Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-018-0187-z

ORIGINAL RESEARCH

Efficient DANNLO classifier for multi-class imbalanced data on Hadoop

S. Satyanarayana[1] · Yerremsetty Tayar[2] · R. Siva Ram Prasad[2]

Received: 27 May 2017 / Accepted: 13 April 2018
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2018

Abstract In recent years, multi-class imbalance data classification is a major problem in big data. In such situations, we focused on developing a new Deep Artificial Neural Network Learning Optimization (DANNLO) Classifier for large collection of imbalanced data. In our proposed work, first the dataset reduction using principal component analysis for dimensionality reduction and initial centroid is computed. Then, parallel hierarchical pillar k-means clustering algorithm based on MapReduce is used to partitioning of an imbalanced data set into similar subset, which can improve the computational cost. The resultant clusters are given as input to the deep ANN for learning. In the next stage, deep neural network has been trained using the back propagation algorithm. In order to optimize the n-dimensional weight space, firefly optimization algorithm is used. Attractiveness and distance of each firefly is computed. Hadoop is used to handle these large volumes of variable size data. Imbalanced datasets is taken from ECDC (European Centre for Disease Prevention and Control) repository. The experimental results illustrated that the proposed method can significantly improve the effectiveness in classifying imbalanced data based on TP rate, F-measure, G-mean measures, confusion matrix, precision, recall, and ROC. The experimental results suggests that DANNLO classifier exceed other ordinary classifiers such as SVM and Random forest classifier on tested imbalanced data sets.

Keywords Imbalanced dataset · Principal component analysis · Clustering · MapReduce · Hadoop · Classification

1 Introduction

Big data is a term that describes the large volume of data which depends on computational time, efficiency and velocity. Big data has emerged in business, scientific and engineering disciplines also social networks observe the user action sites then display improvement of site design, spam and fraud detection [1]. Exploring the large volume of data and extracting useful information and knowledge is a challenge, and sometimes, it is almost infeasible. Big data starts with large volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data [2].

Data science appears with the objective of extracting knowledge from huge volume of data, but new technologies, instruments, etc., are needed to address this objective [3]. Spark highlights as one of the most flexible and powerful engines to perform faster distributed computing in big data by using in-memory primitives. The machine learning makes intelligent decisions automatically [4, 5]. Here, multi class classification is the major problem categorized by two methods such as problem transformation and problem adaptation [6].

✉ S. Satyanarayana
s.satyans1@gmail.com

Yerremsetty Tayar
tayarphd@gmail.com

R. Siva Ram Prasad
raminenisivaram@yahoo.co.in

[1] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation (K L University), Green Fields, Vaddeswaram, Guntur, AP, India

[2] Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, AP, India

Published online: 19 April 2018

Imbalanced data typically defined as a problem with classification problems where the classes are not represented equally. Imbalanced problem consider two classes of one is represented by large samples and another one is represented by lowest samples from the training data and output data sets fit the best data [7]. Data set context in the big data can be divided into structured and unstructured data have an attribute such as feature extraction and feature reduction [8]. This will handle sampling, transformation, denoising, and normalization under this selection of representative subset, earn single input, removal of noise and feature extraction can be done. Types of procedures for sampling instances from large data set is random sampling and stratified sampling [9].

Decomposition approach is used to solve multi class problem, through this approach original problem will be divided into sub problem according to tuple or feature oriented sample and space [10]. When data become too large difficult to capture, store and manage, the decomposed subsets are grouped together by clustering relate this will fine hidden relationship between the data sets [11]. Big data clustering technique classified into single machine clustering and multiple machine clustering. In single machine, select optimal subset of variables as well as in multiple machine data imposes parallel computing to achieve results in least time also speed up calculation and increase scalability [12].

To improve classification performance classifiers are involved namely multilayer perceptron, neural network and support vector machine are combined to form a multi classifier system. In single data sets usually give poor classification performance, but in multilevel classifier combines first level with trained data and supply the input to the next level classifier with no need of fusion methods [13]. To construct classifier based on future query cases domain of medical have several challenges associate with data collection from patients is time consuming also acquire large volume of data [14].

Neural networks are artificial intelligence based technique provide steps towards final decision maker and interconnected nerve cells are neurons which receive information from environment [15]. Cost sensitive learning are effective also imbalance may increase difficulty in real world cost sensitive data set with three types of cost matrices [16]. Formal process to define standard big data benchmark also development of benchmark would span multiple application domains. Hadoop provide the solution of big data as well as apache Hadoop use distributed file system of Hadoop and MapReduce [17, 18].

Hadoop is an open source software for distributed storage and distributed processing also supports data in any form. By designing of MapReduce programming achieve high performance distributed processing and deals with hardware failure. Hadoop distributed file system is an efficient way to store data [19]. Here, master node splits the input data set into sub problems and distribute into the worker nodes, it process smaller problem in parallel manner and give back to the master node, then master node combines all sub problems at that instant perform answer to form output [20].

In this paper, we propose the DANNLO classifier method which is used to balance the number of training examples in such a multi-class setting. We focus on learning from imbalanced data problems in the context of Big Data, especially when faced with the challenge of Volume. MapReduce technique is a framework which is used to handle large volume of data. The objective of this study is to design an imbalanced classification model on Hadoop to improve its behaviour for multi-class imbalanced problems, especially for high imbalanced tasks.

The remaining portion of the paper has been organized as follows, the recent survey of paper is structured in Sect. 2, the proposed methodology of multi-class imbalanced data in Sect. 3, the Sect. 4 derives the experimental analysis and performance metrics, the conclusion part of proposed work has described in Sect. 5, respectively.

2 Related work

Galar et al. [21] proposed an efficient ensemble classifier named EUSBoost which depends on a combination of boosting method with evolutionary under sampling. The use of the evolutionary approach enabled them to choose the most critical samples for the classifier learning step. Extensive volume of datasets were conveyed for examinations, through this high efficiency had achieved. Complexity of the classification increases and difficult to implement in real time platform are the disadvantages in this model.

Jesus et al. [22] presented a resampling approach for classification algorithms that use numerous subsamples. This procedure had been applied to CTC (Consolidated Tree Construction) algorithm over various classification contexts. A robust classification algorithm should not just be able to rank in the top positions for certain classification problems. The robustness of the CTC algorithm against a wide set of classification algorithms with clarifying limit were set up. The serious limitation of the technique is that it includes biased selection and along these lines drives us to reach mistaken determinations.

Lakes et al. [23] proposed an effective approach to estimate SBSTs (seismic building structural types) by combining scarce in situ observations, multi-sensor remote sensing data and machine learning techniques. Experimental outcomes obtained fora representative study area and evaluate the capacities of the exhibited approach. It confirm its great potential for a reliable area-wide estimation of SBSTs and an effective earthquake loss modeling

based on remote sensing, which ought to be additionally investigated in future research exercises. The drawback is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks.

Sun et al. [24] introduced a novel ensemble method, which initially changes over an imbalanced data set into multiple balanced ones and after that manufactures various classifiers on these multiple data with a particular classification algorithm. At last, the classification results of these classifiers for new data are combined by a particular ensemble rule. It is a scientific approach. Therefore, to get a good and representative sample, one should have special knowledge to get good sample and to perform appropriate investigation so that reliable outcome might be accomplished. A primary drawback of ensemble method is that it will only be useful if the classification method does not generate strong classifiers.

Pan et al. [25] presented a parallel large-scale rough set based methods for knowledge acquisition using MapReduce. It was enhanced them on several representative MapReduce runtime systems such as Hadoop, Phoenix and Twister. The experimental results demonstrated that the Hadoop has the best speedup for bigger data sets. Rough set methodologies in real applications are time consuming. It uses smaller training sets for evaluation but our work using large datasets from ECDC repository.

3 Proposed methodology for multi-class imbalanced data on Hadoop

The fundamental target of the proposed work is to accomplish great classification precision for every class in the event of imbalanced data sets. A data set is said to be imbalanced when several classes are under-represented (minority classes) in comparison with others (majority classes).Gaining from imbalanced data is new test in certifiable applications, for example, misrepresentation recognition, medicinal finding, funds and system interruption. Any information set which demonstrates an uneven conveyance between its classes can be analyzed imbalanced. This imbalanced problem is maintained in ML algorithm. Imbalanced issue shows up while the quantity of things of a class is not exactly the quantity of things of another class. To settle the multi-class imbalanced data characterization issue, we concentrate DANNLO classifier for substantial gathering of imbalanced information. In the first place the datasets are connected by PCA and Linear change for lessen the dimensionality of the datasets and the calculation of initial centroid. This dimensionality reduction method is used to exchange the high dimensional dataset into important dataset. At that point the imbalanced dataset is partitioned into smaller subset by utilizing parallel hierarchical pillar k-means clustering algorithm based on map reduce. At that point the resultant subsets (cluster) are offered with regards to the contribution of ANN for learning. The source of inputs are prepared by utilizing back propagation algorithm. Firefly optimization algorithm is used to optimize the weight space within ANN. (HDFS) is used to handle these tremendous volumes of variable size data. Imbalanced data sets is taken from ECDC-dataset storage facility. The proposed work is compared with the performance metrics of TP rate, F-measure, G-mean measures, precision, confusion matrix, recall and ROC.

Figure 1 illustrates the architecture of DANNLO classifier for multi-class imbalanced dataset on Hadoop. Hadoop distributed file system (HDFS) is maintain huge volume of imbalanced data set. The data sets are pre-processing by Principal Component Analysis. PCA is used to reduce the dimensionality of data set. Then these data is taken as the input for clustering. Here MapReduce program is used for making cluster partitions. The map reduce partitions are secured in HDFS. Then the output partitions are given to DANNLO classifier for classification process to obtain the classification output.

3.1 Principal component analysis and linear transformation

Principal Component Analysis is used to decrease the estimation of data. Dimensionality reduction is better way to deal with process high dimensional data sets. Hadoop handles large volume of data. So the dimensionality of the data also high. So we need to reduce the dimensionality for simplicity. It may moreover be useful when the elements in the data sets are noisy. Dimension reduction is commonly performed before process the grouping calculation. The pre-processing steps of PCA algorithm are given below:

- Before executing principal component analysis must determine pre-processing
- Calculate the covariance matrix
- Calculate the eigenvectors
- Building segments and framing a component vector
- Building principal components.

First the data diminished from n-dimension to k-dimension. The sample data are $\{x_1, x_2, \ldots x_n\}$. Then compute the mean by using

$$\mu = \left(\frac{1}{n}\right) \sum_{i=1}^{n} x_i, \tag{1}$$

where μ is the mean value, 'n' is number of samples and x_i is data sample. After computing the mean, calculate the covariance by using the following equation:

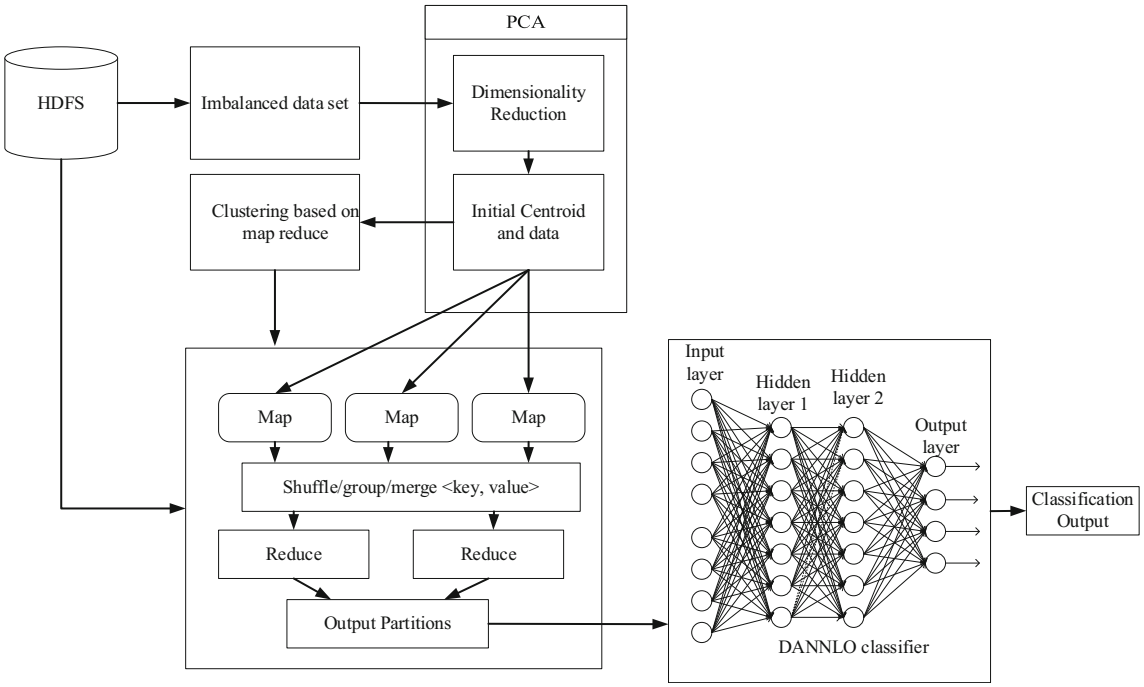

Fig. 1 Architecture of Proposed work

$$C = \left(\frac{1}{n}\right) \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T. \quad (2)$$

Here, n is the quantity of data items.

The eigenvectors are controlled by the accompanying condition:

$$ce_i = \lambda_i e_i. \quad (3)$$

Figure out the eigenvalues and eigenvectors of covariance matrix. The eigenvalues are calculated by the below equation

$$\det(c - \lambda I) = 0. \quad (4)$$

Eigenvector with most noteworthy eigenvalue speak to the principal component.

3.2 Parallel hierarchical pillar k-means clustering algorithm

k-means grouping is for the most part utilized for clustering applications. The point of clustering is to gathering object with the end goal that every object contains relevant object. The procedure of pillar k-means calculation in light of Hadoop contains two sections, the initial segment is to initial clustering centers, and divide the sample data set into a specific size of data blocks for parallel preparing. The second part is to begin the Map and Reduce tasks for parallel processing of algorithm in time, up to process gets the clustering output. Mapper maps input key/value sets to an arrangement of transitional key/value sets. Reducer diminishes an arrangement of middle of the road values which share a key to a smaller arrangement of values. Table 1 illustrates the procedure for clustering with MapReduce.

3.3 Map function

The input is passed to the mapper work line by line. Normally the input is stored on Hadoop file system. The mapper takes the information and makes a few little pieces of information as a <key, value> pairs.

3.4 Reduce function

This stage is the mix of the Shuffle organize and the Reduce arrange. The Reducer's occupation is to handle the information that originates from the mapper. Subsequent to handling, it creates another arrangement of output, which will be secured in the Hadoop DFS (Table 2).

3.5 DANNLO classifier

Deep learning grants to improve the training performance. A DNN (Deep Neural Network) is a feed forward, manufactured neural system that has more than one layer of hidden units between its sources of input and its outputs. The accumulation of output partitions after map reduce taken as the solution of artificial neural systems for learning. Learning is the advance of re-establishing the weights

Table 1 Clustering with MapReduce

Algorithm

Step 1: At first randomly centroid is chosen in view of data.
Step 2: The Input record have initial centroid and data.
Step 3: The "configure" function is used in mapper class which is initially open the record and read the centroids and store in the data structure in Array list format.
Step 4: Mapper read the information record and radiate the nearest centroid with the mark to the reducer.
Step 5: Reducer gather all data and compute the new matching centroids and emit.
Step 6: Files are reading and validating in job structure
if variation amongst old and new centroid is lower than 0.1 then
convergence is arrived
else
Replay step 2 using recent centroids.

Table 2 Confusion matrix

	Positive prediction	Negative prediction
Positive class	107	4
Negative class	17	22

with a specific end goal to create a system that plays out a couple work.

The deep learning algorithms are Deep Boltzmann Machine (DBM), Deep Belief Networks (DBN), Convolutional Neural Network (CNN), and Stacked Auto-Encoders. The network weights are typically random initialized. For each preparation case the system creates an output. Deep learning neural network architectures differ from normal neural networks because they have more hidden layers. Deep learning networks differ from normal neural networks and SVMs because they can be trained in an unsupervised or supervised manner for both unsupervised and supervised learning tasks. DNNs can be discriminatively prepared by back propagating subsidiaries of a cost capacity that measures the inequality between the objective and the actual output delivered for every preparation.

3.5.1 Back propagation algorithm

Back-propagation neural network (BPNN) is a multilayer feed forward network which trains the training data using an error back-propagation mechanism. It has become one of the most widely used neural networks. BPNN can perform a large volume of input–output mappings without knowing their exact mathematical equations. A neural network model of error back propagation algorithm ought to have the capacity to prepare neural systems in the comparable way as the first back propagation algorithm. The system is prepared utilizing back propagation with numerous parameters. The algorithm normally has three layers which are input layer, output layer and hidden layer. Each neuron in hidden layer can get numerous information values from various neurons and a related weight is joined to every input and summed up to deliver output. This output is passed to next hidden layer and a similar operation is repeated.

3.5.2 Firefly optimization

Here firefly optimization algorithm [26] is utilized to enhance the weight space. Progress the attractiveness and distance for every firefly and move to the brighter one. Every firefly advancement relies on upon ingestion of the other one. The firefly calculation takes after three norms,

- Any individual firefly will be pulled into all different fireflies because of all fireflies are unisex.
- The less attractive firefly is pulled into the arbitrarily moving brighter fireflies.
- The attractiveness of every firefly symbolizes the way of the game plan.

The attractiveness of every firefly indicates the description of the arrangement. The grade of attractiveness of a firefly is computed by the coming equation,

$$\beta(r) = \beta_0 \exp(-\gamma r^2), \tag{5}$$

where r—distance among two fireflies, β_0—first attractiveness at r = 0, γ—absorption coefficient.

The distance between any two fireflies i and j at point x_i and x_j properly, can be represented as a Cartesian Euclidean gap is given below

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}\ , \tag{6}$$

where r_{ij} is the space gap of the firefly i and the firefly j and $x_{i,k}$ is the kth segment of the spatial organize X_i of ith firefly and d is the dimensionality of the given issue.

The movement of a firefly i, which is pulled in by a brighter firefly j, is spoken to by the accompanying condition:

$$x_{i+1} = x_i + \beta_0 \times \exp(-\gamma r_{ij}^2) \times (x_j - x_i) + \alpha \times \left(\text{rand} - \frac{1}{2}\right). \tag{7}$$

Here, rand is a random number maker frequently distributed in the space (0, 1). In the majority of the advancement $\beta_0 = 1$ and $\alpha \in [0, 1]$.

The basic function B_i for input is entry by the equation is

$$B_i = \sum_{i=1}^{N} x_i W_{ij}^I. \tag{8}$$

In the above equation x_i is the ith input value, W_{ij}^I is the weights permit among the input and the hidden layer, and N is the aggregate number of input neurons.

The activation function Q is known by following equations

$$Q = \frac{1}{1 + \exp(-B_i)}. \tag{9}$$

The result of the deep neural network is accomplished by the condition given

$$y = \sum_{j=1}^{M} \frac{w_{jk}^O}{1 + \exp\left(-\sum_{i=1}^{N} x_i w_{ij}^I\right)}. \tag{10}$$

In the above equation W_{jk}^O is the weight between hidden and the output layer, N is the total number of hidden neurons, and k is the number of yield.

4 Experimental setup and dataset description

Our experimental results against a number of multi-class problems show that, when the DANNLO classifier is used for pre-processing the data before classification. So that we can obtain highly accurate models that compare to the existing classifiers. In this section explains about the results for multi-class imbalanced data set for classification. To achieve better performance we apply the Deep Artificial neural network technique has been replicated in the MATLAB platform on Hadoop with the following configuration:

Processor: Intel Core 2 Quad @ 2.5 GHz
RAM: 3 GB
Operating system: Windows 7
Mat lab version: R 2016a.

The center of Apache Hadoop comprises of a cache part, known as Hadoop Distributed File System (HDFS), and a handling part called MapReduce. The result of the proposed work is compared with the traditional classifiers such as SVM and random forest.

4.1 Dataset description

Initially data sets are captured from ECDC and the data has various sizes. The datasets contain multiple classes and the class with large size is considered as majority class and with small size is considered as minority class among all classes. Imbalance ratio is calculated by taking the ratio of the size of majority class with minority class. For evaluation purposes the data set was randomly partitioned into training and test set.

4.2 Performance metrics

The performance metrics of TP rate, F-measure, G-mean measures, precision, confusion matrix, recall and ROC of the proposed method is compared with the existing classifier.

$$\text{TP rate or sensitivity or recall} = \frac{TP}{TP + FN}, \tag{11}$$

$$\text{TN rate or specificity} = \frac{TN}{TN + FP}, \tag{12}$$

$$\text{FP rate} = \frac{FP}{TN + FP}, \tag{13}$$

$$\text{FN rate} = \frac{FN}{TP + FN}, \tag{14}$$

$$\text{F - measure} = \frac{(1 + \beta)^2 \times \text{recall} \times \text{precision}}{\beta^2 \times \text{recall} + \text{precision}}, \tag{15}$$

$$\text{G - mean measure} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}, \tag{16}$$

$$\text{Precision or positive predictive value} = \frac{TP}{TP + FP}, \tag{17}$$

where TP is true positive rate, TN is true negative rate, FP is false positive rate and FN is false negative rate. The measurements utilized as a part of imbalanced areas must consider the user preferences and, in this way, ought to consider the data distribution. To satisfy this objective a

Table 3 Performance analysis of various classifier

S. no.	Performance metrics	DANNLO classifier	SVM	Random forest
1.	Sensitivity	0.9640	0.94235	0.90159
2.	Specificity	0.5641	0.54245	0.50169
3.	FP_rate	0.4359	0.41425	0.41567
4.	FN_rate	0.0360	0.07435	0.1357
5.	F1_measure	0.9106	0.50188	0.30677
6.	G_mean	0.7374	0.64738	0.56252
7.	Precision	0.8629	0.8556	0.8450

few execution measures were proposed. A common classification assessment standard is general precision. This metric gives an extensive evaluation when the dataset is moderately adjusted; in any case, this is not the situation in imbalanced situation. It measures the percentage of the examples that are accurately classified.

There is often a trade-off between true positive rate and true negative rate for any classifier, and the same applies for recall and precision. On account of learning amazingly imbalanced data, regularly the uncommon class is of extraordinary interest.

Table 3 illustrate the performance of different classifiers. From the table we clearly show the better performance for DANNLO classifier. Our experimental results against a number of multi-class problems show that, when the DANNLO classifier is used for pre-processing the data before classification. So that we can obtain highly accurate models that compare to the existing classifiers.

4.3 ROC curve

Receiver Operating Characteristic (ROC) curve is a standard method for summarizing classifier performance over a range of trade-offs between true positive and false positive error rates. The Area under the Curve (AUC) is an accepted performance metric for a ROC curve. ROC curves can be reflection of as representing the family of best decision boundaries for relative costs of TP and FP. On an ROC curve the

$$\text{X - axis represents } \%\text{FP} = \frac{TN}{TN + FP}, \tag{18}$$

$$\text{Y - axis represents } \%\text{TP} = \frac{TP}{TP + FN}. \tag{19}$$

Precision is the division of predicted positive examples which are really positive. Figure 2 represents the precision of three classifiers. When comparing to the traditional classifiers, the DANNLO classifier provides better precision. The pre-processing step reduce the dimensionality of the data. So the DANNLO performed well in multi class imbalanced data set.

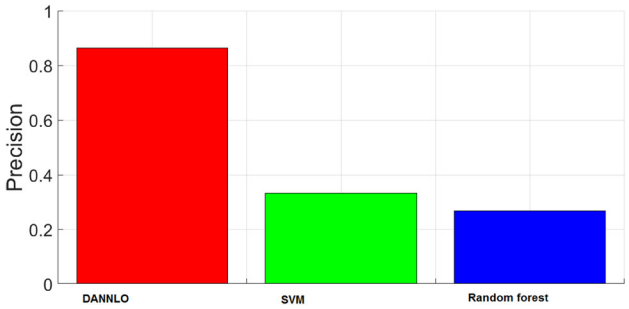

Fig. 2 Precision analysis

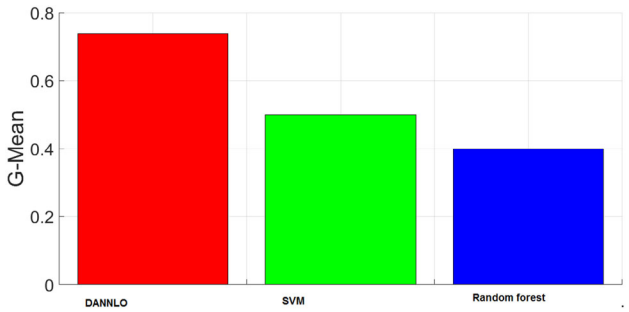

Fig. 3 G-mean analysis

G-mean assesses the degree of inductive bias as far as the proportion of positive accuracy and negative accuracy. Figure 3 demonstrates the G-mean investigation of various classifiers. When contrasting with the current classifiers, the proposed work gives better outcome.

F-measure, is a common metric for binary classification, which can be interpreted as a weighted average of the precision and recall. F-measure is broke down in Fig. 4. The picture clearly shows that the proposed classifier gives better F-measure.

Figure 5 describe the sensitivity analysis. From the figure, the optimization based proposed DANNLO classifier provides high sensitivity. Experiments are carried out in the multi class imbalanced data set. Figures 2, 3, 4 and 5 represent that DANNLO classifier is improve their results in terms of the analysis of precision, G-mean, F-measure and sensitive in comparison with existing classifier.

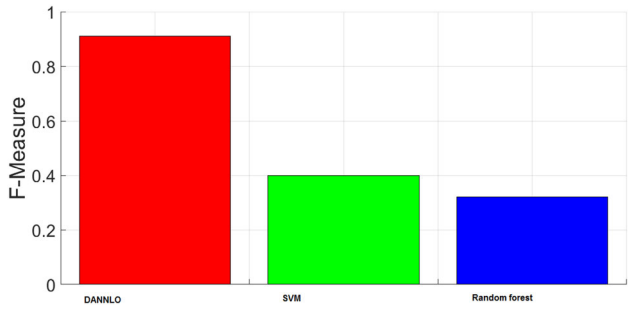

Fig. 4 F-measure analysis

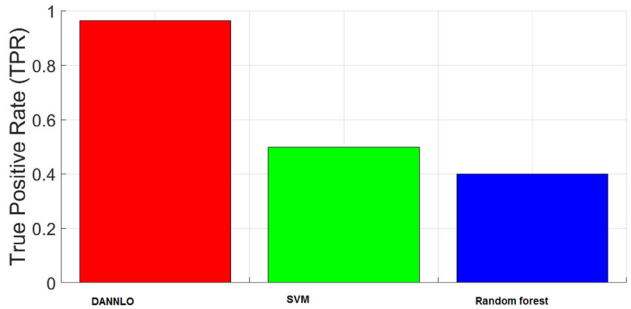

Fig. 5 Sensitivity analysis

Comparing with the existing classifiers, our proposed system shows better performance.

5 Discussion

SVMs (Support Vector Machines) are incredible for generally small data sets with less outliers and Random Forest (RF) needed more data but it require reasonable time to classify the data. Deep learning algorithms work well with larger data sets and it solve the complex problems like image classification, speech recognition and natural language processing. Deep learning works due to the design of the system and the optimization applied to that system.

The network is a directed graph, implying that each hidden unit is associated with numerous other hidden units below it. So each hidden layer going further into the network is a non-linear mix of the layers below it, as a result of all the combining and recombining of the yields from all the past units in combination with their activation functions.

At the point when the optimization routine is applied to the network, each hidden layer then turns into an optimally weighted, non-linear combination of the layer below it. By looking at the above Table 2, it can be inferred that the standard methodologies like SVM and RF independently are insufficient to convey an agreeable classification result. Consequently the classification imbalance problem can be solved with DANNLO technique as it gives most extreme precision and agreeable outcome when contrasted with established models when different size of data used.

6 Conclusion

In order to solve the multi-class imbalance data classification problem, a new Deep Artificial Neural Network Based Learning Optimization (DANNLO) Classifier is presented for large collection of imbalanced data. A big data is the point of attraction because a large amount of data that are currently generating in today's scenario. Our method uses the MapReduce programming model on the Hadoop platform, one of the most popular solutions to effectively deal with big data nowadays. In this way, our model distributes the computation using the map function and then, combines the outputs through the reduce function. Traditional data mining methods cannot handle with requirements asked by huge information. The quality of proposed system tested in terms of precision, recall, F-measure and G-mean measure. The effectiveness of classification with multi-class imbalanced data sets are enhanced with DANNLO classifier. Experimental analysis carried out using datasets of ECDC repository. The results of proposed system demonstrate that DANNLO classifier provides good performance in the imbalanced data problem.

The future scope of this paper suggests that using two or more optimization technique i.e. hybridization in deep convolutional neural network provides better solution for class imbalance problem and it will helpful for the researchers to achieve more accuracy in classification.

References

1. Hu H, Wen Y, Chua T-S, Li X (2014) Toward scalable systems for big data analytics: a technology tutorial. IEEE Access 2:652–687
2. Wu X, Zhu X, Wu G-Q, Ding W (2014) Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97–107
3. Triguero I, Peralta D, B J, García S, Herrera F (2015) MRPR: a MapReduce solution for prototype reduction in big data classification. Neurocomputing 150:331–345 **(Elsevier)**
4. Ou G, Murphey YL (2007) Multi-class pattern classification using neural networks. Pattern Recognit 40(1):4–18 **(Elsevier)**
5. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR) 34(1):1–47
6. López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141 **(Elsevier)**
7. Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378 **(Elsevier)**

8. Lee J, Lapira E, Bagheri B, Kao H-A (2013) Recent advances and trends in predictive manufacturing systems in big data environment. Manuf Lett 1(1):38–41 **(Elsevier)**
9. Dubey R, Zhou J, Wang Y, Thompson PM, Ye J (2014) Analysis of sampling techniques for imbalanced data: an $n = 648$ ADNI study. NeuroImage 87:220–241
10. Rokach L (2006) Decomposition methodology for classification tasks: a meta decomposer framework. Pattern Anal Appl 9(2):257–271 **(Elsevier)**
11. Kumar CN, Rao KN, Govardhan A, Sandhya N (2015) Subset K-means approach for handling imbalanced-distributed data. In: Emerging ICT for bridging the future—proceedings of the 49th annual convention of the Computer Society of India CSI, Springer, vol 2, pp 497–508
12. Shim K (2012) MapReduce algorithms for big data analysis. Proc VLDB Endow 5(12):2016–2017 **(ACM)**
13. Polat K, Güneş S (2009) A new feature selection method on classification of medical datasets: kernel F-score feature selection. Expert Syst Appl 36(7):10367–10373
14. Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD (2008) Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. Neural Netw 21(2):427–436 **(Elsevier)**
15. Partovi FY, Anandarajan M (2002) Classifying inventory using an artificial neural network approach. Comput Ind Eng 41(4):389–404 **(Elsevier)**
16. Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77
17. Chen Y, Raab F, Katz R (2014) From tpc-c to big data benchmarks: a functional workload model. In: Specifying big data benchmarks, Springer, pp 28–43
18. Pal A, Agrawal S (2014) An experimental approach towards big data for analyzing memory utilization on a Hadoop cluster using HDFS and MapReduce. In: Networks & soft computing (ICNSC), IEEE, pp 442–447
19. Dittrich J, Quiané-Ruiz JA (2012) Efficient big data processing in Hadoop MapReduce. Proc VLDB Endow 5(12):2014–2015 **(ACM)**
20. del Río S, López V, Benítez JM, Herrera F (2014) On the use of MapReduce for imbalanced big data using random forest. Inf Sci 285:112–137 **(Elsevier)**
21. Krawczyk B, Galar M, Jeleń Ł, Herrera F (2016) Evolutionary under sampling boosting for imbalanced classification of breast cancer malignancy. Appl Soft Comput 38:714–726 **(Elsevier)**
22. Ibarguren I, Pérez JM, Muguerza J, Gurrutxaga I, Arbelaitz O (2015) Coverage-based resampling: building robust consolidated decision trees. Knowl Based Syst 79:51–67 **(Elsevier)**
23. Geiß C, Pelizari PA, Marconcini M, Sengara W, Edwards M, Lakes T, Taubenböck H (2015) Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques. ISPRS J Photogramm Remote Sens 104:175–188 **Elsevier**
24. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recognit 48(5):1623–1637 **(Elsevier)**
25. Zhang J, Wong JS, Li T, Pan Y (2014) A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems. Int J Approx Reason 55(3):896–907 **(Elsevier)**
26. Nayak J, Naik B, Behera HS (2016) A novel nature inspired firefly algorithm with higher order neural network: performance analysis. Eng Sci Technol Int J 19(1):197–211