



Adaptive Exponential Bat algorithm and deep learning for big data classification

S MD MUJEEB^{1,*}, R PRAVEEN SAM² and K MADHAVI³

¹Department of CSE, Jawaharlal Nehru Technological University Anantapur (JNTUA), Ananthapuramu 515002, India

²Department of CSE, G Pulla Reddy Engineering College (Autonomous), Kurnool 518007, India

³Department of CSE, JNTUA College of Engineering Ananthapuramu, Ananthapuramu 515002, India
e-mail: mujeeb.smd@gmail.com; rpraveensam.cse@gprec.ac.in; kasamadhavi@yahoo.com

MS received 26 November 2019; revised 1 July 2020; accepted 13 September 2020

Abstract. The concept of big data refers to the huge amount of information that the organizations process, analyse and store. In the real-world scenario, some big data possess other features such as credit card fraud detection big data, extreme weather forecast big data and so on. In order to deal with the problem of classifying the binary imbalanced big data, based on MapReduce framework (MRF), an enhanced model is proposed for the process of classification in this paper. An optimization based on MRF is used for dealing with the imbalanced big data using the deep learning network for classification. The mappers in the MRF carry out the feature selection process with the proposed Adaptive E-Bat algorithm, which is a combination of adaptive, Exponential Weighted Moving Average (EWMA) and the Bat algorithm (BA) concepts. Using the features, the reducers perform the classification using Deep Belief Network (DBN) that is trained with the proposed Adaptive E-Bat algorithm. The performance of the proposed Adaptive E-Bat DBN method is evaluated in terms of metrics, namely accuracy and True Positive rate (TPR); a higher accuracy of 0.8998 and higher TPR of 0.9144 are obtained, that show the superiority of the proposed Adaptive E-Bat DBN method in effective big data classification.

Keywords. Big data; MapReduce framework; Exponential Weighted Moving Average; adaptive; Deep Belief Network.

1. Introduction

With the fast development of networking, data storage and the data collection capacity, big data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences [1, 2]. The increase in the speed of digital data storage, especially in online learning management system [3] and e-commerce domain, is one of the reasons for the increasing importance and rapid growth in the field of data mining [4]. Big data do not just imply large volumes of data but also the necessity for scalability, which is to ensure a response in an acceptable elapsed time [5]. Big data are usually processed in distributed environment with a number of connected machines supporting applications typically termed as Big Data Analytics [6, 7]. However, the amount of sensitive data typically processed in Big Data Analytics has made Big Data Analytics applications an eye catch to anomalous users [8].

Huge data need to be explored in an efficient way and are converted as valuable knowledge that is used by enterprises to develop their competitive advantage [1, 9]. However, there is a considerable gap among the contemporary processing and the storage capacities that demonstrate the ability to capture, store data and utilize it. Hence, dedicated tools and methods are needed to be developed to mine the enormous amount of incoming data while additionally considering that each record is analysed only once for the reduction of the overall computing costs [4]. MapReduce framework (MRF) was the initial programming paradigm developed to deal with the concept of big data [5]. In the recent years a new large-scale processing framework, known as Apache Spark [10, 11], is obtaining more importance in the big data domain because of its enhanced performance in both the iterative and incremental procedures. Lazy learning [12], termed as instance-based learning, is considered as the most simple and effective schemes in case of supervised learning [13]. The generalization is deferred till a query is developed to the case-base, but the distance among every pair of cases needs to be calculated. These methods seem to have very slower classification

*For correspondence
Published online: 03 February 2021

phase as compared with their counterparts. In addition the lazy learners, termed as k -nearest neighbour (k -NN), tend to accumulate the instances from data streams that make the use of data related to the outdated concepts to carry out the decision-making process [2, 14].

The classification of imbalanced datasets has puzzled most of the researchers, as the expected distribution of data cannot be obtained due to various reasons, specifically in case of cost-sensitive business scenarios. In case of the unbalanced distribution of data in the same sample space, some resampling methods are selected that sacrifice some of the features in the construction of relatively balanced training datasets. In addition the virtual samples are developed to balance the distribution of data that improves the recognition rate of the minority class termed as recall rate, but sacrifices the precision of the classification model [15]. MRF allows for automatically processing data in an easy and transparent way through a cluster of computers. The user only needs to implement two operators namely Map and Reduce. The neighbour query and the editing process are performed in the MRF process, thus reducing the communication overhead [16, 14]. The implementation of MRF runs on a large cluster of commodity machines and it is highly scalable [10, 11]. Consultation time is the time between when an object is presented to a system for an inference to be made and the time when the inference is completed [12]. The traditional focus on data mining problems can introduce advanced data types such as text, time series, discrete sequences, spatial data, graph data and social networks [13].

This paper proposes an optimization-based MRF to deal with the imbalanced data by adapting the deep learning concept in classification. The mappers in the MRF perform the feature selection process using the proposed Adaptive E-Bat algorithm, which modifies the update equation of the E-Bat algorithm by making it adaptive in order to handle the real-time data. Based on the selected features the reducers perform the classification using the Deep Belief Network (DBN), which is trained using the proposed Adaptive E-Bat algorithm.

Main contributions are as follows.

- Adaptive E-Bat algorithm: The Adaptive E-Bat algorithm is developed by integrating the adaptive concept, EWMA and Bat algorithm (BA).
- Adaptive E-Bat DBN: The DBN is trained by the proposed adaptive E-Bat algorithm, which guarantees the classification accuracy with the effective parallelism of servers, responsible for processing the subsets of big data.

The paper is organized as follows. The introduction to the need for big data classification is detailed in section 1 and section 2 details the literature review of the existing methods of big data classification with their drawbacks. In

section 3, the proposed Adaptive E-Bat DBN method of big data classification is presented and section 4 details the results of the proposed Adaptive E-Bat DNN method. Finally, section 5 concludes the paper.

2. Motivation

In this section a literature survey of various methods used for big data classification is presented, and the challenges of the existing methods are discussed.

2.1 Literature survey

Eight literature related to the filter design are discussed as follows. Ramírez-Gallego *et al* [14] design a method, known as Nearest Neighbour Classification, which is capable of handling high-dimensional scenarios, but the changes in drift that occur in the data are neglected during data classification. Zhai *et al* [17] develop a method, known as fuzzy-integral-based extreme learning machine (ELM) ensemble, which possesses a simple structure and is easy for implementation. The drawback of this method is regarding the failure in using for multi-classification of imbalanced data. Varatharajan *et al* [18] modelled a method, named as Linear Discriminant Analysis (LDA) with an enhanced Support Vector Machine (SVM) that used reduced data for classification with enhanced accuracy, but the usage of large data environments affected the performance. Elkano *et al* [19] designed a method, known as Fuzzy Rule-Based Classification Systems (FRBCSs), with increased accuracy, but could not address the problem of size-up. Singh *et al* [20] presented a Distribution Preserving Kernel Support Vector Machine (DiP-SVM) where both the first- and second-order statistics of the whole dataset were retained in all the partitions to obtain a minimal loss in classification accuracy. However, this method highly depended on the initial selected sets. Duan *et al* [21] developed an efficient ELM based on Spark framework (SELM), with the inclusion of three parallel sub-algorithms for classification of big data. This method possessed the highest speed-up with increased accuracy, but these algorithms required several copies for each task during MRF works. Chen *et al* [22] designed the Parallel Random Forest (PRF) algorithm for big data on the Apache Spark platform for the improvement of accuracy in case of large, high-dimensional and noisy data. However, these tasks needed only the data of current feature variable and target feature variable. Hababeh *et al* [23] developed an integrated methodology for classification to safeguard the big data before executing duplication, data mobility and analysis. This classification method possessed high performance by avoiding the redundant encryption and decryption pro-

cesses in case of public files. However, it imposed additional security overhead that decreased the performance mostly while transmitting a huge amount of data.

2.2 Challenges

The various challenges involved in this research are detailed here as follows:

- ELM has been proved to be an efficient and fast classification algorithm [24] with reduction in storage space. However, as the size of the training data increases the conventional ELM is not capable of providing efficient classification ability [21].
- The PELM algorithms based on MRF used in handling the big data classification undergo many map and reduce tasks at the stages. The intermediate results produced during the map stages are written in the disks, and during the reduce stages they are taken from the disks into a Hadoop distributed file system (HDFS). This process increases the communication cost and I/O overhead and in addition affects the efficiency and the learning speed of the system [21].
- The arrival time of datasets possesses uncertainty in the robust resource allocation of data processing on a heterogeneous parallel system [25, 22].
- Apache Spark MLlib [26] parallelized the Random Forest (RF) algorithm termed as Spark-MLRF based on a data parallel optimization for the improvement of the performance of the algorithm, but failed in terms of cost and accuracy [22].

3. Proposed Adaptive Exponential BAT algorithm for big data classification

Big data classification is normally used to provide effective analysis of big data that are generated from distributed sources. There are various algorithms that deal with the big data, but most of them fail due to processing complexity, processing the data of missing attributes and additional attributes that are new. In order to minimize the computational time and to deal with the distributed data, MRF is used. The big data obtained from the distributed sources are perfectly managed with the use of the MRF and the issues related to computation are rectified with effective feature selection criteria that operate with less number of features. The proposed adaptive E-Bat algorithm is developed for the classification of big data using the deep learning approach that guarantees classification accuracy with effective parallelism of servers that is responsible for processing the subsets of big data. The two important functions of MRF are map function and reduce function, which carry out the action of mapping the input data as relevant patterns and

reduce the intermediate data available in mappers to produce the desired output.

3.1 Principle of BA

The working principle of the BA [27, 28] is based on the behaviour of bat echolocation, which uses the echolocation or the SONAR to find the prey. The bats make use of the time delays developed by the received signals from its initial time, and the variation in loudness for sensing the surroundings. Bats possess a magical way for placing the objects and search for prey depending upon the changes in their velocity and position, and have the tendency to adjust the wavelength and pulse rates of the emitted pulses. This algorithm possesses better convergence rate at the initial stage and has the capability of switching among both the exploration and the exploitation phases while obtaining the optimal location. The automatic switching occurs due to variations in pulse emission rates and loudness while searching over the global solution and this algorithm has the ability to solve multi-modal optimization algorithms. The update equation of the BA is expressed as

$$B_x^{s+1} = B_x^s + k_x^s \quad (1)$$

where B_x^{s+1} indicates the x^{th} bat position in $(s+1)^{th}$ iteration, B_x^s indicates the x^{th} bat position in s^{th} iteration and k_x^s indicates the x^{th} bat velocity in s^{th} iteration. The velocity of the bat in s^{th} iteration is given as

$$k_x^s = k_x^{s+1} - (B_x^s - B_{bat}^*) \lambda_x. \quad (2)$$

The position of the bat in the $(s+1)^{th}$ iteration is

$$B_{x,bat}^{s+1} = B_x^s (1 - \lambda_x) + k_x^s + B_{bat}^* \times \lambda_x \quad (3)$$

where B_{bat}^* indicates the best position of the bat, λ_x is the frequency of the x^{th} bat and k_x^{s+1} indicates the velocity of the bat in $(s+1)^{th}$ iteration.

3.2 Principle of Exponential Weighted Moving Average (EWMA)

EWMA [29] averages the data to obtain very less weight to a data, which is removed with time. EWMA is a monitoring process and it is essential to know the present estimate of the variance rate and guarantees the governance of volatility of data. The equation of EWMA is expressed as

$$B_{EWMA}^s = \eta \times B_x^s + (1 - \eta) \times B_{EWMA}^{s-1} \quad (4)$$

$$B_x^s = \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \quad (5)$$

where B_{EWMA}^s indicates the current record, B_{EWMA}^{s-1} indicates the record of the previous iteration and η is a constant

between 0 and 1. Its current prediction depends on the historical values that are multiplied by a weight. Varying the sampling interval permits the information about the process to be produced more quickly, and hence prevents the continued production of poor results.

3.3 Development of the proposed adaptive E-Bat algorithm using the adaptive, BAT and EWMA principles

An adaptive algorithm is an algorithm that changes its behaviour at the time it runs, based on information available and on a priori-defined reward mechanism. The update rule of the proposed adaptive E-Bat algorithm is developed with the combination of the adaptive concept in E-Bat algorithm, where E-Bat is the integration of EWMA in BA. The convergence of the E-Bat algorithm is improved as the position update of bat relies on the position of the bat in previous iteration and the best position of the bat that was obtained so far, and is derived by substituting equation (5) in equation (3) as

$$B_{x,bat}^{s+1} = \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \times (1 - \lambda_x) + k_x^{s+1} + B_{bat}^* \times \lambda_x \tag{6}$$

From the velocity equation expressed in (2)

$$k_x^{s+1} - k_x^s = (B_x^s - B_{bat}^*) \lambda_x \tag{7}$$

$$\begin{aligned} (B_x^s - B_{bat}^*) \lambda_x &= \left(\frac{3}{2}\right)^\beta k_x^{s+1} - \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta}{3}\right) k_x^s \\ &+ \left(\frac{3}{2}\right)^\beta \left(\frac{\beta(8\beta - 5)}{9}\right) k_x^{s-1} \\ &+ \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta(\beta - 1)(7 - 8\beta)}{81}\right) k_x^{s-2} \end{aligned} \tag{8}$$

Subtract the term $B_x^s \lambda_x$ from equation (6) on both sides to include the adaptive concept in this equation:

$$\begin{aligned} B_{x,bat}^{s+1} - B_x^s \lambda_x &= \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \times (1 - \lambda_x) \\ &+ k_x^{s+1} - B_x^s \lambda_x + B_{bat}^* \times \lambda_x \end{aligned} \tag{9}$$

$$\begin{aligned} B_{x,bat}^{s+1} - B_x^s \lambda_x &= \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \times (1 - \lambda_x) \\ &+ k_x^{s+1} - (B_x^s - B_{bat}^*) \lambda_x. \end{aligned} \tag{10}$$

Substitute equation (8) in equation (10):

$$\begin{aligned} B_{x,bat}^{s+1} - B_x^s \lambda_x &= \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \times (1 - \lambda_x) + k_x^{s+1} \\ &- \left\{ \left(\frac{3}{2}\right)^\beta k_x^{s+1} - \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta}{3}\right) k_x^s + \left(\frac{3}{2}\right)^\beta \left(\frac{\beta(8\beta - 5)}{9}\right) k_x^{s-1} \right. \\ &\quad \left. + \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta(\beta - 1)(7 - 8\beta)}{81}\right) k_x^{s-2} \right\} \end{aligned} \tag{11}$$

$$\begin{aligned} B_{x,bat}^{s+1} &= B_x^s \lambda_x + \frac{1}{\eta} [B_{EWMA}^s - (1 - \eta) \times B_{EWMA}^{s-1}] \times (1 - \lambda_x) \\ &+ k_x^{s+1} - \left(\frac{3}{2}\right)^\beta k_x^{s+1} + \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta}{3}\right) k_x^s \\ &- \left(\frac{3}{2}\right)^\beta \left(\frac{\beta(8\beta - 5)}{9}\right) k_x^{s-1} \\ &- \left(\frac{3}{2}\right)^\beta \left(\frac{4\beta(\beta - 1)(7 - 8\beta)}{81}\right) k_x^{s-2} \end{aligned} \tag{12}$$

The solutions are updated using equation (12) that possesses the combination of adaptive concept, BA and EWMA concepts. This expression acts as the input to the MRF that undergoes in its phases, such as mapper and reducer, to produce the optimal centroids.

3.4 Algorithmic steps of proposed adaptive E-Bat algorithm

The algorithmic steps involved in the proposed adaptive E-Bat algorithm are detailed as follows.

a) *Initialization*: The first step of the proposed Adaptive E-Bat algorithm is the initialization of bat population in the search space as

$$\text{Bat population, } B_x ; (1 \leq x \leq a) \tag{13}$$

where a represents the total bats, and B_x represents the position of the x^{th} bat in the search space.

b) *Evaluate the fitness*: The fitness function depends on two factors, namely number of features and accuracy, and it is expected to solve the maximization function. The fitness function is expressed as

$$R = R_1 + R_2 \tag{14}$$

where R_1 is the fitness depending on number of features and R_2 is the fitness depending on accuracy. The value of R_1 must be minimum, which indicates the few number of relevant features to assure enhanced classification. The minimum fitness is responsible for maximum objective function by subtracting the relevant features from unity as expressed in equation (15). The fitness function R_2 depends on accuracy and it must be high for an effective method as expressed in equation (16):

$$R_1 = 1 - \frac{|r_{selected}|}{T} \tag{15}$$

$$R_2 = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

where T is the total features, TP represents the true positive, TN represents the true negative, FP represents the false positive and FN represents the false negative. The solution that obtains maximum fitness measure is chosen as the best solution. The solution encoding provides a representation of the solution produced by the proposed Adaptive E-Bat algorithm.

c) *Update of solution using proposed Adaptive E-Bat algorithm:* The update rule of the proposed adaptive E-Bat algorithm is developed with the combination of the adaptive, BA and the EWMA concepts. While updating the solution, there exist two conditions: the first based on pulse emission rate and the second based on loudness and fitness. In the first condition the random number is compared to the pulse emission rate, and if the random number exceeds the pulse emission rate the position is updated based on random walk. In the second condition, if the random number is less than the pulse emission

rate, it is compared to the fitness and the loudness of the bat. If the random number exceeds the loudness and fitness exceeds the fitness of best solution, then the new solution is updated using equation (12). After position update, the loudness decreases and the pulse emission rate increases for the successive iterations. The evaluation of loudness and emission rate for the x^{th} bat is expressed as

$$D_x^{s+1} = \tau D_x^s \tag{17}$$

$$E_x^s = E_x^0 [1 - \exp(-\omega s)] \tag{18}$$

where D_x^s represents the loudness in previous iteration, E_x^s is the emission rate and E_x^0 is the initial emission rate; τ and ω are constants.

d) *Ranking of bats on the basis of fitness:* The solutions are ranked based on the fitness and the solution with highest fitness measure is chosen as the best solution B^* .

e) *Termination:* The process is continued for the maximum number of iterations and stopped after the generation of global optimal solution. The pseudocode of the proposed Adaptive E-Bat algorithm is shown in algorithm 1.

Algorithm 1. Pseudocode of proposed Adaptive E-Bat algorithm.

Proposed Adaptive E-Bat algorithm	
1	Input : Population of bats $B_x ; (1 \leq x \leq a)$
2	Output : Best position of Bat B^*
3	Start
4	Initialization
5	For $\forall a$
6	Calculate the pulse emission rate and the loudness
7	While ($s < \text{Maximum iterations}$)
8	Produce new solutions using equation (12)
9	If ($Rand > E_x$)
10	{
11	Update position based on random walk
12	}
13	End if
14	Generate new solutions based on random fly
15	If ($Rand < D_x \ \&\& \ Fitness(B_x) < Fitness(B^*)$)
16	{
17	Update the bat position
18	Increment pulse emission rate E_x decrement loudness D_x
19	}
20	End if
21	Rank the solution based on fitness
22	Estimate the current best solution
23	End While
24	Produce the best solution B_{bat}^*
25	Stop

3.5 MRF in big data classification using the proposed adaptive E-Bat algorithm

Big data classification is used to perform effective analysis of big data that are produced using various sources. There exist a lot of algorithms that perform big data classification, but fail to satisfy certain requirements, such as processing the data of missing attributes and processing of additional attributes that are produced newly. For the minimization of computational time and to deal with distributed data the MRF is used, and the big data produced from various sources are managed perfectly using MRF. The proposed classification criteria provide the classification accuracy with effective parallelism of servers involved in the processing of subsets of big data. The two important functions of MRF are map function and reduce function, which perform the mapping of input data as relevant patterns and reduce the intermediate data present in the mappers to generate the desired output. Figure 1 depicts the block diagram of proposed big data classification strategy.

3.5a Adaptive E-Bat algorithm for the selection of features in the Mapper module of MRF

The MRF possesses increased power of processing due to the presence of a number of servers in the mapper phase that operate in parallel. The processing time and the flexibility of the big data are enhanced with the use of the MRF in such a way that the input big data are divided into various subsets of data and each individual mapper processes a subset to produce the desired output. The structure of MRF is as shown in figure 2.

Consider an input big data represented as I with number of attributes expressed as

$$I = \{a_{ij}\}; (1 \leq i \leq P); (1 \leq j \leq l) \quad (19)$$

where a_{ij} indicates the data of big data I representing j^{th} attribute of i^{th} data. There are P number of data points and l number of attributes for all the data points.

a) Mapper phase

The important function of the mapper phase is the extraction of features, where the highly relevant features are selected to perform dimensional reduction for the maximization of classification accuracy. The mapper phase uses the mapper function, and the adaptive E-Bat algorithm. The selected attribute of the data point is marked as '1', and the unselected attribute is marked as '0', which implies that the features are mapped as binary. In the initial step the big data are divided as sub-sets of data, which is expressed as

$$a_{ij} = \{I_g\}; (1 \leq g \leq T) \quad (20)$$

where T represents the total sub-sets of data developed from the big data. The total sub-sets of data is equal to that of the total number mappers in the mapper phase and is expressed as

$$K = \{K_1, K_2, \dots, K_g, \dots, K_T\}. \quad (21)$$

The input to the g^{th} mapper is expressed as

$$I_g = \{M_{c,l}\}; (1 \leq c \leq W); (1 \leq l \leq L) \quad (22)$$

where L indicates the total attributes and W indicates the total data points of g^{th} sub-set in such a way that $W < P$. The mapper uses the proposed Adaptive E-Bat algorithm for the selection of optimal features. The outputs obtained from all the mappers are framed together to develop the intermediate data, represented as G . The intermediate data acts as the input to the reducer phase, which is processed for the generation of the expected output using the proposed Adaptive E-Bat algorithm. For the proposed system to provide enhanced performance the solution must obtain maximum fitness measure and such a solution is selected as the best solution. The fitness function depends on two factors: number of features and accuracy, as expressed in equation (14). The features chosen by the proposed Adaptive E-Bat algorithm is expressed as

$$h = \{h_1, h_2, \dots, h_y, \dots, h_m\} \quad (23)$$

where m represents the total number of reduced features. Thus, the solution vector is a vector of selected features in case of feature selection with the feature size of $[1 \times m]$.

b) Reducer phase

The selected feature undergoes the classification process performed using the DBN network. The role of DBN [30] is the extraction and the recognition of the patterns present in the data sequences. The developed DBNs are trained with the labelled data for the maximization of fitness, and thus to predict the data with the outputs of the past or previously recorded outputs to provide enhanced accuracy in prediction. The structure of DBN comprises more than one Restricted Boltzmann Machine (RBM) and one Multi-Layer Perceptron (MLP) layer. Each layer of RBM and MLP represents the architecture of NN and the layers are developed with the interconnection of neurons. In incremental DBN, two RBMs are taken into consideration and the input to the RBM1 is the feature vector corresponding to the reduced features. The inputs are multiplied with the weights of input neuron to generate the output of the hidden layer that produces the input to the RBM2. The inputs in RBM2 are processed with the hidden weights of RBM2 to generate the input of MLP layer that processes the weights and produces the final output. The weights of DBN are estimated using the proposed Adaptive E-Bat algorithm, and the structure of incremental DBN is as shown in figure 3.

Consider that there are two RBMs, namely RBM1 and RBM2, and the input to RBM1 is the feature vector obtained from the big data. The input and hidden neurons in the input layer of RBM1 are expressed as

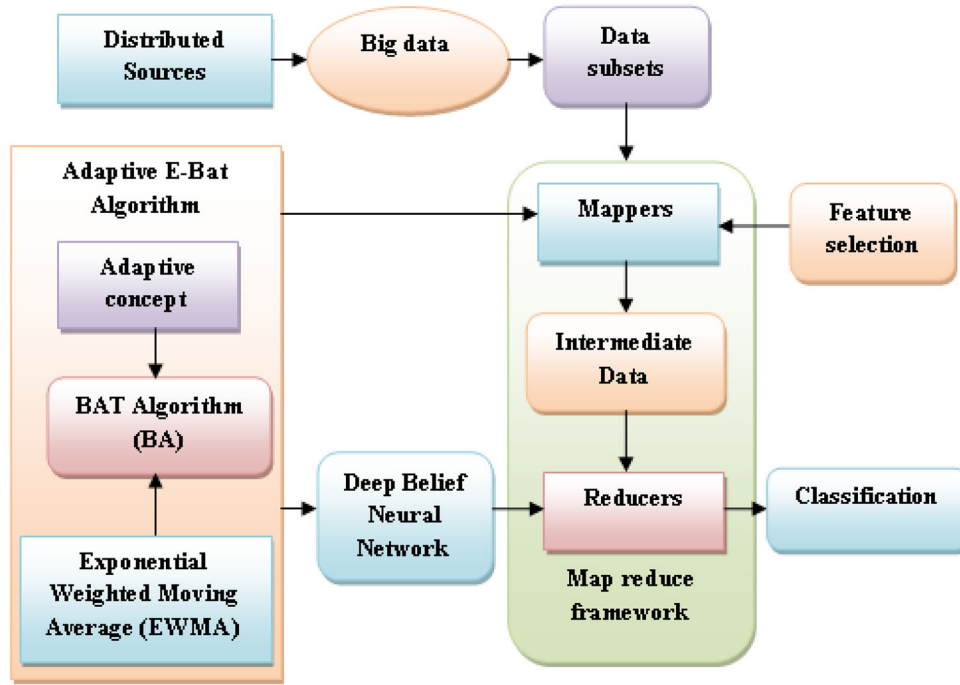


Figure 1. Block diagram of proposed big data classification technique.

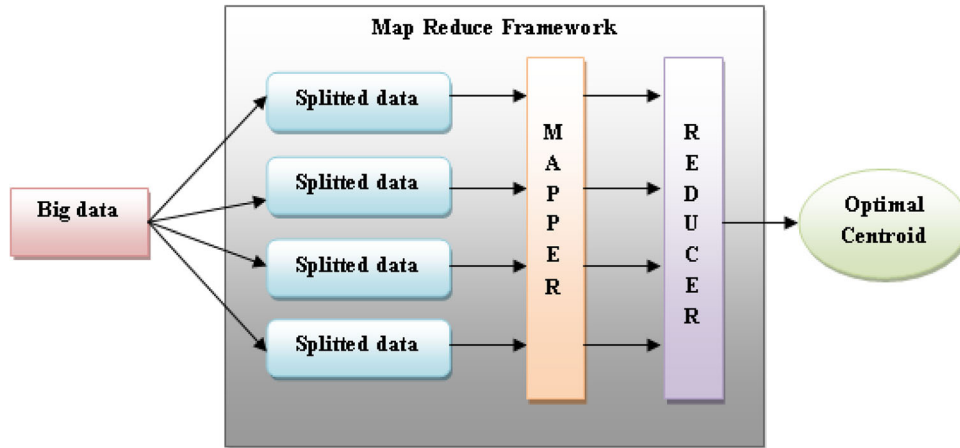


Figure 2. Structure of Map Reduce Framework.

$$A^1 = \{A_1^1, A_2^1, A_3^1, \dots, A_m^1, \dots, A_n^1\}; \quad 1 \leq m \leq n \quad (24)$$

$$C^1 = \{C_1^1, C_2^1, \dots, C_d^1, \dots, C_e^1\}; \quad 1 \leq d \leq e \quad (25)$$

where A_m^1 represents the m^{th} input neuron that is present in RBM1 and the count of input neurons of RBM1 is equal to the dimension of feature vector. There are n number of neurons in the input layer of RBM1 to perform classification. Let the total number of the hidden neurons in the RBM1 be e and let d^{th} hidden neuron in RBM2 be C_d^1 . The

biases of visible and hidden neurons of RBM1 are expressed as

$$N^1 = \{N_1^1, N_2^1, N_3^1, \dots, N_m^1, \dots, N_n^1\} \quad (26)$$

$$F^1 = \{F_1^1, F_2^1, \dots, F_d^1, \dots, F_e^1\}. \quad (27)$$

The biases of hidden and input layer of RBM1 are equal to that of the total neurons in both the layers and the weights of RBM1 are expressed as

$$u^1 = \{u_{md}^1\}; 1 \leq m \leq n; 1 \leq d \leq e \quad (28)$$

where u_{md}^1 represents the weights of RBM1 and it is the weight among m^{th} input neuron and d^{th} hidden neuron of RBM1. The dimension of weights is expressed as $(n \times e)$. Hence, the output of RBM1 is expressed as

$$C_d^1 = \vartheta \left[F_d^1 + \sum_m h_m^1 u_{md}^1 \right] \quad (29)$$

where ϑ represents the activation function in RBM1 and h_m^1 represents the feature vector as in equation (23). The output of the RBM1 is expressed as

$$C^1 = \{C_d^1\}; 1 \leq d \leq e. \quad (30)$$

The output from RBM1 is fed as the input to RBM2 and the output of RBM2 is estimated using these equations. The output of RBM2 is indicated as G_d^2 , which is fed as the input to the MLP layer. The input neurons in MLP are expressed as

$$Q^w = \{Q_1^w, Q_2^w, \dots, Q_d^w, \dots, Q_e^w\} = \{S_d^2\}; 1 \leq d \leq e \quad (31)$$

where y indicates the total input neurons in MLP layer. The hidden neurons of MLP are expressed as

$$U^w = \{U_1^w, U_2^w, \dots, U_t^w, \dots, U_v^w\}; 1 \leq t \leq v \quad (32)$$

where m represents the total hidden neurons of the MLP. The bias of the hidden neurons is expressed as

$$X^w = \{X_1^w, X_2^w, \dots, X_z^w, \dots, X_o^w\}; 1 \leq z \leq o \quad (33)$$

where S represents the output neurons in MLP layer. The weights among the input and the hidden layers are given as

$$u^{mlp} = \{u_{dt}^{mlp}\}; 1 \leq d \leq e; 1 \leq t \leq v \quad (28)$$

where u_{dt}^{mlp} represents the weight vector among d^{th} input neuron and the t^{th} hidden neuron. The output of the hidden layer in MLP is based on the bias and weights and is expressed as

$$Y = \left[\sum_{d=1}^e u_{dt}^{mlp} \times Z_d \right] u_t^w \quad \forall Z_d = C_d^2 \quad (34)$$

where u_t^w indicates the bias of output layer. The weight vector among the hidden and output layers is represented as χ^C and is expressed as

$$u^C = \{u_{tz}^C\}; 1 \leq t \leq v; 1 \leq z \leq o. \quad (35)$$

Hence, the output of MLP is estimated as

$$X_z = \sum_{t=1}^v u_{tz}^C \times Y \quad (36)$$

where u_{tz}^C represents the weights among the hidden and output neurons in MLP, and Y is the output of the hidden layer.

- i) *Training of RBM layers:* The training of RBM1 and RBM2 is carried out based on the proposed Adaptive E-Bat algorithm, which estimates the weights based on maximum fitness.
- ii) *Training of MLP layer:* The steps involved in the training of MLP layer are detailed as here.a)
 - a) Produce the weight vectors u^C and u^{mlp} in random as expressed in equations (36) and (34), respectively.
 - b) Read the input vector C_d^2 that is obtained from the output layer of RBM2.
 - c) Estimate the values of Y and X_z using equations (35) and (37), respectively.
 - d) Calculate the error of MLP layer with the estimated and target output as follows:

$$\varpi_{avg}^1 = \frac{1}{\psi} \sum_{H=1}^{\psi} (X_z - O)^2 \quad (37)$$

where X_z indicates the attained output, O represents the expected output and ψ is the total training samples.

- f) Estimate the average error function ϖ_{avg}^1 using the weight vector that is updated with the proposed Adaptive E-Bat algorithm.
- g) Repeat steps 2–6 till the best weight vector is obtained.

4. Results and discussion

This section details the results of the proposed Adaptive E-Bat DBN in big data classification and the comparative analysis involving the existing methods of big data classification is discussed.

4.1 Experimental set-up

The experimentation of the proposed Adaptive E-Bat DBN method is performed in JAVA that runs in a PC with Windows 8 OS. Table 1 shows the list of parameters used for the experimentation.

4.2 Dataset description

The analysis is carried out using six standard datasets, which are taken from UCI machine repository. The

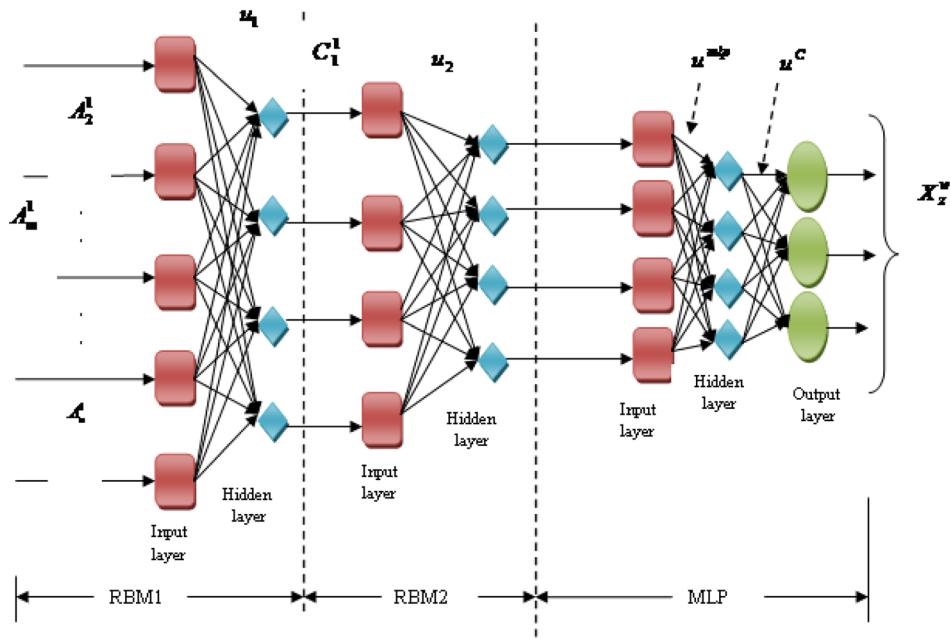


Figure 3. Structure of incremental Deep Belief Neural Network model.

standard datasets include breast cancer [31], Hepatitis [32], Pima Indian diabetes dataset [33], Heart disease dataset [34], Poker Hand dataset [35] and SUSY dataset [36].

4.2a Breast cancer dataset: The breast cancer dataset comprises 9 attributes of the categorical characteristics with the total of 286 instances.

4.2b Heart disease dataset: The heart disease database consists of databases such as Cleveland, Hungary, Switzerland and VA long beach. Among the available four databases, three databases, namely Cleveland, Hungary and Switzerland, are employed for experimentation. The nature of the database and its attributes are characterized as categorical, real, multivariate and integer with a total of 303 instances and 75 attributes.

4.2c Hepatitis dataset: The Hepatitis dataset comprises 155 instances and 19 attributes, with real, categorical and integer characteristics. The nature of the dataset is multivariate.

4.2d Pima Indian diabetes dataset: The Pima Indian diabetes database comprises 768 instances and 8 plus class attributes.

4.2e Poker Hand dataset: The Poker Hand dataset comprises 1025010 instances and 11 attributes, with categorical and integer characteristics. The nature of the dataset is multivariate.

4.2f SUSY dataset: The SUSY dataset comprises 5000000 instances and 18 attributes, with real characteristics. The nature of the dataset is N/A.

4.3 Comparative methods

The proposed Adaptive E-Bat method of big data classification is compared with the existing methods of big data classification, including Fuzzy [19], *K*-Nearest Neighbours (KNN) [14], SVM [18], BA + Neural Network (BAT+NN) [37], E-BatNN and Adaptive E-Bat DBN, to prove the superiority of the proposed method. BAT+NN is developed through the combination of BAT to update weights of NN, and E-BatNN is the integration of EWMA and the BA for updating the weights of NN.

4.4 Performance metrics

The metrics used for the performance analysis of the proposed Adaptive E-Bat DBN are the accuracy and True Positive rate (TPR) and are expressed as follows.

4.4a Accuracy: The term accuracy is defined as the rate of accurate classification of big data and is expressed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (33)$$

where *TP* represents the true positive, *TN* indicates the true negative, *FP* represents the false positive and *FN* is the false negative.

4.4b TPR: TPR is defined as the ratio of true positive to the total real positives present in the data, given as

$$TPR = \frac{TP}{P}. \quad (34)$$

Table 1. Parameters.

Parameters	Values
Number of hidden layers	3
Hidden layer 1-neurons	50
Hidden layer 2-neurons	25
Hidden layer 3-neurons	25
Activation function	Sigmoid
Learning rate	0.2

4.5 Comparative analysis

This section details the comparative analysis of the proposed Adaptive E-Bat DBN method of big data classification based on the performance metrics, such as accuracy and TPR.

4.5a Analysis using breast cancer dataset: The comparative analysis of the big data classification methods using the breast cancer dataset is depicted in figure 4. Figure 4a shows the accuracy of the methods for various training percentages based on breast cancer dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0086, 0.4964, 0.7752, 0.7758,

0.8762 and 0.8783, respectively. Figure 4b shows the TPR of the methods for various training percentages based on breast cancer dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0143, 0.1333, 0.4, 0.7949, 0.7975 and 0.8188, respectively. Basically, the TPR decreases with the increase in training percentage; however, the proposed EBatNN method obtained a better TPR as compared with the existing methods.

4.5b Using Cleveland dataset: The comparative analysis of the big data classification methods using the Cleveland dataset is depicted in figure 5. Figure 5a shows the accuracy of the methods for various training percentages based on Cleveland dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.4632, 0.5526, 0.5574, 0.5587, 0.7203 and 0.7891, respectively. Figure 5b shows the TPR of the methods for various training percentages based on Cleveland dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.5574, 0.5645, 0.6222, 0.6346, 0.791 and 0.8838, respectively.

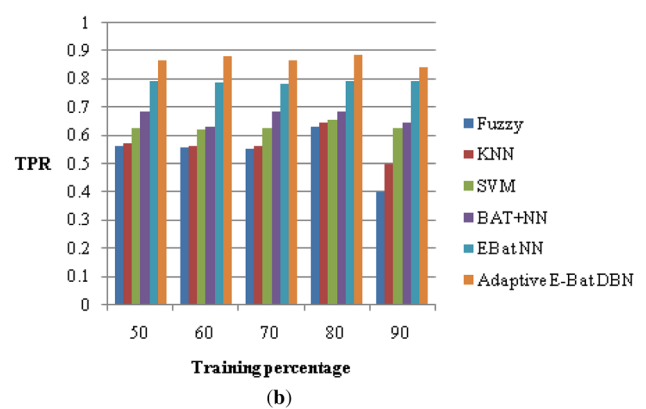
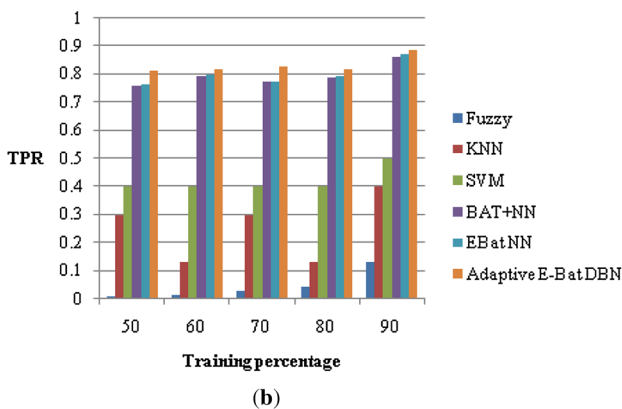
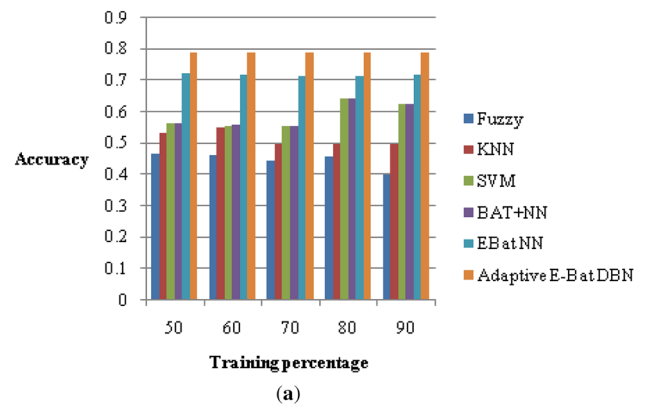
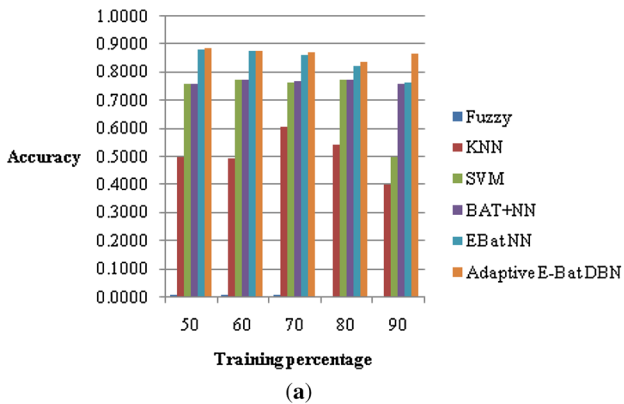


Figure 4. Analysis using breast cancer dataset based on a) accuracy and b) TPR.

Figure 5. Analysis using Cleveland dataset based on a) accuracy and b) TPR.

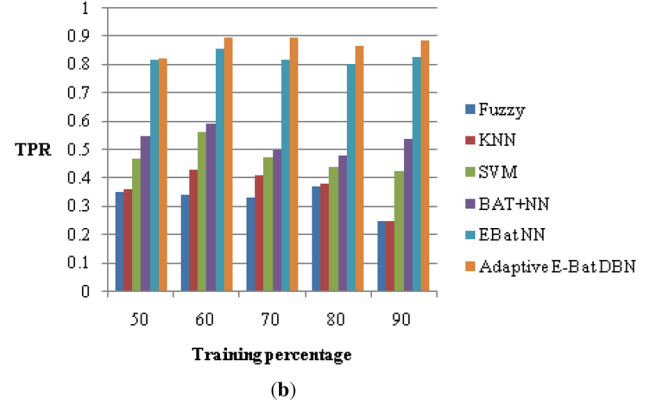
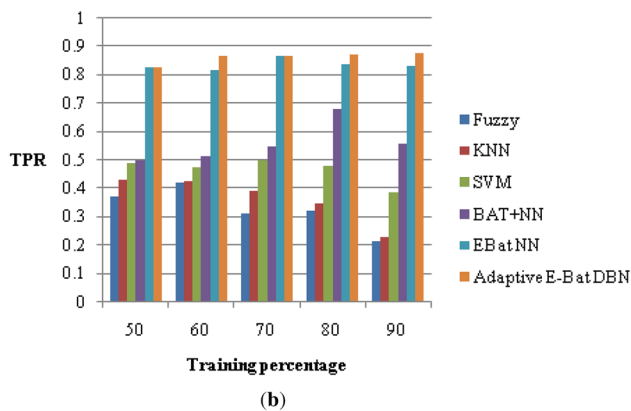
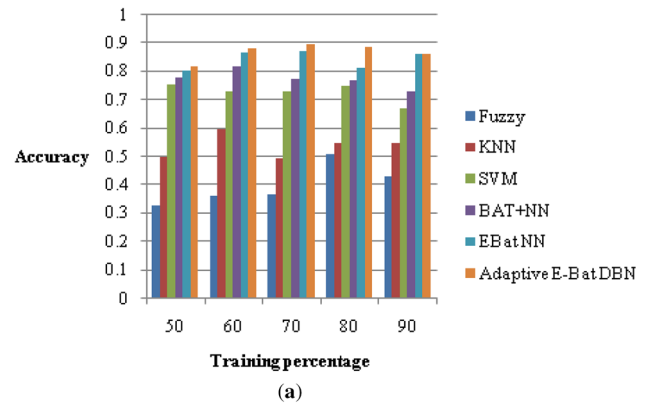
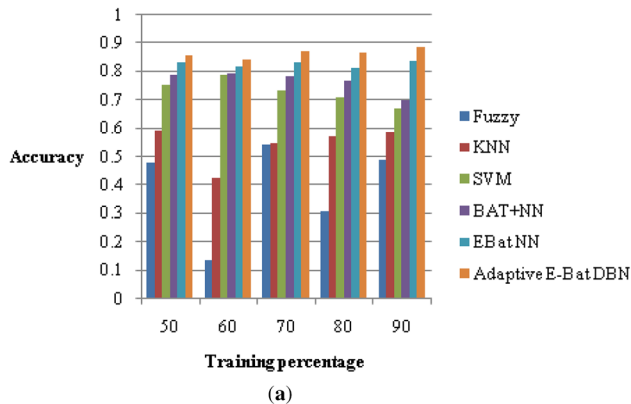


Figure 6. Analysis using Hungary dataset based on a) accuracy and b) TPR.

Figure 7. Analysis using Switzerland dataset based on a) accuracy and b) TPR.

4.5c Using Hungary dataset: The comparative analysis of the big data classification methods using the Hungary dataset is depicted in figure 6. Figure 6a shows the accuracy of the methods for various training percentages based on Hungary dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.1371, 0.427, 0.791, 0.7935, 0.82 and 0.8409, respectively. Figure 6b shows the TPR of the methods for various training percentages based on Hungary dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.4231, 0.4242, 0.475, 0.5135, 0.8202 and 0.8664, respectively.

4.5d Using Switzerland dataset: The comparative analysis of the big data classification methods using the Switzerland dataset is depicted in figure 7. Figure 7a shows the accuracy of the methods for various training percentages based on Switzerland dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.3623, 0.6, 0.7295, 0.8178, 0.8666 and 0.8831, respectively. Figure 7b shows the TPR of the methods for various training percentages based on Switzerland dataset. When the training percentage is 60, the TPR of the methods, such

as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.3455, 0.4324, 0.566, 0.5946, 0.859 and 0.8959, respectively.

4.5e Using Hepatitis dataset: The comparative analysis of the big data classification methods using the Hepatitis dataset is depicted in figure 8. Figure 8a shows the accuracy of the methods for various training percentages based on Hepatitis dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0635, 0.0775, 0.4902, 0.5667, 0.6543 and 0.7893, respectively. Figure 8b shows the TPR of the methods for various training percentages based on Hepatitis dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0635, 0.0923, 0.5809, 0.6333, 0.6977 and 0.7882, respectively.

4.5f Using Pima India diabetes dataset: The comparative analysis of the big data classification methods using the Pima India dataset is depicted in figure 9. Figure 9a shows the accuracy of the methods for various training percentages based on Pima India dataset. When the training percentage is 60, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.071, 0.4777, 0.5072, 0.7176, 0.7179 and

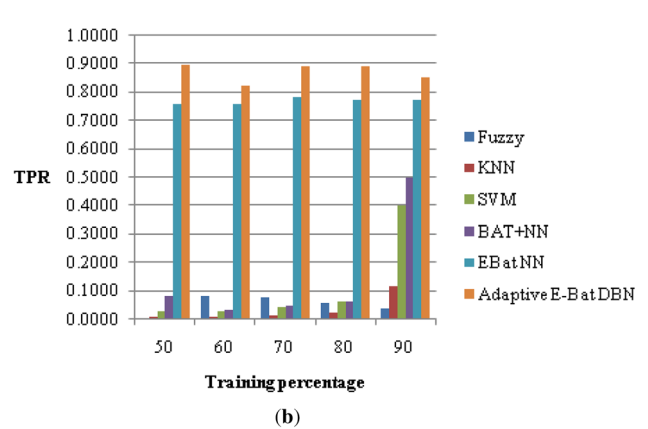
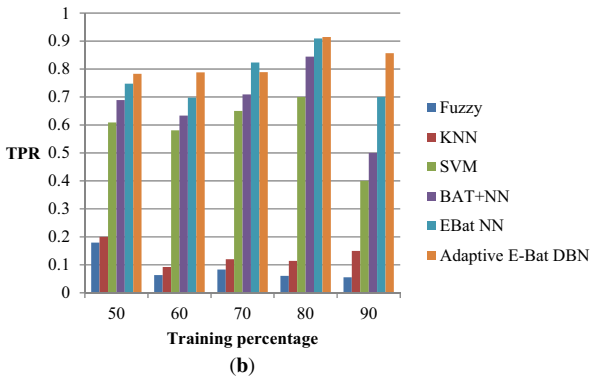
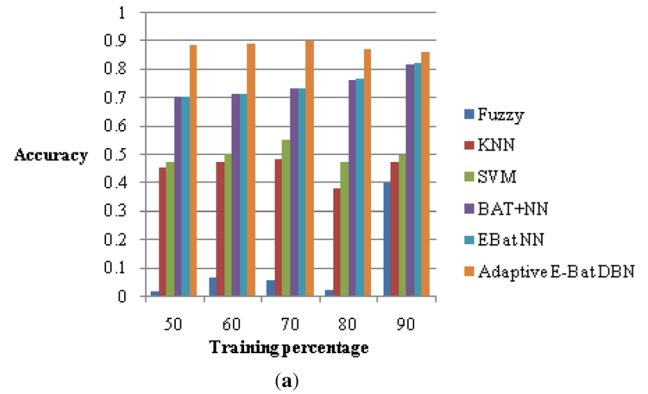
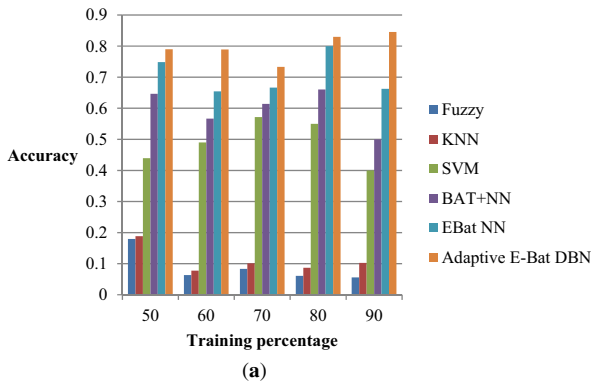


Figure 8. Analysis using Hepatitis dataset based on a) accuracy and b) TPR.

Figure 9. Analysis using Pima India dataset based on a) accuracy and b) TPR.

0.8906, respectively. Figure 9b shows the TPR of the methods for various training percentages based on Pima India dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0851, 0.0115, 0.0294, 0.0337, 0.76 and 0.8216, respectively.

4.5g Using Poker Hand dataset: The comparative analysis of the big data classification methods using the Poker Hand dataset is depicted in figure 10. Figure 10a shows the accuracy of the methods for various training percentages based on Poker Hand dataset. When the training percentage is 70, the accuracy of the methods, such as Fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.1281, 0.24, 0.3348, 0.4967, 0.8141 and 0.8843, respectively. Figure 10b shows the TPR of the methods for various training percentages based on Poker Hand dataset. When the training percentage is 70, the TPR of the methods, such as Fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0618, 0.0618, 0.07, 0.4961, 0.8288 and 0.8493, respectively.

4.5h Using SUSY dataset: The comparative analysis of the big data classification methods using the SUSY dataset is depicted in figure 11. Figure 11a shows the accuracy of the methods for various training percentages based on SUSY dataset. When the training percentage is 60, the accuracy of the methods, such as Fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.0856, 0.2,

0.3, 0.4444, 0.8236 and 0.8482, respectively. Figure 11b shows the TPR of the methods for various training percentages based on SUSY dataset. When the training percentage is 60, the TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and proposed Adaptive E-Bat DBN, is 0.1499, 0.1499, 0.02856, 0.4, 0.8104 and 0.8873, respectively.

4.6 Comparative discussion

Table 2 shows the comparative analysis of the proposed Adaptive E-Bat DBN method and the existing methods in terms of accuracy and TPR. The accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the breast cancer dataset is 0.0086, 0.5, 0.7596, 0.76, 0.8829 and 0.8873, respectively. The TPR of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the breast cancer dataset is 0.133, 0.4, 0.5, 0.865, 0.872 and 0.888, respectively. Similarly, the accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the Cleveland dataset is 0.446, 0.5, 0.554, 0.556, 0.713 and 0.79, respectively. The TPR of methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the

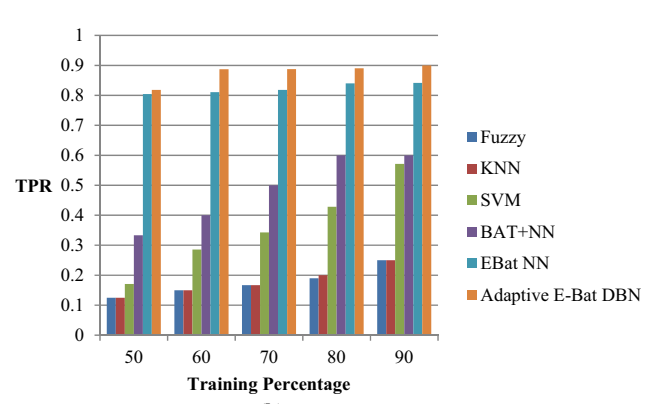
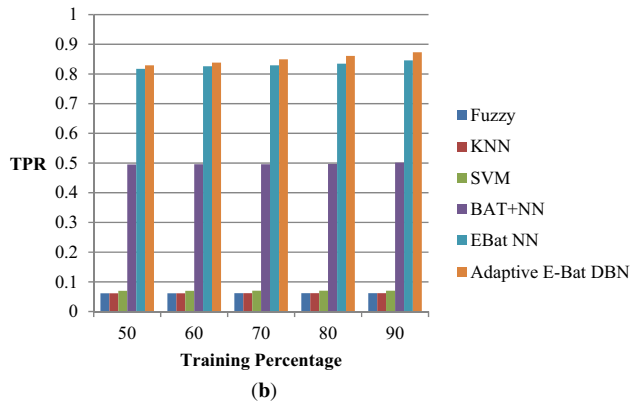
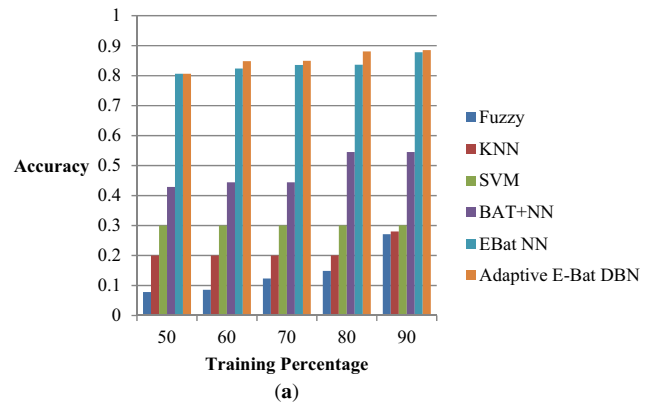
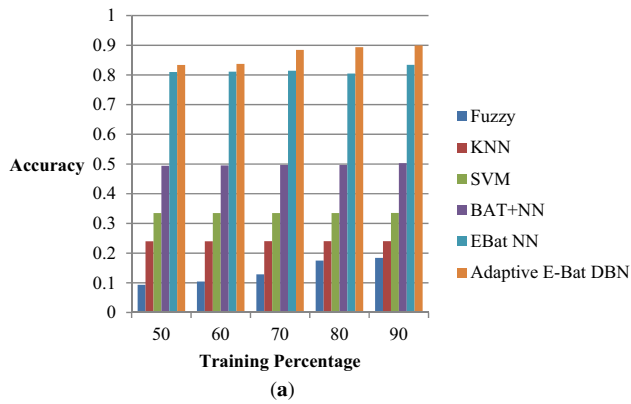


Figure 10. Analysis using Poker Hand dataset based on a) accuracy and b) TPR.

Figure 11. Analysis using SUSY dataset based on a) accuracy and b) TPR.

proposed Adaptive E-Bat DBN, using the Cleveland dataset is 0.6308, 0.6452, 0.6563, 0.6842, 0.7955 and 0.8854, respectively. The accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the Hungary dataset is 0.4879, 0.5889, 0.6719, 0.7015, 0.8362 and 0.8887, respectively. The TPR of methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the Hungary dataset is 0.2143, 0.2308, 0.3889, 0.5577, 0.8336 and 0.879, respectively. The accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the Poker Hand dataset is 0.1838, 0.2400, 0.3351, 0.5029, 0.8340 and 0.8993, respectively. The TPR of methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the Poker Hand dataset is 0.0619, 0.0619, 0.0701, 0.5014, 0.8457 and 0.8728, respectively. The accuracy of the methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the SUSY dataset is 0.2712, 0.2802, 0.3007, 0.5454, 0.8780 and 0.8850, respectively. The TPR of methods, such as fuzzy, KNN, SVM, BAT+NN, EBatNN and the proposed Adaptive E-Bat DBN, using the SUSY dataset is 0.2498, 0.2498, 0.5709, 0.6, 0.8414 and 0.8998, respectively. From the analysis, it is clear that the accuracy and TPR of the proposed Adaptive E-Bat DBN method are

better as compared with the conventional methods of big data classification.

5. Conclusion

Big Data is a collection of large amount of data that becomes tedious to be processed using the traditional methods of data processing. In other words, a dataset is named as Big Data when it is tedious to store, process and visualize the data using the state-of-art methods. In this paper, an optimization-dependent MRF is utilized to manage the imbalanced data using the concept of deep learning in classification. The mappers in the MRF perform the feature selection using the proposed Adaptive E-Bat algorithm, which is the hybridization of EWMA and BA integrated with adaptive concept. With the selected features, the reducers perform the process of classification using DBN that is trained using the proposed Adaptive E-Bat algorithm. The analysis of the proposed method is performed in terms of metrics, such as accuracy and True Positive rate (TPR). The proposed method obtained an increased accuracy and TPR of 0.8998 and 0.9144, respectively, which is high as compared with the existing methods. In future, this method will be extended to deal

Table 2. Comparative discussion involving the methods of big data classification.

Datasets	Metrics	Methods					
		Fuzzy	KNN	SVM	BAT+NN	EBatNN	Proposed Adaptive E-Bat
Breast cancer	Accuracy	0.0086	0.5000	0.7596	0.7600	0.8829	0.8873
	TPR	0.133	0.400	0.500	0.865	0.872	0.888
Cleveland	Accuracy	0.446	0.500	0.554	0.556	0.713	0.790
	TPR	0.6308	0.6452	0.6563	0.6842	0.7955	0.8854
Hungary	Accuracy	0.4879	0.5889	0.6719	0.7015	0.8362	0.8887
	TPR	0.2143	0.2308	0.3889	0.5577	0.8336	0.8790
Switzerland	Accuracy	0.3680	0.4944	0.7283	0.7754	0.8722	0.8990
	TPR	0.3333	0.4138	0.4750	0.5000	0.8195	0.8983
Hepatitis	Accuracy	0.0556	0.1026	0.4000	0.5000	0.6626	0.8457
	TPR	0.0606	0.1143	0.7000	0.8444	0.9091	0.9144
Pima Indian diabetes	Accuracy	0.0574	0.4870	0.5532	0.7340	0.7345	0.8998
	TPR	0.0061	0.0093	0.0273	0.0851	0.7595	0.8992
Poker Hand dataset	Accuracy	0.1838	0.2400	0.3351	0.5029	0.8340	0.8993
	TPR	0.0619	0.0619	0.0701	0.5014	0.8457	0.8728
SUSY dataset	Accuracy	0.2712	0.2802	0.3007	0.5454	0.8780	0.8850
	TPR	0.2498	0.2498	0.5709	0.6	0.8414	0.8998

with the security constraints during the classification of big data.

References

- [1] Wu X, Zhu X, Wu G Q and Ding W 2014 Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering* 26(1): 97–107
- [2] H. Karau, A. Konwinski, P. Wendell and M. Zaharia 2015 *Learning Spark: lightning-fast Big Data Analytics*.
- [3] Ekhoor-Top learning management system from <https://ekhoor.com/> (2016)
- [4] U. Fayyad and R. Uthurusamy 2002 Evolving data into mining solutions for insights. *Communications of Computers in Entertainment* 45(8): 28–31
- [5] A. Fernández *et al* 2014 Big data with cloud computing: an insight on the computing environment, MapReduce, and programming frameworks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4(5): 380–409
- [6] Mazumder, S., Bhadoria, R.S. and Deka, G.C 2017 *Distributed computing in big data analytics (Scalable computing and communications)*
- [7] Swarnkar, M. and Bhadoria, R.S 2017 Security issues and challenges in big data analytics in distributed environment. In: *Distributed computing in big data analytics*, pp. 83–94
- [8] Sharma, U. and Bhadoria, R.S 2016 Supportive architectural analysis for big data. In: *The human element of big data*. Chapman and Hall/CRC, pp. 137–154
- [9] J. Gama 2010 *Knowledge discovery from data streams*.
- [10] J. Dean and S. Ghemawat 2004 MapReduce: simplified data processing on large clusters. In: *Proceedings of OSDI*, pp. 137–150
- [11] Mayer-Schönberger V and K. Cukier 2013 *Big data: a revolution that will transform how we live”, work and think*
- [12] D. Aha 1997 *Lazy learning*. Dordrecht, The Netherlands: Kluwer
- [13] C. C. Aggarwal 2015 *Data mining: the textbook*. Cham, Switzerland: Springer
- [14] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, J. M. Benítez and F. Herrera 2017 Nearest neighbor classification for high-speed big data streams using Spark. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(10): 2727–2739
- [15] Wu Z, Lin W, Zhang Z, Wen A and Lin L 2017 An ensemble random forest algorithm for insurance big data analysis. In: *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*
- [16] D. Han, C. G. Giraud-Carrier and S. Li 2015 Efficient mining of high-speed uncertain data streams. *Applied Intelligence* 43(4): 773–785
- [17] Zhai, J., Zhang, S., Zhang, M. and Liu, X 2018 Fuzzy integral-based ELM ensemble for imbalanced big data classification. *Soft Computing* 22: 3519–3531
- [18] R. Varatharajan, Manogaran G and Priyan M K 2018 A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing. *Multimedia Tools and Applications* 77: 10195–10215
- [19] Elkano M, Galar M, Sanz J and Bustince H 2018 CHI-BD: a fuzzy rule-based classification system for Big Data classification problems. *Fuzzy Sets and Systems* 348: 75–101
- [20] Singh D, Roy D and Krishna Mohan C 2017 DiP-SVM: Distribution preserving kernel support vector machine for Big Data. *IEEE Transactions on Big Data* 3(1): 79–90
- [21] Duan M, Li K, Liao X and Li K 2018 A parallel multiclassification algorithm for big data using an extreme

- learning machine. *IEEE Transactions on Neural Networks and Learning Systems* 29(6): 2337–2351
- [22] Chen J, Li K, Zhuo Tang S, Bilal K, Yu S, Weng C and Li K 2017 A parallel random forest algorithm for big data in a spark cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems* 28(4): 919–933
- [23] Hababeh I, Gharaibeh A, Nofal S and Khalil I 2018 An integrated methodology for big data classification and security for improving cloud systems data mobility. *IEEE Access* 7: 9153–9163
- [24] Y. Yang, Y.Wang and X. Yuan 2012 Bidirectional extreme learning machine for regression problem and its learning effectiveness. *IEEE Transactions on Neural Networks and Learning Systems* 23(9): 1498–1505
- [25] L. D. Briceno, H. J. Siegel, A. A. Maciejewski, M. Oltikar, and J. Brateman 2011 Heuristics for robust resource allocation of satellite weather data processing on a heterogeneous parallel system. *IEEE Transactions on Parallel and Distributed Systems* 22(11): 1780–1787
- [26] A. Spark June 2016 *Spark MLlib – random forest*. Website
- [27] Yang, X.S 2011 Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation* 3(5): 267–274
- [28] Fister I, Fong S and Brest J 2014 A Novel Hybrid Self-Adaptive Bat Algorithm., *The Scientific World Journal* (Recent Advances in Information Technology) 2014 <https://doi.org/10.1155/2014/709738>
- [29] Saccucci M S, Amin R W and Lucas J M 1992 Exponentially weighted moving average control schemes with variable sampling intervals. *Communications in Statistics – Simulation and Computation* 21(3): 627–657
- [30] Vojt B J 2016 *Deep neural networks and their implementation.*, Master Thesis, Department of Theoretical Computer Science and Mathematical Logic, Prague
- [31] *Breast cancer dataset*. <http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29> (accessed in March 2018)
- [32] *Hepatitis dataset*. <https://archive.ics.uci.edu/ml/datasets/hepatitis> (accessed in March 2018)
- [33] *Pima Indian diabetes dataset*. <https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes> (accessed in March 2018)
- [34] *Heart disease dataset*. <http://archive.ics.uci.edu/ml/datasets/heart+disease> (accessed in March 2018)
- [35] *Poker hand data set*. <https://archive.ics.uci.edu/ml/datasets/Poker+Hand> (accessed in 2002)
- [36] SUSY data set. <https://archive.ics.uci.edu/ml/datasets/SUSY#> (accessed in July 2014)
- [37] H. Ke, D. Chen, X. Li, Y. Tang, T. Shah and R. Ranjan Towards brain big data classification: epileptic EEG identification with a lightweight VGGNet on global MIC. *IEEE Access* PP(99): 1-1