Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

doi:10.3233/APC210198

DSAE- Deep Stack Auto Encoder and RCBO- Rider Chaotic Biogeography Optimization Algorithm for Big Data Classification

Anilkumar V. Brahmane [1], Dr B Chaitanya Krishna
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Abstract – In today's era Big data classification is a very crucial and equally widely arise issue is many applications. Not only engineering applications but also in social, agricultural, banking, educational and many more applications are there in science and engineering where accurate big data classification is required. We proposed a very novel and efficient methodology for big data classification using Deep stack encoder and Rider chaotic biogeography algorithms. Our proposed algorithms are the combinations of two algorithms. First one is Rider Optimization algorithm and second one is chaotic biogeography-based optimization algorithm. So, we named it as RCBO which is integration is ROA and CBBO. Our proposed system also uses the Deep stack auto encoder for the purpose of training the system which actually produced the accurate classification. The Apache spark platform is used initial distribution of the data from master node to slave nodes. Our proposed system is tested and executed on the UCI Machine learning data set which gives the excellent results while comparing with other algorithms such as KNN classification, Extreme Learning Machine Random Forest algorithms.

Keywords – Deep learning, Big Data Classification, Apache Spark

1. Introduction

Data mining algorithms are used for extracting the meaningful data from the big data. Various applications are available in day-to-day life where these data mining algorithms are playing important role for classifications and clustering.

Some of such algorithms are KNN- K- Nearest Neighbor classification, ELM- Extreme Learning Machine, Random Forest algorithm and many more. We study all these above algorithms and find out that each one of these is having some advantages and disadvantages. No doubt that these algorithms are giving the accurate results. But as the Vs (Volume, velocity, variety, veracity **and** value) related to big data is changing from platform to platform, from application to applications the data which is generating is heterogeneous in nature and many times its difficult to achieve the accurate

[1] Anilkumar V. Brahmane,Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, India.

Email: brahmaneanilkumarcomp@sanjivani.org.in.

classification. Benefits of KNN are; 1) No Training Period 2) Since the KNN calculation requires no preparation prior to making forecasts, new information can be added flawlessly which won't affect the precision of the calculation. 3) KNN is extremely simple to execute. Drawbacks of KNN are; 1) doesn't function admirably with enormous dataset 2) Doesn't function admirably with high measurements 3) Need highlight scaling 4) Touchy to loud information, missing qualities and exceptions. Similarly, ELM – extreme machine learning is a proficient learning calculation for the single secret layer feed forward neural organizations. Contrasted and the other customary neural organization calculation it has the benefit of over-fitting issues and moderate preparing speed.

For accurate classification neural network plays crucial role. Deep learning is essential for a more extensive group of AI techniques. Learning can be regulated, semi-directed or unaided. After training the deep neural network efficiently we can achieve the accurate results. And as far as the data analytics is concern the Apache Spark is the widely used data processing and data analysis tool for big data. The purpose our research is for accurate data classification using Apache Spark based on Optimization algorithms. We used the RCBO that is Rider Chaotic Biogeography Based Optimization. This is the hybrid method made up of ROA- Rider Optimization Algorithm and CBBO – Chaotic Biogeography Based Optimization. The paper consists of; section 1 is Introduction to big data classification. Section 2 is related work and pros and cons. Section 3 proposed big data classification methodology. Section 4 is result analysis. Section 5 is conclusion.

2. Related work – Pros and Cons

Enough work was carried out in past related to classification of big data. We studied some of this existing system and tried to learn the Pros and Cons about it. Some of it is listed here.

S. Ramírez-Gallego, mentioned in his paper about the incremental use of instance selection for big data [1]. This paper is having the pros that it constantly improves the performance. The Cons is that this paper is not addressing anything about huge size and redundancy problems. M.Duan address in his paper about ELM – extreme learning machine based on Spark Framework- SELM [2]. The advantage of this system is that it focuses on highest speedup due to SELM. The disadvantages are that more memory is needed for this system. Elsebakhi, E., mentioned in his paper about new large-scale machine learning classifier [3]. Good thing in his paper that it mentioned about saving the computational time. Cons about this system, that the performance of system is better with minimal values only. W. Lin focused on ensemble random forest algorithm with parallel computations [4]. This system possesses the strong point it uses SVM which yield performance. Problem with this system is that prediction is not accurate. Hernández address in his paper about machine learning to optimize parallelism in big data [5]. This system accurately predicts the execution time. But suffered due to complexity. Ramírez-Gallego mentioned in his paper about distributed discretization algorithm [6]. The advantage of this system is that its accuracy and simplicity. But concept of drift is affecting this system. Karim addresses in his research work about ASP-Tree Construction Algorithm [7]. The best point about this system it reduces the search time and space time. The problem with the system is that difficult to generate

gigantic synthetic dataset. B. Zhao proposed the LDA training system [8]. The pros about this system that it's having good scalability. The cons are that auto-tuning is major challenges of this system.

After learning and understanding the above system thoroughly we come to many conclusions and eventually reach to the point where we found the challenges in the existing system which are listed below. We also focus how to overcome these challenges.

1. Apache Spark is outstanding for its capacity to manage gigantic data using circled memory with an open-source stage. Regardless, the display of the Spark is trying issue while procuring the best yield from the Spark as the settings of Spark game plan using colossal limits configuration ominously impact the show in a gigantic degree.
2. The lively augmentation of advancement provoked the partner of tremendous data which came to appoint where the normal data taking care of are insufficient to offer the lucid and computational game plans.
3. In the programmable gathering condition challenged a couple of obstructions. From the start, different applications should be improved meanwhile and should have been taken care of with data enlisting which is a troublesome endeavor. Likewise, the flexibility of issue in central yet a tangled task. Moreover, the gathering masterminds the enlisting resources in an amazing manner which intensifies the block in the application. Along these lines, the clsuet enrolling needs a functioning response for perform different calculations.
4. The huge issue in changing data and isolating features joins obtainment of obvious data and encoding the data in numerical portrayal. Be that as it may, the extraction of basic data from the substance data and managing picture or sound data is a critical issue.
5. Attaining a totally suitable data model using Spark and MPI to give versatility, interoperability, and execution improvement adept for consistent data processing is a huge issue not fulfilled by any standard stage.These challenges are summarized as below;
6. Apache Spark configurations which can be reconfigure to achieve the better performance.
7. Cluster computing to support large scale data.
8. Cluster's environment has to be re- written in parallel manner.
9. Extracting useful features from big data.
10. Data model availability which suitable for Apache Spark architecture.

After finding above challenges, we decided our objectives which are;

1. To distribute the data equally among the salve's nodes.
2. To develop the feature vector comprising of optimized feature vector based on optimal weight.
3. To develop training algorithm using deep learning for accurate classification.
4. To get the accurate big data classification.

2.1 Existing algorithm problem

Apache Spark, popularly known for big data processing capability, is a distributed open-source platform that uses the concept of distributed memory to facilitate big data processing proficiently. From the aspect of performance, it is still a big challenge to obtain the best output from Spark, since the Spark configuration settings with large parameters configuration affect its performance at large extent [9]. The rapid development of technology has led to generation of large-scale data, and it has reached a point where sequential and traditional data processing model are not able to provide all the analytical and computational solutions. Hence, to overcome these challenges a number of clusters and distributed computing approaches and frameworks were put forward to support large-scale data intensive applications [10]. Programmable Clusters environment has brought several challenges: Firstly, many applications need to be rewritten in a parallel manner, and the programmable Clusters need to process more types of data computing; Secondly, the fault tolerance of the Clusters is more important and difficult; Thirdly, Clusters dynamically configure the computing resources between shared users, which increases the interference of the applications. With the rapid increase of applications, Cluster's computing requires a working solution to suit different calculations [11]. Common challenges during data transformation and feature extraction include: Taking categorical data (such as country for geolocation or category for a movie) and encoding it in a numerical representation. Extracting useful features from text data, Dealing with image or audio data [12]. Achieving a data model fully compatible for Spark and MPI that provides scalability, performance and interoperability suitable for scientific data assimilation remains a challenge not fully satisfied by any existing platform, and this is the goal of our framework [13]. The emerging industrial big data have '5V' characteristics (volume, velocity, variety, veracity, and value), which challenges the traditional prognostics models [14].

2.2 Proposed system benefits

The benefits of the big data classification are demonstrated in this section.

2.2.1. Banking

The process of managing and evaluating the data of banks and other financial services organizations contains huge amount of client data which includes personal and security information. In banking and finance, reference data, trade and market data, transaction data can be structured or unstructured based on the collected information. Thus, the classification of big data enables to manage all the data in one place in a structured manner.

2.2.2. Cloud Computing

The communication between the servers using information technology produces huge amount of data. These data needs to processed and stored for proper functioning. Thus, the cloud is used as an online storage model for processing huge amount of data.

2.2.3. Healthcare

The big data contains huge amount of data that is available for healthcare providers to monitor the health risks. Thus, the healthcare information and the rising care for health has adapted a big space in making strategic business decisions.

2.2.4. Data mining

The big data classification uses data mining for predicting the meaningful data instead of taking unnecessary data. It utilizes the data and analytics for identifying the best practices for the classification.

2.2.5. Stocks

It is simpler to analyze the trending stocks as per the classified result. The classification can help to check the interest of the people approaching the stocks

2.3 Technical clearness in proposed work

In proposed work the specialized angle is clarify underneath;

1. We are utilizing the Apache Spark. Apache Spark is having starting hubs and last hubs.
2. Our proposed work is to play out the order utilizing two stages; highlight choice and large information arrangement.
3. These element choice and arrangement is acting in the underlying hubs and last hubs of Apache Spark.
4. Big Data arrangement is begun in starting hub of Spark. We additionally called this underlying hub as expert hub.
5. This ace hub accumulates the Big information from different sources over the web. This expert hub conveys this information among slave hubs.
6. The slave hubs will play out the component choice utilizing ideal highlights choice utilizing proposed proficient improvement strategy. The name of this streamlining procedure is RCBO-Rider Chaotic Biography improvement
7. These chose highlights by utilizing RCBO will be given to definite hubs of Spark for where the Deep learning calculation will be utilized to prepare the framework. In our propose work this part is significant and we carry out it utilizing Proposed RCBO based profound stack auto encoder.
8. This Proposed RCBO based profound stack auto encoder will be train to arrange the large information. Here the outcomes from all slave hubs as ideal element chose will be given to group the information. Our proposed RCBO based profound stack auto encoder will play out the exact.
9. We proposed the RCBO which is the novel procedure comprise of two things 1) Rider Optimization Algorithm (ROA) and 2) tumultuous biogeography-based-streamlining (CBBO).

3. Proposed system

This section explains the working of proposed system. As shown in fig. 1 the architecture of our proposed system consists of Apache Spark framework. The Apache Spark consists of Master node and Slave nodes.

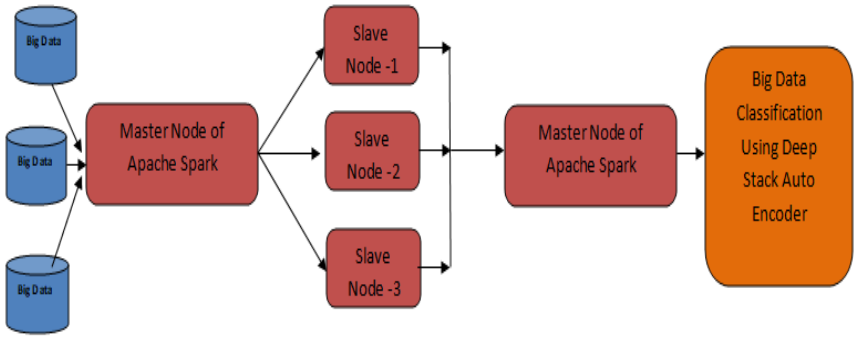

Figure 1. Architecture of the proposed system

As shown in the figure 1 the resources for the big data are various like agricultural data, stock market data, educational data, government data, employee data etc. This big data is collected and given to master node of the Apache Spark. Here in Master node the Big data is partition equally among the slave nodes of Apache Spark. These two nodes also known as initial nodes and final nodes of Spark. In the slave nodes the proposed Rider Chaotic Biogeography Optimization – RCBO is used to select the optimal features of the big data. Each slave nodes submit these selected optimal features to master node. In the master node the actual classification of big data take place. For accurate classification again the proposed RCBO based deep stack auto encoder algorithm is applied on optimal features.

Sources of Big data – Big data can be collected through various sources like government employee, agriculture, healthcare, clients etc.
Master node of Apache Spark- In the first phase the big data collected from various sources are given to master node of Apache Spark where it partitioned among slave nodes. In the second phase the selected optimal features are again integrated together from each slave nodes and fed to master node where the classification will take place.
Slave node of Apache Spark – The partitioned big data is given to slave nodes where the proposed RCBO algorithm is used to select the optimal features from big data. Each slave nodes runs the proposed RCBO algorithm for selecting the optimal features.
Deep stack auto encoder – The selected optimal features will be taken as the input and the neural network will be trained to classify the accurate data. Deep stack auto encoder is the ANN which is having encoder and decoder which decode the input and encode the output.
RCBO – rider chaotic biogeography-based optimization algorithm and deep stack auto encoder are both very novel and effective algorithms for our system.

3.1 Rider Chaotic Biogeography Based Optimization – RCBO

3.1.1. Essential BBO Algorithm

Biogeography-based improvement (BBO), recommended by Simon, is a novel populace based streamlining strategy for tackling worldwide advancement issues. It depends on the idea of biogeography, which is the investigation of the relocation, speciation, and elimination of species. In biogeography, natural surroundings imply a

biological region which is occupied by a specific plant or creature species and is geologically secluded from different environments. Every one of the territories is considered as a person with its environment appropriateness list (HSI) to gauge the decency for living. A living space with a high HSI demonstrates that it is more fit as living spots for natural species and will in general have countless species while an environment with a low HSI shows that it is less appropriate for species to live there and will in general have few species.

The elements of the development of the species among various living spaces is primarily administered by boundaries called migration and resettlement rate and these two boundaries relies on the quantity of species in the natural surroundings**.**

Mayhem hypothesis is a part of arithmetic zeroing in on the investigation of confusion—conditions of dynamical frameworks whose evidently arbitrary conditions of turmoil and inconsistencies are regularly administered by deterministic laws that are exceptionally touchy to beginning conditions. Chaos hypothesis is an interdisciplinary hypothesis expressing that, inside the obvious irregularity of turbulent complex frameworks, there are fundamental examples, interconnectedness, steady input circles, redundancy, self-closeness, fractals, and self-organization. The butterfly impact, a hidden standard of bedlam, portrays how a little change in one condition of a deterministic nonlinear framework can bring about huge contrasts in a later state (implying that there is delicate reliance on introductory conditions). A similitude for this conduct is that a butterfly fluttering its wings in China can cause a storm in Texas.

Little contrasts in starting conditions, for example, those because of blunders in estimations or because of adjusting mistakes in mathematical calculation, can yield broadly wandering results for such dynamical frameworks, delivering long haul forecast of their conduct unimaginable in general. This can happen despite the fact that these frameworks are deterministic, implying that their future conduct follows a novel evolution and is completely dictated by their underlying conditions, with no arbitrary components involved. at the end of the day, the deterministic idea of these frameworks doesn't make them predictable. This conduct is known as deterministic tumult, or essentially confusion. The hypothesis was summed up by Edward Lorenz as:

Mayhem: When the present decides the future, however the surmised present doesn't roughly decide what's to come.

Turbulent conduct exists in numerous regular frameworks, including liquid stream, heartbeat inconsistencies, climate and climate. It likewise happens immediately in certain frameworks with fake segments, for example, the financial exchange and street traffic. This conduct can be concentrated through the examination of a tumultuous numerical model, or through logical procedures, for example, repeat plots and Poincaré maps. Mayhem hypothesis has applications in an assortment of controls, including meteorology, anthropology, social science, physics, natural science, software engineering, designing, financial aspects, science, environment, pandemic emergency management, and theory. The hypothesis framed the reason for such fields of study as perplexing dynamical frameworks, edge of mayhem hypothesis, and self-get together cycles.

3.1.2. CBBO Algorithm

Because of the adaptability and power in tackling advancement issues, BBO calculation has effectively stimulated extraordinary interest. Notwithstanding, a few defects actually exist on this calculation, like the huge number of cycles to arrive at the

worldwide ideal arrangement and the propensity to meet to nearby best arrangements. To beat these defects of the old style BBO calculation, CBBO, which coordinates BBO with bedlam hypothesis, was proposed in our work. After the change activity of every age, lead the tumultuous pursuit to pick better arrangements into future. Along these lines, our proposed calculation exploits the qualities of the turbulent variable to make the people of sub ages circulated ergodically in the characterized space and consequently to keep away from the untimely of the people.

3.1.3. BBO Algorithm

The biogeography-based headway (BBO) estimation is gotten from the biogeography discipline, which is generally established on the transport of species in nature. Species have certain rules according to which development is coordinated among segregated islands through various obstacles. Species can comprehend developments among these islands by drifting, using the breeze, and various ways. The means in the calculations and detail flowchart are appeared underneath.

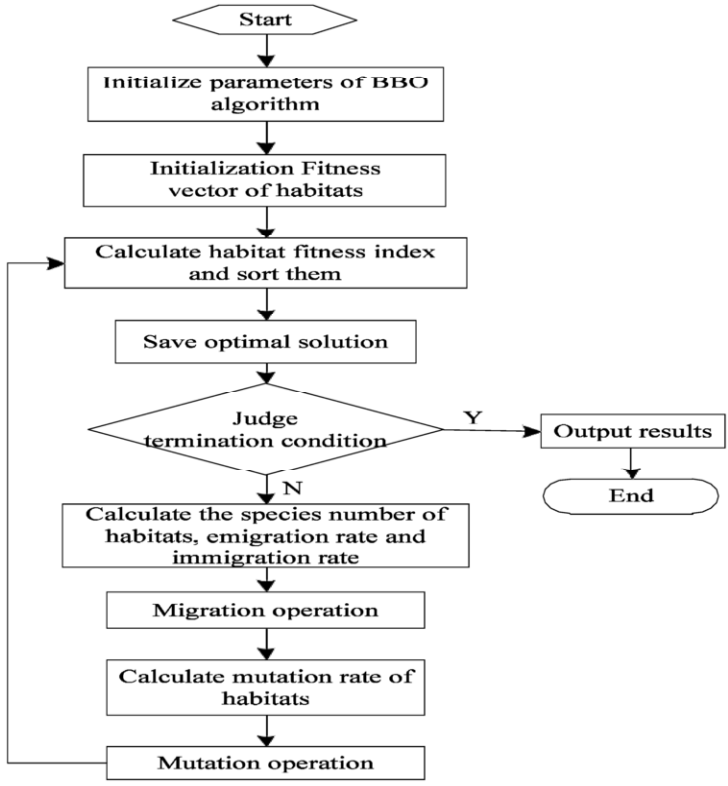

Figure 2. Steps for optimization algorithm

3.1.4. ROA Algorithm

The ROA considers a couple of rider gatherings, who travel to a typical objective area for turning into the champ of the race, to form its thought. The quantity of gatherings

considered is four, where the quantity of riders in each gathering is chosen similarly from the absolute number of riders. The four gatherings of riders are sidestepping rider, supporter, overtaker, and assailant. Every single gathering follows various techniques to arrive at the objective, as follows.

1. The detour rider means to arrive at the objective by bypassing the main way.
2. The supporter follows the main rider in a large portion of the pivot.
3. The overtaker follows his own situation to arrive at the objective, as per the close by area of the main rider.
4. The assailant takes the situation of the rider to arrive at the objective point, using the most extreme speed.

Despite the fact that the riders follow a predefined procedure, the principal variables to arrive at the objective are the right riding of the vehicle by legitimate treatment of the directing, stuff, gas pedal, and brake. For each time moment, the riders change their situations toward the objective by changing these boundaries and follow the predefined system dependent on the current achievement rate, which is contrarily relative to the distance between the situation of the riders and the objective.
The main rider is characterized dependent on the achievement rate at the current time moment. This cycle is proceeded, until the riders go into off time, which is the most extreme time given for the riders to arrive at the objective. After the off time, the rider, who is the main rider, is named as the champ of the riding race. By following this anecdotal idea, another advancement calculation is being created, as portrayed in figure 3.

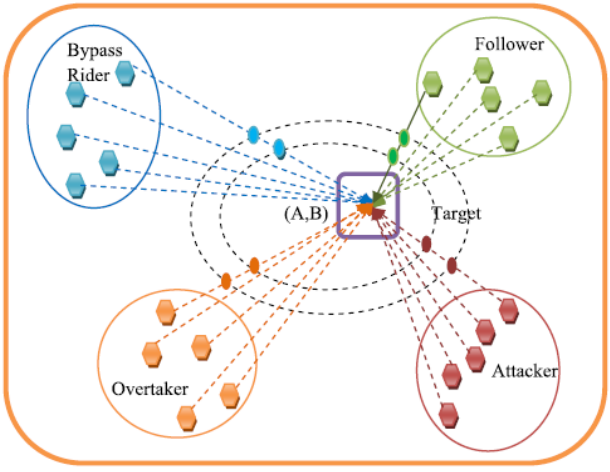

Figure. 3. Overview of ROA

Our main contribution to this system is the implementation of RCBO. This RCBO is the integration of two algorithms one is ROA- Rider optimization algorithm and second is CBBO- Chaotic Based Biogeography Optimization.

In every slave node of Apache Spark Feature selection is done by using RCBO. Optimal features are selected using following steps;

1) Big data is partition in to different subsets.
2) Subsets are equal to slave nodes, where the feature selection is carried out.
3) Solution vector is created which consist of selected features.
4) The optimal features are selected depending upon the fitness function.
5) Fitness function is based on minimization problem.
6) Solution provides less MSE is selected as the efficient solution.

3.2 Proposed system algorithm for selecting the optimal features

Our proposed system for big data classification is implemented with the help of combinations of two optimization algorithms. They are ROA – Rider Optimization Algorithm and CBBO – Chaotic Biography based Optimization Algorithm. The for the optimal feature's selection the proposed RBCO algorithm is used. Rider optimization algorithm is based on the concept of a group of riders riding towards achieving their goal. Chaotic Biography optimization algorithm is based on the concept of species migration.

1) Initialize the population
2) Calculate the fitness function
3) Find out the optimal solution

The first step is to initialize the all-available solutions which are calculated based on the fitness functions. This population of the available solution can be represented by the set of solutions.
The fitness function is used to calculate the best solution. The Mean Square Error this fitness function is use to calculate the optimal solution.
To find out the optimal solution using proposed RCBO, we use the characteristics of CBBO and characteristics of ROA algorithms. By integrating the characteristics of these tow algorithms, we get the proposed RCBO algorithm.
Based on the selected optimal solutions which are nothing but the optimal features the feature vector is created. This features vector is consisting of the optimal solutions. So, we can form a set of optimal solutions which is called as features vector.

3.3 RCBO based Deep stack auto encoder

After getting the optimal features as a result of RCBO, these optimal features are again feeds to master node of the Apache Spark. In master node all collected optimal features from all slave nodes are integrated and the classification process is carried out on this selected and integrated optimal feature.

For accurate classification we used the training neural network, which is deep neural network we term it as deep stack auto encoder. Fig .4 shows the architecture of training neural network for accurate classification.

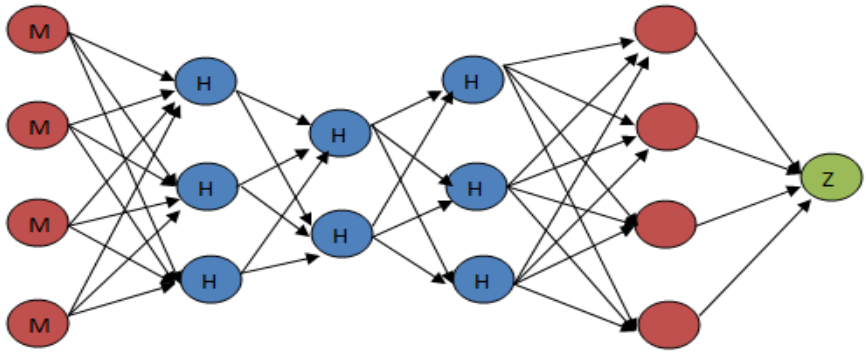

Figure 4. Training the neural network using Deep stack auto encoder

As shown in the figure 4 the input to the neural network is the optimal features which are selected initially using the RCBO. Multiple hidden layers are used to shuffle the input vector to yield the better accurate results. The training to this deep stack auto encoder is done using the RCBO – rider chaotic biogeography-based optimization algorithm.

3.4 Proposed steps for big data classification using the deep stack auto encoder

The feature vector which is obtained in optimal feature selection is given as input to big data classification. In classification process the deep stack auto encoder is used as shown in the figure 2. This is the ANN which is trained to give the accurate classification. This auto encoder is having the input layer, hidden layers, and output layer. The auto encoder is having encoder and decoder which gives the accurate classification.

The proper weights and the bias is given to this neural network for producing the accurate results The proposed RCBO is used here again to train the neural network. This step is executed in the master node of apache spark.

4. Result analysis

4.1. Dataset

We use the Cover Type Data Set which is the UCI machine learning dataset. The details about the dataset are given below;

Table 1. Characteristics of the dataset

Data Set Characteristics:	Multivariate	**Number of Instances:**	581012	**Area:**	Life
Attribute Characteristics:	Categorical, Integer	**Number of Attributes:**	54	**Date Donated**	1998-08-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	282406

4.2. Informational index Information

Anticipating woodland cover type from cartographic factors just (no distantly detected information). The genuine timberland cover type for a given perception (30 x 30 meter cell) was resolved from US Forest Service (USFS) Region 2 Resource Information System (RIS) information. Free factors were gotten from information initially acquired from US Geological Survey (USGS) and USFS information. Information is in crude structure (not scaled) and contains parallel (0 or 1) sections of information for subjective autonomous factors (wild regions and soil types).
This investigation region incorporates four wild regions situated in the Roosevelt National Forest of northern Colorado. These regions address timberlands with negligible human-caused unsettling influences, so that current woodland cover types are more an aftereffect of environmental cycles instead of backwoods the board rehearses.

Some foundation data for these four wild regions: Neota (region 2) presumably has the most noteworthy mean elevational worth of the 4 wild regions. Rawah (region 1) and Comanche Peak (region 3) would have a lower mean elevational esteem, while cache la Poudre (region 4) would have the most reduced mean elevational esteem.
Concerning essential significant tree species there, Neota would have tidy/fir (type 1), while Rawah and Comanche Peak would likely have lodgepole pine (type 2) as their essential species, trailed by tidy/fir and aspen (type 5). Store la Poudre would will in general have Ponderosa pine (type 3), Douglas-fir (type 6), and cottonwood/willow (type 4).
The Rawah and Comanche Peak regions would will in general be more commonplace of the by and large dataset than either the Neota or Cache la Poudre, because of their arrangement of tree species and scope of prescient variable qualities (rise, and so on) Cache la Poudre would most likely be more extraordinary than the others, because of its moderately low height territory and species creation.

4.3. Attribute Information

Given is the trait name, characteristic sort, the estimation unit and a concise depiction. The timberland cover type is the characterization issue. The request for this posting compares to the request for numerals along the columns of the data set.

Table 2. Attribute Information

Name	Data type	Measurement	Description
Elevation	quantitative	meters	Elevation in meters
Aspect	quantitative	Azimuth	Aspect in degrees azimuth
Slope	quantitative	degrees	Slope in degrees
Horizontal_Distance_ To_Hydrology	quantitative	meters	Horz Dist to nearest surface water features
Vertical_Distance_To _Hydrology	quantitative	meters	Vert Dist to nearest surface water features
Horizontal_Distance_ To_Roadways	quantitative	meters	Horz Dist to nearest roadway
Hillshade_9am	quantitative	0 to 255 index	Hillshade index at 9am, summer solstice
Hillshade_Noon	quantitative	0 to 255 index	Hillshade index at noon, summer soltice
Hillshade_3pm	quantitative	0 to 255 index	Hillshade index at 3pm, summer solstice
Horizontal_Distance_ To_Fire_Points	quantitative	meters	Horz Dist to nearest wildfire ignition points
Wilderness_Area	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Soil_Type	qualitative	0 (absence) or 1 (presence)	Soil Type designation
Cover_Type (7 types)	integer	1 to 7	Forest Cover Type designation

We tested and executed our system using the UCI machine learning data sets forest cover type data set. The analysis parameters we used are; Accuracy, Sensitivity, Specificity. We compare our results with ELM, K-NN and random forest algorithms. Every time our results are better than these classification algorithms.

We tested the system using tanning data. We got the better results are compare to ELM, K-NN and random forest algorithms. Similarly, we tested the system using number of features selected again we got the better results as compare to ELM, K-NN, and random forest algorithm. These results of our system are shown in the fig. 5.

For the purpose of performance analysis the proposed system is executed and tested with respective accuracy, sensitivity, and specificity.

Our proposed system is executed on above data set. We get following results.

1. Performance analysis – After comparing the 50% of training data we observed that our proposed algorithms give better exactness. The various layers of deep stack auto encoder give the better results.
2. The responsiveness also gives the better results at various layers.
3. The specificity also gives the better results at various layer of learning algorithm.
4. Performance analysis - We compare the proposed system upon selecting the features. We observed that the proposed system gives the better results at various layers of learning algorithm.
5. We compare the proposed system upon selecting 5 features and we get better results for exactness, responsiveness and specificity.
6. Comparative analysis - We compare the proposed system against the algorithms such as ELM, Random Forest, K-NN algorithms.
7. We test the results using 50 % training data against above algorithms; we observed our proposed system gives better results for parameters such as exactness, responsiveness and specificity.
8. Comparative analysis - We tested our system using 5 selected features and observed better results against such

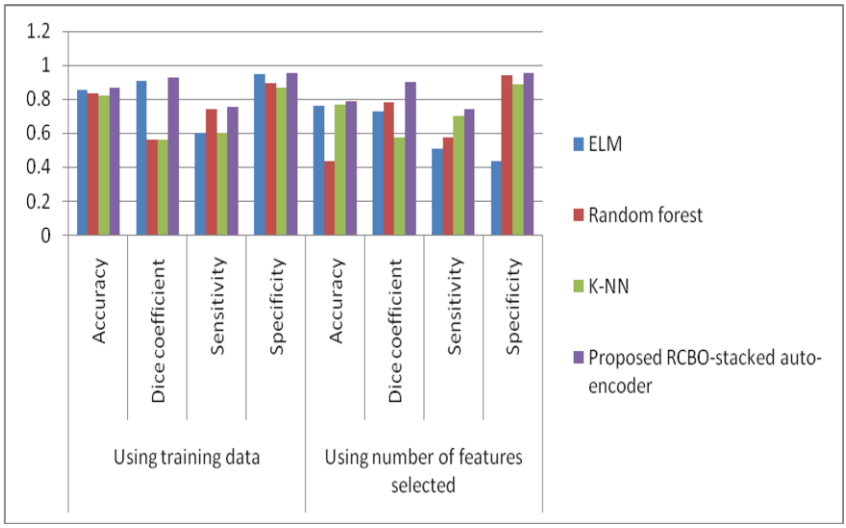

Figure 5. Result Analysis

5. Conclusion

In this paper we propsed the system using Rider Chaotic Biogeography Based Optimization algorithm and using the deep stack auto encoder for accurate classification of big data. We use the Apache Spark framework for gathering the big data in the master node of Apache Spark and then finding the optimal features of this

big data in the slave node of Apache Spark. Our proposed RCBO – algorithm is the integration of the ROA – rider chaotic biogeography optimization and CBBO- Chaotic Biogeography based Optimization algorithm. We proposed the deep stack auto encoder as a training deep neural network for getting the accurate classification of the big data. We tested out system with existing algorithms like ELM, K-NN, and random forest algorithms. We obtained the better results as compare to these existing algorithms.

References

[1] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, J. M. Benítez and F. Herrera, "Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 10, pp. 2727-2739, October 2017.

[2] M. Duan, K. Li, X. Liao and K. Li, "A Parallel Multi classification Algorithm for Big Data Using an Extreme Learning Machine," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 6, pp. 2337-2351, June 2018.

[3] Elsebakhi, E., Lee, F., Schendel, E., Haque, A., Kathireason, N., Pathare, T., Syed, N. and Al-Ali, R., "Large-scale machine learning based on functional networks for biomedical big data with high performance computing platforms," Journal of Computational Science, vol. 11, pp. 69-81, 2015.

[4] W. Lin, Z. Wu, L. Lin, A. Wen and J. Li, "An Ensemble Random Forest Algorithm for Insurance Big Data Analysis," in IEEE Access, vol. 5, pp. 16568-16575, 2017.

[5] Hernández, Á.B., Perez, M.S., Gupta, S. and Muntés-Mulero, V., Using machine learning to optimize parallelism in big data applications. Future Generation Computer Systems, vol. 86, pp.1076-1092, 2018.

[6] Ramírez-Gallego, S., García, S., Benítez, J.M. and Herrera, F., "A distributed evolutionary multivariate discretizer for big data processing on apache spark," Swarm and Evolutionary Computation, vol. 38, pp. 240-250, 2018.

[7] Karim, M.R., Cochez, M., Beyan, O.D., Ahmed, C.F. and Decker, S., "Mining maximal frequent patterns in transactional databases and dynamic data streams: a spark-based approach," Information Sciences, vol. 432, pp.278-300, 2018.

[8] B. Zhao, H. Zhou, G. Li and Y. Huang, "ZenLDA: Large-scale topic model training on distributed data-parallel platform," in Big Data Mining and Analytics, vol. 1, no. 1, pp. 57-74, March 2018

[9] M. A. Rahman, J. Hossen and V. C, "SMBSP: A Self-Tuning Approach using Machine Learning to Improve Performance of Spark in Big Data Processing," 2018 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, pp. 274-279, 2018.

[10] S. Srivastava, A. Nigam and R. Kumari, "Work-in-Progress: Towards Efficient and Scalable Big Data Analytics: Mapreduce vs. RDD's," 2017 International Conference on Information Technology (ICIT), Bhubaneswar, pp. 272-275, 2017.

[11] Z. Han and Y. Zhang, "Spark: A Big Data Processing Platform Based on Memory Computing," 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Nanjing, pp. 172-176, 2015.

[12]]J. Fu, J. Sun and K. Wang, "SPARK – A Big Data Processing Platform for Machine Learning," 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, pp. 48-51, 2016.

[13] S. Caíno-Lores, J. Carretero, B. Nicolae, O. Yildiz and T. Peterka, "Spark-DIY: A Framework for Interoperable Spark Operations with High Performance Block-Based Data Models," 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT), Zurich, pp. 1-10, 2018.

[14] J. Yan, Y. Meng, L. Lu and C. Guo, "Big-data-driven based intelligent prognostics scheme in industry 4.0 environment," 2017 Prognostics and System Health Management Conference (PHM-Harbin), Harbin, pp. 1-5, 2017.