on complete features and features selected subset are compared with WEKA tool and Conventional Back-propagation Neural Network. Based on the experimental results, MapReduce-based Neural Network algorithm give the superior efficiency in training time and accuracy with reduced number of features selected.

1. Introduction

Unprecedented rate of data according to the development of technology in Internet of Things (IoT) devices and social media, structured and unstructured, is generated across the globe. This leads to volume of high-dimensional data which technically constitutes the term known as 'Big Data'. Mostly, big data is characterized with five V's.

i. Volume: the amount of data generated
ii. Velocity: the rate at which the data is being generated
iii. Variety: the heterogeneity of data sources
iv. Veracity: the quality of data to process
v. Variability: data whose meaning is constantly changing

When deeper analysis is required to find insights that are hidden from high voluminous data, machine learning may be more suitable to use [12]. But machine learning algorithms work slowly for large data sets. Hence feature selection has become one of important issues in classification because it results in less training time and may have a considerable effect on accuracy of the classifier. It is used to select optimal feature subsets that are suitable to use in model construction.

In pattern recognition and many other classification applications, artificial neural networks (ANNs) have been widely used. Back-propagation neural network (BPNN) is one of the most popular ANNs. It can approximate any continuous non-linear functions by arbitrary precision with an enough number of neurons [7]. Normally, BPNN training requires a significant amount of time when the size of the training data is large [11]. To fulfill the potentials of neural networks in big data applications, the computation process must be speeded up with parallel computing techniques [2]. In recent development, new ideas in terms of BPNN classification by using parallel environment like Hadoop were provided according to the development of cloud platforms. MapReduce has become a standard computing model in support of big data applications [6]. It provides a reliable, fault-tolerant, scalable, and resilient computing framework for processing and storing massive datasets.

In case of BPNN in MapReduce, each mapper constructs one BPNN and generates various combination of weight index as the "key" and the "value" is used to keep track of the weight value and global average error. After that the reducer collects all mappers' outputs that have the same key and

chooses the best weight value with the smallest global average error.

The rest of the paper is organized as follows. Section 2 gives a review on the related work. Section 3 explains about the feature selection method. Section 4 describes the theory of artificial neural network. Section 5 presents the architecture of Hadoop and MapReduce. The proposed system is demonstrated in section 6 and section 7 evaluates the performance of the proposed algorithms and analyzes the experimental results. Section 8 concludes the paper and presented what the future works would be.

2. Related Work

Many researchers employed feature selection before model construction and parallel design for traditional data mining algorithms using Hadoop and MapReduce architecture are admitted to facilitate for their researches.

Changlong Li et al. [3] implemented an Artificial Neural Networks in MapReduce paradigm. Their experimental technique shows that its results are a great influence in optimizing the system performance and speeding the system up.

Rachana Sharma et al. [10] implemented two traditional machine learning algorithms (Naïve Bayes and K-Nearest Neighbors) using MapReduce paradigm. They have also implemented the standard algorithms for both the classifier in WEKA 3.7 and compared the results of classifier in terms of accuracy and training time for both platforms. Their experiment shows MapReduce platform is faster than WEKA. It is found that WEKA face scalability issue as they move from 20% of the dataset to 100%, while MapReduce prove to be more efficient with larger datasets.

Navjit Singh and Anantdeep Kaur [9] admitted paper that present about feature selection for artificial neural network based intrusion detection system. In this paper, they compared the performance in terms of Detection Accuracy (DA) of Multilayer Perceptron (MLP) based Intrusion Detection system using three feature selection methods: Chi-square, Gain Ratio and Information Gain. The result showed that the Chi-square gave the best detection accuracy and fastest method out of the all.

3. Feature Selection

Feature selection is the process of selecting a subset of relevant features to use in model construction [5]. It is used to select optimal features that are suitable to use in model construction.

3.1. Chi-square feature selection method

[8] Chi-square method is used to test independence level to identify whether there is a considerable relationship between two attributes. The χ^2 value is computed as

$$\chi 2 = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \quad (1)$$

$$e_{ij} = \frac{n_{*j} n_{i*}}{n} \quad (2)$$

Where,

n_{i*}= total number of samples with i[th] the feature value.

n_{*j}= total number of samples in class j.

n = total number for samples.

This method consists of the following two steps:

Step 1: This step is to find P-value by the calculated $\chi 2$ value and degree of freedom.

Step 2: The P-value of Step 1 is to test whether support or reject the null hypothesis in Step 2. The null hypothesis assumes that there is no significant difference between the two attributes. If the P-value <= chosen confidence level, then the null hypothesis is rejected, otherwise it accepts the null hypothesis.

4. Artificial Neural Network

In Artificial Neural Network, a group of artificial neurons are interconnected. A neural network has at least two physical components, namely, the processing elements (neurons) and the connections (links) between them. Every link has a weight parameter associated with it. There are three kinds of neurons.

Input neurons: Neurons that receive stimuli from outside the network.

Output neurons: Neurons whose outputs are used externally.

Hidden neurons: Neurons that receive stimuli from other neurons and whose output is a stimulus for other neurons.

Neural network has one or more layer of neurons followed by output neurons [7].

4.1. Back-propagation Neural Network

Back-propagation Neural Network (BPNN) is a multilayer network including input layer, hidden layer, and output layer. A typical neural network

structure is shown in Figure 1, which consists of an arbitrary number of inputs and outputs and it consists of two steps:

(1) Feed forward the values, and

(2) Calculate the error and propagate it back to the earlier layers [2].

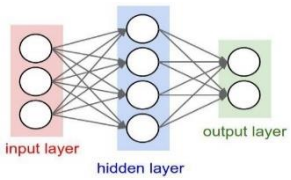

Figure 1. A typical neural network structure

5. Hadoop/MapReduceArchitecture

5.1. Hadoop

Apache Hadoop is an open source distributed processing framework that was designated to run on low-cost hardware [1]. It supports rapid and reliable computations. Data in Hadoop framework is saved in Hadoop File System (HDFS). It splits input file into small chunks known as data blocks. HDFS data blocks are the smallest unit of data in a file system. The default size of the HDFS block size is 128 MB which can be customized by requirement [4]. An HDFS cluster contains name node and data node. Name node is responsible for storing metadata about the data node and managing that data node which store actual data.

5.2. MapReduce

MapReduce is a computation model and software framework for writing applications which are run on Hadoop. It consists of two mainly steps; Map and Reduce. Each step is done parallel on sets of <key, value> pairs. Basically, a mapper function is responsible for actual data processing and generates intermediate results in the form of ⟨key, value⟩ pairs. The data to be processed by an individual mapper is represented by block in HDFS. The number of map tasks is equal to the number of block split [4]. A reducer collects the output results from multiple mappers with secondary processing including sorting and merging the intermediate results based on the key values. Finally, the reducer function generates the computation results [2].

6. Overview of the Proposed System

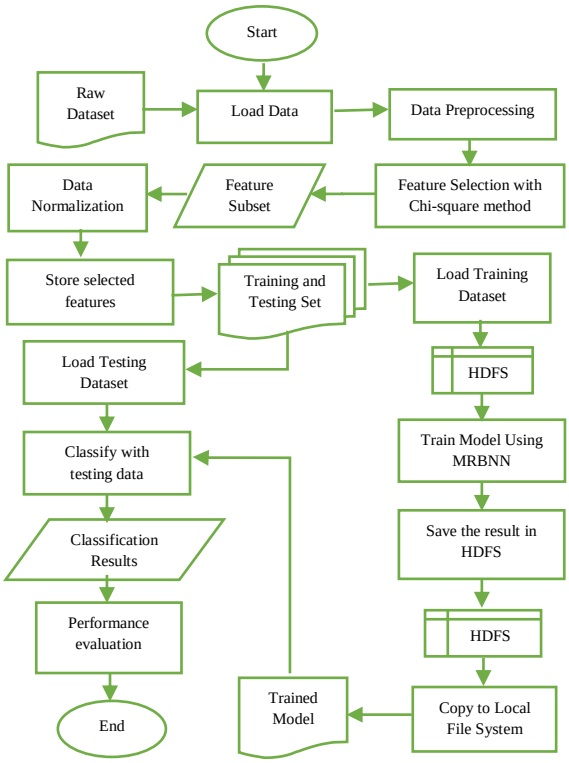

Figure 2. Overview of the proposed system

Figure 2 shows the overview of the proposed system. According to figure, data preprocessing is performed to the input dataset. The proposed system can be classified by the two stages. In the first stage, feature selection is performed by using Chi-square based feature selection method. In the second stage, normalization is performed to the feature selected set and then the result data are split into training and testing set using holdout method. Two thirds of the data are to the training and remaining one third is allocated as the test sets. To train MapReduce-based Backpropagation Neural Network (BPNN) model, the training partition is loaded into the Hadoop File System (HDFS). And then, the MapReduce-based BPNN training is performed to the loaded training set and the outputted model is saved in HDFS and copied it into local file system. The performance evaluation is performed on the classification results of the generated model using testing dataset. In the next sub section, the detail of MapReduce-based BPNN algorithm will be presented.

6.1. MapReduce-based BPNN algorithm

In case of BPNN in MapReduce, each mapper constructs one BPNN for each input split (chunk). Number of mapper's is determined by the number of splits for the input path. In this experiment, input split

size is specified as four megabytes. The number of mappers is equal to the number of input splits. Each split is responsible for each mapper using the map function. The reducer part takes the results of individual mappers and processes them to get the final result. The idea of the model is to build individual classifier on each split. And the reducer chooses the best weight that fit in the smallest error rate from all of these classifiers and save the final trained results to HDFS. The MapReduce-based BPNN algorithm is described as below.

Algorithm 1. MapReduce-based BPNN

Input: The user provides input file location on HDFS, number of input node (iNode), and number of output node (oNode) via input arguments.
Output: The final received Weights are the trained result we want.
m mappers and **one** reducer
Each mapper constructs one BPNN with input node (iNode), output node (oNode) and hidden node (hNode) that was computed by iNode and oNode for each data split.

1. Initialize Randomize Weights ()
2. For iteration <= maxEpoch do
3. Each neuron$_j$ of hidden layer computes

$$I_{jh} = \sum_{i=1}^{n} w_{ij}\ a_i + \theta \quad (3)$$

$$o_{jh} = \frac{1}{\left(1+e^{-I_{jh}}\right)} \quad (4)$$

4. Each neuron$_j$ in output layer computes

$$I_{jo} = \sum_{i=1}^{h} w_{ij}\ o_{jh} + \theta \quad (5)$$

$$o_{jo} = \frac{1}{\left(1+e^{-I_{jo}}\right)} \quad (6)$$

5. In each output, compute

$$Err_{jo} = o_{jo}\,(1-o_{jo})\,(target_j - o_{jo}) \quad (7)$$

$$mse = (target - o_{jo})^2 \quad (8)$$

$$E[e^2] = E[e^2] + mse \quad (9)$$

6. In hidden layer, compute

$$Err_{jh} = o_{jh}\,(1-o_{jh})\ \sum_{i=1}^{n} Err_i\, w_{io} \quad (10)$$

7. For all weight between input and hidden layer do

$$\Delta w_{jh} = w_{ij} + Err_{jh} * o_{jh} \quad (11)$$

8. End for
9. For all weight between hidden and output layer do

$$\Delta w_{jo} = w_{ij} + Err_{jo} * o_{jh} \quad (12)$$

10. End for
11. iteration++;
12. End for
13. $E[e^2]$ =$E[e^2]$/mapper input size
14. For all weight between input and hidden layer
15. output <key, value> pair: <w_{jh}, $E[e^2]$+':'+Δw_{jh} >
16. End for
17. For all weight between hidden and output layer
18. output <key, value> pair: <w_{jo}, $E[e^2]$+':'+Δw_{jo}>
19. End for
    Mapper process finish
20. Reducer collects weight values that have the same key and chooses the best weight value from them
21. for all values of same key do
22. bestWeight = weight value with the smallest error
23. end for
24. output <key, value> pair: <key, bestWeight>

End

As shown in Algorithm 1, MapReduce-based BPNN, Map and Reduce are two different steps. Each step is done in parallel on sets of <key, value> pairs. Each mapper function takes its training data after dividing it on the set of mappers. Each mapper performs backpropagation neural network training which is calculating global average error and weight values that are good fit for the training data and sends results to reducer. After that the reducer collects all mappers' outputs that have the same key and chooses the best weight value with the smallest global average error. The final result, neural network training data, is then written to Hadoop File System (HDFS).

7. Performance Evaluation

In this paper, classification accuracy, training time and number of features selected are measured. All experiments were carried out using holdout method. Two thirds of the data are allocated to the training set, and remaining one third is allocated as the test set. The classifier's accuracy, the possibility of the algorithm that is able to correctly predict positive and negative tuple is calculated by the equation.

$$Accuracy = \frac{TP+TN}{Total\ Number\ of\ Test\ Data} \quad (13)$$

Where,
TP = positive tuples correctly classified as positive
NP = negative tuples correctly classified as negative

7.1. Experimental Environment

In order to evaluate the proposed algorithms, a Hadoop cluster (Pseudo Distributed Mode) is stablished. In Pseudo Distributed Mode, master and slave servers actually run on the same server.

The experimental environment is as follows:

- Operating System: Ubuntu 16.04, CPU: Core i7@3 GHz, Memory: 6 GB

The software versions are as follows

- Hadoop version: 2.7.2 64 bit, and JDK 1.8

In the next sections, the datasets used and performance result are discussed. For MapReduce-

based Neural Network classifier, it is constructed using one and two hidden layers.

7.2. Datasets used

In this experiment, six different data sets, Diabetic Classification and Intrusion Detection (KDD 99), Customer Churn Prediction (KDD 2009), Thyroid Disease Diagnosis, Insurance Data, and Human Activation Recognition (HAR), are used as case study. Table 1 summarizes the main characteristics of these datasets. For each dataset, the number of instances, number of attributes and number of classes are shown in Table 1.

Table 1. Data set description

Dataset Name	No. of Instances	No. of attributes	No. of classes
Diabetics Data	101767	38	2
KDD Cup 1999	125975	42	2
Customer Churn Prediction	15333	307	2
Thyroid Disease	9172	30	2
Insurance	9823	86	2
HAR	4856	352	3

7.3. Performance Result

The analysis includes comparison of three Neural Network models: MapReduce-based BPNN, Multi-Layer Perceptron (MLP) that was implemented in the tool of WEKA 3.8 and Conventional BPNN. Chi-square feature selection method is employed to select features before passing the data sets to the classifier. By using Chi-square feature selection algorithm, it reduced features from 29 to 16, 37 to 26, 41 to 35, 85 to 29, 306 to 126 and 351 to 223 in Thyroid Disease, Diabetic, KDD99, Insurance, Churn Data and HAR datasets respectively. The result details of the average training time in seconds and classification accuracy are presented before feature selection and after feature selection in Table 2. The result details of number of selected features, training time in seconds and accuracy of these three models are presented in that table.

Figure 3 shows the training time comparison of the three models' construction on six datasets with complete features. It is observed that MapReduce-based BPNN is the fastest method out of the all when the dataset becomes large. Conventional BPNN is the laziest model to train for all datasets. According to Table 2 and Figure 3, training time of MapReduce-based BPNN is slightly longer than MLP in WEKA tool in small datasets (Thyroid Disease and Insurance).

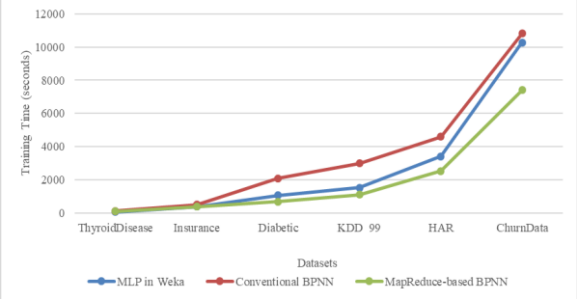

Figure 3. Training time comparison of classifiers on complete features

Figure 4 shows the classification accuracy comparison of the three models on six datasets with complete features. According to the nature of MapReduce-based BPNN algorithm that trains on subset of dataset by splitting the original dataset, it is observed that the classification accuracy of the sequential execution MLP in WEKA tool is higher than the classification accuracy of MapReduce-based BPNN algorithm except from insurance dataset.

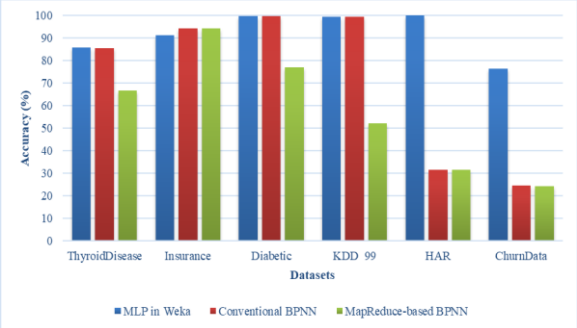

Figure 4. Accuracy comparison of classifiers on complete features

Figure 5 shows the training time comparison of the three models on six datasets after making feature selection. These three models were constructed by passing features subset that was generated by Chi-square feature selection method. The detail comparison results with the reduced number of features are presented in Table 2. By comparing Figure 3 and Figure 5, it can be seen that model construction time after making feature selection is significantly reduced for all of the three models on all of six datasets.

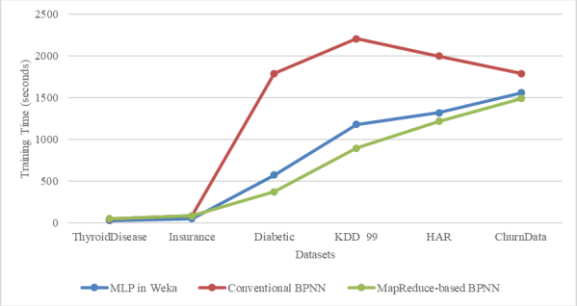

Figure 5. Training time comparison of classifiers on features selected subset

Figure 6 shows the classification accuracy comparison of the three models on six datasets after making feature selection. According to the figure, it can be seen that MapReduce-based BPNN is mostly affected by Chi-square feature selection method because the classification accuracy significantly increases on feature selected subset, but also for the Thyroid Disease dataset accuracy is increased from 85.7 to 88.14 and 85.37 to 87.97 respectively in MLP of WEKA and Conventional BPNN. The accuracy is stable in the remaining datasets by feature selection. By comparing Figure 4 and 6, it is observed that feature selection can retain a suitably accuracy that represent in the complete features by selecting minimal features subset from a problem domain that helps to decrease training time.

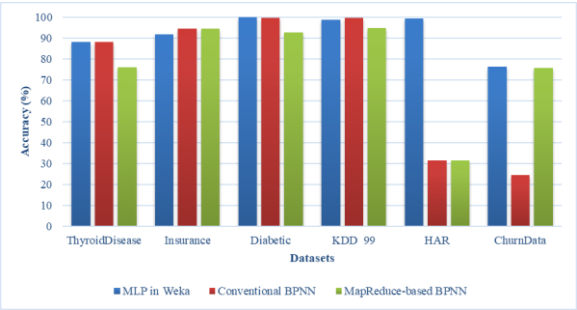

Figure 6. Accuracy comparison of classifiers on features selected subset

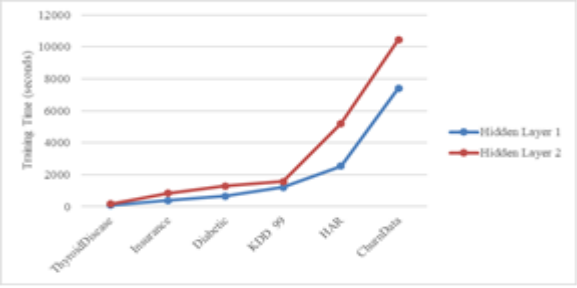

Figure 7. Training time comparison of MapReduce-based BPNN on two different hidden layers

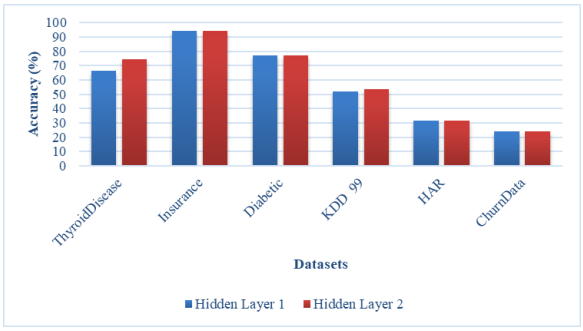

Figure 8. Accuracy comparison of MapReduce-based BPNN on two different hidden layers

The analysis is also made by increasing one more hidden layer in MapReduce-based BPNN algorithm implementation. The training time and accuracy comparison of MapReduce-based BPNN algorithm on two different hidden layers are shown in Figure 7 and 8 respectively. By comparing hidden layer one and layer two implementation, the classification accuracy of layer two implementation isn't so different from the layer one implementation. One of the case studies, ThyroidDisease dataset, is only affected that tends to increase accuracy from 67% to 74% but the accuracy of the remaining case studies is stable. Although accuracy isn't so many, it suffers more training time to build classifier model.

8. Conclusion and Future Work

This paper focus on Chi-square feature selection method and MapReduce-based BPNN. The comparison is also made with MLP in WEKA 3.8 and Conventional BPNN by using six different datasets.

Since MapReduce paradigm is designed for handling big data efficiently, it can be seen that it takes less time of building the model when the size of training data becomes large as compared to MLP in WEKA tool and Conventional BPNN. It takes more training time than standard WEKA tool when the data used in small size datasets. It is found that the accuracy of Conventional BPNN is much higher than MapReduce-based BPNN because conventional BPNN is trained on complete records of dataset while MapReduce-based BPNN algorithm is trained on dataset subset by splitting. And it also found that increasing hidden layer tends to increase training time although it's not so different in classification accuracy when compared to hidden layer 1 implementation.

According to the experimental results, training time of the three models reduced by the aids of Chi-

Table 2. Experimental result details

Dataset Name	All attributes							Selected feature set using Chi-square method						
	#of features	MapReduce-based BPNN		MLP in WEKA		Conventional BPNN		#of features	MapReduce-based BPNN		MLP in WEKA		Conventional BPNN	
		Time (sec)	Accuracy	Time (sec)	Accuracy	Time (sec)	Accuracy		Time (sec)	Accuracy	Time (sec)	Accuracy	Time (sec)	Accuracy
Diabetics Data	37	674	77.081	1060	99.79	2093	99.79	26	374	92.49	575	99.75	1789	99.68
KDD 99	41	1095	52.1	1523	99.34	2995	99.49	35	896	94.64	1180	98.64	2207	99.42
Churn Data	306	7409	24.25	10276	76.32	10820	24.43	126	1490	75.57	1560	76.25	1789	24.43
Thyroid Disease	29	92	66.514	54	85.70	111	85.37	16	54	75.96	30	88.14	51	87.97
Insurance	85	384	94.29	372	91.11	494	94.29	29	89	94.29	55	91.67	87	94.29
HAR	351	2529	31.46	3420	99.88	4597	31.46	223	1218	31.46	1320	99.38	1999	31.46

square feature selection method for all of case studies because feature selection tends to reduce the processor and memory usage. It also aids to increase accuracy or retain a suitably accuracy in representing the original features by selection a minimal feature subset from a problem domain.

The future work will be dedicated to test with dataset that has multi-class label.

REFERENCES

[1] Apache Hadoop, [online] available: http://Hadoop. apache.org/. 2018.

[2] Cao J, Cui H, Shi H, Jiao H, *"Big Data: A Parallel Particle Swarm Optimization- Back-Propagation Neural Network Algorithm Based on MapReduce",* PLos ONE 11(6): e0157551, June 2016.

[3] Changlong Li, Xuehai Zhou, Kun Lu, Chao Wang, Dong Dai. *"Implementing of Artificial Neural Networks in MapReduce Optimization".*

[4] DataFlair Team, *"MapReduce InputSplit vs Block in Hadoop",* [online] available: https://data-flair.training/ blogs/mapreduce-inputsplit-vs-block-hadoop/, April 2017.

[5] Jason Brownlee. *"An Introduction of Feature Selection",* [online] available: https://machinelearning mastery.com/an-introduction-to-feature-selection/, October 2014.

[6] Malak EI Bakry, Soha Safwat, Osman Hegazy, *"Big Data Classification using Fuzzy K-Nearest Neighbor"*, International Journal of Computer Applications (0975-887), Volume 132 -No.10, December 2015.

[7] M. H. Hgan, H. B. Demuth, and M. H. Beale, *"Neural Network Design"*, PWS Publishing, 1996.

[8] Nachirat Rachburee and Wattana Punlumjeck, *"Big Data Analytics: Feature Selection and Machine Learning for Intrusion Detection on Microsoft Azure Platform"*, Journal of Telecommunication, Electronic and Computer Engineering, April 2017.

[9] Navjit Singh, Anantdeep Kaur, M. Tech, *"Feature Selection for Artificial Neural Network Based Intrusion Detection System"*, International Journal for Technological Research in Engineering Volume 2, Issue 11, July 2015.

[10] Rachana Sharma, Priyanka Sharma, Preeti Mishra and Emmanuel S. Pilli, *"Towards MapReduce based classification approaches for Intrusion Detection*". 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence). January 2016.

[11] R. Gu, F. Shen, and Y.Huang, *"A parallel computing platform for training large scale neural networks",* IEEE International Conference on Big Data, October 2013, pp. 376–384.

[12] S.Rajeswari, R.Lawrance, *"Classification model To Predict the Learners' Academic Performance using Big Data"*, IEEE, 2016.