Intelligent Automation
& Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2022.028804
Article

Deep Learning Model for Big Data Classification in Apache Spark Environment

T. M. Nithya[1,*], R. Umanesan[2], T. Kalavathidevi[3], C. Selvarathi[4] and A. Kavitha[5]

[1]Department of Computer Science and Engineering, K. Ramakrishnan College of Engineering, Trichy, Tamilnadu, 620009, India
[2]Department of Information and Communication Engineering, Anna University, Chennai, Tamilnadu, 600025, India
[3]Department of Electronics and Instrumentation Engineering, Kongu Engineering College, Erode, Tamilnadu, 638060, India
[4]Department of Computer Science and Engineering, M. Kumarasamy College of Engineering, Karur, Tamilnadu, 639113, India
[5]Department of Electronics and Communication Engineering, K. Ramakrishnan College of Engineering, Trichy, Tamilnadu, 620009, India
*Corresponding Author: T. M. Nithya. Email: nithyacsekrce@gmail.com

Received: 18 February 2022; Accepted: 14 April 2022; Published: 11 September 2023

Abstract: Big data analytics is a popular research topic due to its applicability in various real time applications. The recent advent of machine learning and deep learning models can be applied to analyze big data with better performance. Since big data involves numerous features and necessitates high computational time, feature selection methodologies using metaheuristic optimization algorithms can be adopted to choose optimum set of features and thereby improves the overall classification performance. This study proposes a new sigmoid butterfly optimization method with an optimum gated recurrent unit (SBOA-OGRU) model for big data classification in Apache Spark. The SBOA-OGRU technique involves the design of SBOA based feature selection technique to choose an optimum subset of features. In addition, OGRU based classification model is employed to classify the big data into appropriate classes. Besides, the hyperparameter tuning of the GRU model takes place using Adam optimizer. Furthermore, the Apache Spark platform is applied for processing big data in an effective way. In order to ensure the betterment of the SBOA-OGRU technique, a wide range of experiments were performed and the experimental results highlighted the supremacy of the SBOA-OGRU technique.

Keywords: Big data; apache spark; classification; feature selection; gated recurrent unit; adam optimizer

1 Introduction

One of the current study issues in business intelligence is analyzing massive data and extracting values from it. Therefore, the researchers are limited to academic world but are also extensively employed in other domains like commerce, science, and technology [1]. Big data are made from heterogeneous and numerous resources, involving emails, online transactions, social networking sites, video recordings, audio recordings, etc. Each company/organization that generates this data must examine and manage it [2]. A large data collection that makes typical data analysis and management procedures for obtaining, managing, analyzing, and storing meaningful data difficult is defined as big data [3]. In order to estimate big data, it is important to use an appropriate analyses tool. This urgent need to control the increasing number of data have led to the main concern in developing suitable big data framework. Significant researches have studied several big data fields, for example, management, infrastructure, mining, security, data searching, and so on. Big data frameworks have been designed for identifying analytics to use versatile, quick, and reliable computation design, provide effective quality attributes involving accessibility, flexibility, and resource pooling with ease-of-use and on-demand [4]. These constantly expanding requirements are critical in improving large-scale industry data analytics frameworks.

When running any algorithms on large amounts of data, the need for appropriate big data analysis frameworks becomes evident. Typically, a single Central processing unit (CPU) is used in a local system; however, as data gathering and dataset size grow, multiple core Graphics processing units (GPU) are increasingly being used to improve performance. Parallel processing could be performed quite easier because of the fundamental distributed framework, however GPU is not often economically available/feasible; so, the requirements remain for the tool that utilizes available CPU in a distributed manner within local system [5]. Many publicly available technologies harness massive data volumes by allocating computation and data storage through a cluster of computers. It included many small projects, for example, MapReduce, Hadoop distributed file system (HDFS), Hive, HBase, Pig, and so on.

The rest of the paper Section 2 elaborates the literature review made on the proposed research in the past two decades, Section 3 discusses the proposed model with detailed methodologies, Section 4 presents the results and discussion with findings and Section 5 discusses research conclusion.

2 Literature Study

Classification stands out as one of the most computationally difficult machine learning (ML) problems [6]. Furthermore, in the age of big data, even basic operations like hash mapping/single inner product may take longer due to a large number of operands involved and their dispersion across a shared file system. Furthermore, most classifiers' repetitive training strategy results in a higher number of computations [7]. To enhance accuracy, feature transformation/selection might take place beforehand of the actual classification [8]. Even though this pre-processing phase requires time, the resultant feature set might have much lesser attributes that might more demonstrated the several classes in an easily separable manner. Consecutively, this results in an effective classification method based on overall computational and accuracy time [9]. This study designs a new sigmoid butterfly optimization algorithm with optimal gated recurrent unit (SBOA-OGRU) model for big data classification in Apache Spark environment [10–12]. The SBOA-OGRU technique involves the design of SBOA based feature selection technique to choose an optimum subset of features [13–15]. In addition, OGRU based classification model is employed to classify the big data into appropriate classes. Besides, the hyperparameter tuning of the Gated Recurrent Unit (GRU) model takes place using Adam optimizer. For verifying the supremacy of the SBOA-OGRU technique, a comprehensive simulation analysis is carried out and the experimental results highlighted the supremacy of the SBOA-OGRU technique.

3 The Proposed SBOA-OGRU Technique

In this work, a new SBOA-OGRU technique for big data categorization on the Apache Spark platform is developed. The SBOA-OGRU technique has two stages: Feature Selection (FS) based on SBOA and categorization based on OGRU. Additionally, the Apache Spark platform is used to efficiently process large amounts of data. The next sections go through how these modules work in detail.

3.1 Apache Spark Tool

The Apache Spark has been distributed computing environment utilizing under the big data condition which is developed most powerful structures. The spark provides a unified and whole architecture for controlling the varying necessities to big data handling with variation of datasets (text data, video/image, graph data, and so on) in distinct sources (real-time streaming, batch). The spark structure is generated for overcoming these issues of Hadoop structure based on their creator. Actually, the Spark architecture is demonstrated that performing more than Hadoop during several conditions (in excess of 100 times in memory). An image feature has image design dependent upon that it could explain the image with see. An important plat of features under the image biomedical classifier is for transforming visual data as to vector spaces. Therefore, it could be carried out mathematical functions on them and define same vectors. For performing FS, an initial problem has been detecting features on biomedical images. The amount of them is distinct according to an image, thus it can be more any clauses for making feature vector continuously have a similar size [16]. Afterward, it can generate vector descriptors dependent upon features; all descriptors have a similar size. During this technique, it must be noticeable the feature extraction in the unlabeled/labeled images is carried out with several images under the big data contexts in terms of distinct V of big data (volume, velocity, variety, variability, and veracity). But, the efficiencies of Spark structure are reduced from any conditions: particularly in the feature extraction, during the condition in which there are any small images from the dataset (unlabeled or labeled biomedical images). Other samples are when the size of image regarded as to varying in one another, it is reason an unbalanced loading from the Spark. For solving this issue, the researchers established 2 techniques: order in feature extraction and feature extraction with segmentation. The execution of another 2 techniques has solved this issue of unbalanced loading and running time of all jobs is also similar. This procedure of classifier is a function which is happening if novel unlabeled information derives from the models. The Apache Spark framework is depicted in Fig. 1.

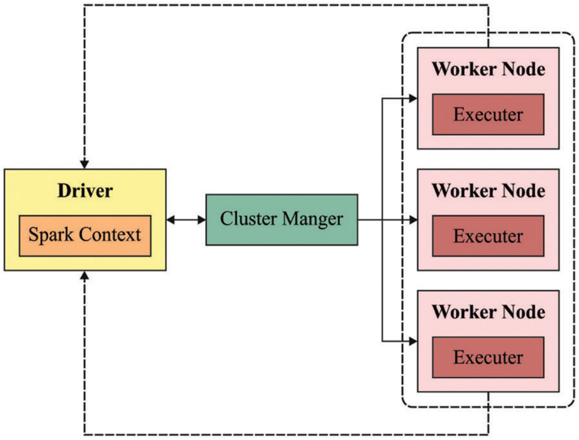

Figure 1: Architecture of Apache Spark

During this technique, combined predictions as well as query are carried out in similar MapReduce stage. With implementing the Spark structure, there derives a benefit to work in big data platforms and utilize their embedding libraries as MLlib (Machine Learning libraries).

3.2 Algorithmic Design of SBOA-FS Technique

BOA imitates the butterfly searching performance. The Butterfly Optimization Algorithm (BOA) shows competitive outcomes with respect to avoidance, convergence, exploitation, and exploration of local optimum, related to another optimized technique. In BOA, superior to other optimize techniques, carries out in most famous function with combined single-exponential as well as multi-exponential distribution. The basic ability of BOA was performance and maximum rate of convergence because of utilizing an arbitrary, decentralized step. In BOA, a butterfly is generating scent. These scents are compared with fitness of butterfly that is computed employing an objective function of issue. If the butterfly transfers from one place to another place in the search spaces, their fitness is altering similarly. The scent that is created by butterflies is felt by distinct butterflies present from the region and collection of social learning models was designed. If the butterfly sense fragrance in the optimum butterfly from the search space, it generates a stride near that butterfly and this phase is named the global search stage of BOA [17]. During the second state, if the butterfly could not capable of detecting the scent of any other butterfly from the search spaces, it can generate arbitrary stride and this phase is termed as local search stage. In BOA, the fragrance has been expressed in Eq. (1):

$$pf_i = cI^a \tag{1}$$

The butterfly is a place vector which is upgraded in the optimized procedure utilizing in Eq. (2):

$$x_i^{(t+1)} = x_i^t + F_i^{(t+1)} \tag{2}$$

There are 2 essential steps in BOA technique: global search step as well as local search step. These steps are expressed in Eqs. (3) and (4) as:

$$F_i^{(t+1)} = \left(r^2 \times g^* - x_i^t\right) \times pf_i \tag{3}$$

$$F_i^{(t+1)} = \left(r^2 \times x_j^t - x_k^t\right) \times pf_i \tag{4}$$

where x_j^t and x_k^t are jth and kth butterflies in the solution spaces. When x_j^t and x_k^t goes to the similar populations and r represents the uniform arbitrary probabilities 0 and 1 afterward Eq. (4) develops the local random walk. The switch probability p has been utilized for switching amongst general global to local searches.

For solving the feature selection (FS) issue, a novel version of BOA, SBOA, is projected that utilized a sigmoid (S-shaped) function making the butterfly for moving in binary search spaces. This sigmoid function is provided in Eq. (5):

$$S\left(F_i^k(t)\right) = \frac{1}{1 + e^{-F_i^k(t)}} \tag{5}$$

where $F_i^k(t)$ refers the continuous-valued fragrance of ith butterflies from the kth dimensional at iterations t.

Afterward, the stochastic threshold has been implemented as stated in Eq. (6) for reaching the binary solution against sigmoid functions. The S-shape function map the infinite input efficiently to finite outcome.

$$X_i^k(t+1) = \begin{cases} 0, & if\ rand < S\left(F_i^k(t)\right) \\ 1, & otherwise \end{cases} \tag{6}$$

where $X_i^k(t)$ and $F_i^k(t)$ signifies the place and fragrance of ith butterfly at iterations t from kth dimensional.

According to the binary nature of FS issue, the search agents were limited to binary [0,1] values only. Therefore, all the solutions from SBOA are considered as single dimension vectors in that the length of vector is dependent upon the amount of features. All the cells of vector are comprised of one or zero. Value one illustrates that the equivalent feature was selected but the value zero demonstrates that the feature could not choose.

Hence, the FS is regarded as multi-objective optimized issue. In SBOA, optimum solutions contain a minimal amount of features with maximum classifier accuracy. Therefore, the fitness function (FF) of this technique is expressed in Eq. (7):

$$Fitness \ = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|} \tag{7}$$

where $\gamma_R(D)$ implies the classification error rate of classifiers, $|R|$ implies the cardinality of elected feature subsets, $|N|$ indicates the entire amount of features from the original datasets, α, and β demonstrated that 2 parameters equivalent to the significance of classifier quality and subset length; $\alpha \in [0, \ 1]$ and $\beta = (1 - \alpha)$ are assumed in the literature.

This FF has been utilized from every optimized technique for evaluating the solution with generating balance amongst classifier accuracy and the amount of elected features.

3.3 Data Classification Using OGRU Model

Once the features were selected, the next step is the data classification process where the data instances are allotted to distinct class labels using OGRU model. Recurrent Neural Network (RNNs) are appropriate for non-linear time-series modeling. The RNN has input layer x, hidden layer h, and resultant layer y. If controlling time-series data, the RNN has been unfolded as exact portion. The resultant as well as hidden layers are computed based on Eqs. (8) & (9) correspondingly.

$$y_t = g\left(\mathrm{s}_t * w_{hy}\right) \tag{8}$$

$$\mathrm{s}_t = f\left(x_t * w_{\mathrm{sx}} + \mathrm{s}_{t-1} * w_{\mathrm{ss}}\right) \tag{9}$$

In spite of, their popularity as universal function nearer and simple execution, the RNN technique has been handled with gradient vanishing or exploding issues. During the trained procedure of RNNs, gradients have computed in the resultant layer to initial layer of RNN. When the gradients are lesser than one, the gradient of the initial many layers is developed small with several multiplications. Conversely, the gradient has been developed extremely huge when the gradient is superior to one. So, it sometimes reasons the gradient for developing nearly 0 or extreme huge if it gains the primary layer of RNN. Accordingly, the weight of initial layer is not attain upgraded during the trained procedure. So, easy RNNs could not be appropriate to any difficult issues.

During this work, the GRU-based technique has been presented for dealing with multivariate time-series imagery information which resolves the vanishing gradient issue of a classic RNN [18]. As demonstrated in Fig. 2, according to preceding output h_{t-1} and present input x_t, the reset gate has been utilized for determining that part of data must be reset as computed in Eq. (10), but an upgraded gate has been utilized for upgrading the outcome of GRU h_t, as computed in Eq. (11). The candidate hidden layer has been computed based on Eq. (12). The present outcome is attained based on Eq. (13). The gates such as z_t and r_t, and parameters like W_z, W_r and W of GRU has been upgraded during the trained procedure.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{10}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{11}$$

$$h_t^{'} = \tan h(W \cdot [r_t * h_{t-1}, x_t]) \tag{12}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h_t^{'} \tag{13}$$

For optimally adjusting the hyperparameters involved in the GRU model, the Adam optimizer is applied to it.

In the Adam algorithm, the exponential decaying average of past gradient m_k and past squared gradient v_k is taken into account as:

$$m_k = \beta_1 m_{k-1} + (1-\beta_1) g_k, \tag{14}$$

$$\mathrm{v_k} = \beta_2 v_{k-1} + (1-\beta_2) g_k^2, \tag{15}$$

whereas g_k represent the gradient, β_1 & β_2 denotes the decay rate, that is closer to 1. Note that m_k & v_k as estimate of the first moment (the mean) and second moment (the uncentered variance) of the gradient, correspondingly [19]. This bias is counteracted with the bias-corrected 1st and 2nd moment estimate, as

$$\hat{m} = \frac{\mathrm{m_k}}{1-\beta_1^{\mathrm{k}}}, \tag{16}$$

$$\hat{\mathrm{v}}_{\mathrm{k}} = \frac{\mathrm{v_k}}{1-\beta_2^{\mathrm{k}}}. \tag{17}$$

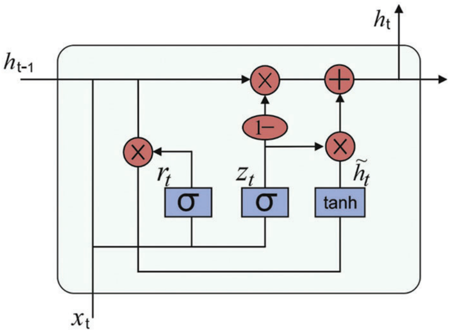

Figure 2: GRU structure

Therefore, Adam update rule was given by:

$$w^{(k+1)} = w^{(k)} - \alpha \cdot \frac{\hat{m}}{\sqrt{\hat{v},k + \delta}}, \tag{18}$$

In which δ represents the smoothing term utilized for avoiding division by zero. The evaluation process of Adam approach is summarized as in Algorithm 1.

Algorithm 1: Adam algorithm

Data: given the initial value $w^{(0)} = w_0$, the number of samples n, the step size α, and the tolerance ε. Set $k = 0$.

Step 1: compute the augmented objective function.

Step 2: evaluate the stochastic gradient.

Step 3: set the arbitrary index j.

Step 4: calculate the decaying average of past and past squared gradients.

Step 5: compute the bias corrected first moment estimates.

Step 6: upgrade the vector $w^{(k)}$. When $\|w^{(k+1)} - w^{(k)}\| < \varepsilon$, then end the iteration. Or else, set $k = k + 1$, and repeat from Step 1.

Remark:

The default value for the decay rate is $\beta_1 = 0.9$ & $\beta_2 = 0.999$, the tolerance is

$\varepsilon = 10^{-6}$, as well as the learning rate is $\alpha = 0.001$.

4 Experimental Validation

On the Epsilon and ECBDL14-ROS datasets, the SBOA-OGRU approach is tested for performance validation. The former dataset holds 400000 training and 100000 testing instances. Besides, the latter dataset comprises 65003913 training and 2897917 testing instances. The number of features under Epsilon and ECBDL14-ROS datasets are 2000 and 631, respectively. The findings of the SBOA-FS approach on the two datasets are shown in Table 1 and Fig. 3. According to the results, the SBOA-FS technique selected a set of 1051 features from a total of 2000 features on the Epsilon dataset. A total of 349 features out of 631 are chosen on the ECBDL14-ROS dataset.

Table 1: Selected features of proposed SBOA-FS method

Dataset	Total features	SBOA-FS
Epsilon	2000	1051
ECBDL14-ROS	631	349

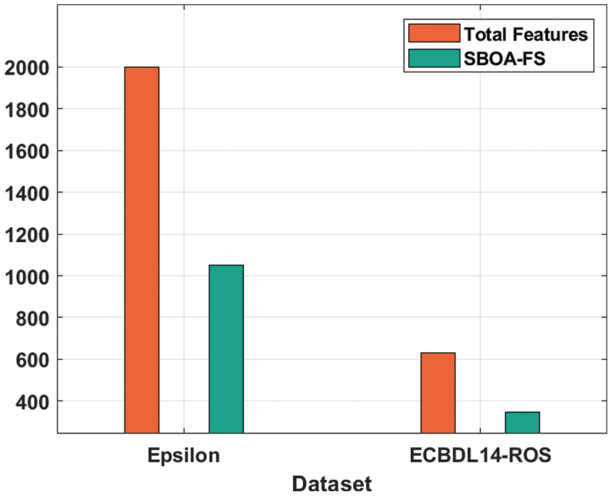

Figure 3: FS analysis of SBOA-FS model

The AUC analysis of the proposed model on the applied Epsilon dataset is given in Table 2 and Fig. 4. The results portrayed that the SBOA-FS under DL-OGRU technique has gained maximum AUC over the other techniques. With SBOA-FS approach, the Deep Learning–Optimal Gated Recurrent Network (DL-ORGU) model has obtained a higher AUC of 96.87% whereas the SVMC, LRC, and NBC techniques have attained a lower AUC of 86.34%, 88.91%, and 90.48%, respectively. At the same time, without feature selection approach, the DL-ORGU model has resulted from an increased AUC of 96.87% whereas the Support Vector Mapping Convergence (SVMC), Logistic Regression Convergence (LRC), and Naïve Bayes Convergence (NBC) techniques have accomplished a reduced Area Under Curve (AUC) of 59.00%, 62%, and 64%, respectively.

The Training Runtime (TRT) analysis of the presented technique with existing techniques on the test Epsilon dataset is provided in Table 3 and Fig. 5. The results depicted that the SBOA-FS under DL-OGRU technique has accomplished reduced TRT over the other techniques. With SBOA-FS approach, the DL-ORGU model has resulted in a decreased TRT of 167.71 s whereas the SVMC, LRC, and NBC techniques have accomplished an increased TRT of 210.61, 272.45, and 187.20 s, respectively. Besides, without feature selection approach, the DL-ORGU model has demonstrated effective outcomes with the minimal TRT of 302.63 s whereas the SVMC, LRC, and NBC techniques have attained a maximum TRT of 400.38, 430.48, and 340.42 s, respectively.

Table 2: AUC results of various classifiers on epsilon dataset

FS Methods	SVMC	LRC	NBC	DL-OGRU
SBOA-FS	86.34	88.91	90.48	96.87
MR-EFS	68.00	70.00	72.00	82.51
Sequential CHC	65.00	67.00	68.00	78.64
Without feature selection	59.00	62.00	64.00	76.43

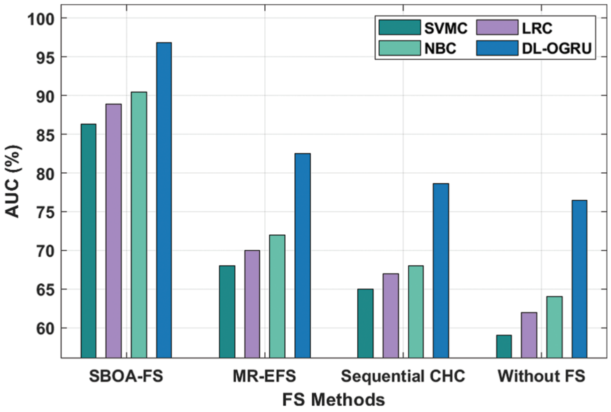

Figure 4: AUC analysis of DL-OGRU model on Epsilon dataset

Table 3: Training runtime (in seconds) of various classifiers on epsilon dataset

FS methods	SVMC	LRC	NBC	DL-OGRU
SBOA-FS	201.61	272.45	187.20	167.71
MR-EFS	334.18	367.29	264.26	232.16
Sequential CHC	345.27	398.07	300.21	259.14
Without feature selection	400.38	430.48	340.42	302.63

The AUC analysis of the presented technique on the applied ECBDL14-ROS dataset is offered in Table 4 and Fig. 6. The outcomes depicted that the SBOA-FS under DL-OGRU technique has reached maximal AUC over the other algorithms. With SBOA-FS technique, the DL-ORGU system has achieved an enhanced AUC of 94.88% whereas the SVMC, LRC, and NBC algorithms have reached a minimum AUC of 85.72%, 89.34%, and 91.57% correspondingly. Concurrently, without feature selection manner, the DL-ORGU methodology has resulted from a superior AUC of 80.09% whereas the SVMC, LRC, and NBC manners have accomplished a decreased AUC of 56%, 58%, and 61% correspondingly.

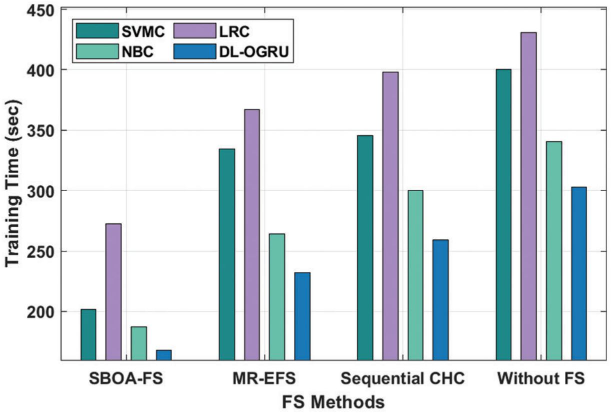

Figure 5: Training runtime analysis of DL-OGRU model on Epsilon dataset

Table 4: AUC results of various classifiers on ECBDL14-ROS dataset

FS methods	SVMC	LRC	NBC	DL-OGRU
SBOA-FS	85.72	89.34	91.57	94.88
MR-EFS	64.00	64.00	67.00	80.40
Sequential CHC	62.00	63.00	65.00	78.64
Without feature selection	56.00	58.00	61.00	80.09

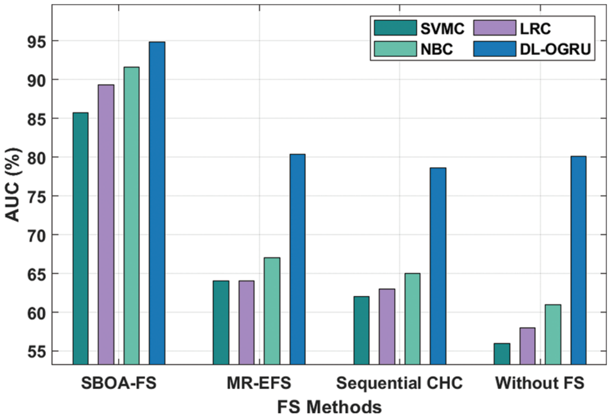

Figure 6: AUC analysis of DL-OGRU model on ECBDL14-ROS dataset

The TRT analysis of the projected approach with state-of-art approaches on the test ECBDL14-ROS dataset is provided in Table 5 and Fig. 7. The outcomes showcased that the SBOA-FS under DL-OGRU manner has accomplished minimum TRT over the other approaches. With SBOA-FS manner, the DL-ORGU system has resulted in a minimal TRT of 108.50 s whereas the SVMC, LRC, and NBC methodologies have accomplished an enhanced TRT of 776.49, 715.32, and 143.76 s correspondingly. In addition, without feature selection approach, the DL-ORGU system has exhibited effectual outcomes with the minimal TRT of 356.89 s whereas the SVMC, LRC, and NBC algorithms have gained an increased TRT of 978.37, 1012.47, and 369.98 s correspondingly.

Table 5: Training runtime (in seconds) of various classifiers on ECBDL14-ROS dataset

FS methods	SVMC	LRC	NBC	DL-OGRU
SBOA-FS	776.49	715.32	143.76	108.50
MR-EFS	864.28	978.38	215.09	210.68
Sequential CHC	912.40	986.45	300.26	297.13
Without feature selection	978.37	1012.47	369.98	356.89

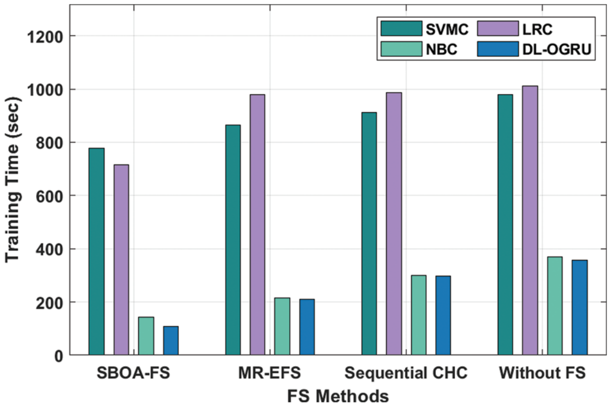

Figure 7: Training runtime analysis of DL-OGRU model on ECBDL14-ROS dataset

By looking into the above results and discussion, it is evident that the proposed model is found to be superior to other methods and it can be utilized for big data classification in real time environment.

5 Conclusion

In this work, a new SBOA-OGRU technique for big data categorization on the Apache Spark platform is developed. The SBOA-OGRU technique has two stages: FS based on SBOA and categorization based on OGRU. The SBOA technique is mostly used to select an optimal subset of characteristics. Moreover, the data classification process where the data instances are allotted to distinct class labels using OGRU model. The Adam optimizer is used to change the hyperparameters in the GRU model to their optimal values. Additionally, the Apache Spark platform is applied for processing big data in an effectual way. A wide range of studies was carried out to validate the SBOA-OGRU technique's supremacy, and the experimental results demonstrated the SBOA-OGRU technique. The above experimental details showed that the proposed model offers maximum performance on the applied Epsilon and ECBDL14-ROS Dataset. The simulation outcome indicated that the DL-ORGU model has achieved maximum AUC values of 94.88. Density-based clustering strategies could improve the SBOA-OGRU model's performance in the future.

Acknowledgement: The author extending their thanks to K. Ramakrishnan College of Engineering for providing the research facilities to complete the article.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: T. M. Nithya, R. Umanesan, T. Kalavathidevi; data collection: C. Selvarathi and A Kavitha;

analysis and interpretation of results: T. M. Nithya, T. Kalavathidevi; draft manuscript preparation: T. M. Nithya, C. Selvarathi. All authors reviewed the results and approved the final version of the manuscript.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. A. Khan, M. Karim and Y. Kim, "A two-stage big data analytics framework with real world applications using spark machine learning and long Short-term memory network," *Symmetry*, vol. 10, no. 10, pp. 485–496, 2018.

[2] L. R. Nair and S. D. Shetty, "Applying spark based machine learning model on streaming big data for health status prediction," *Computers & Electrical Engineering*, vol. 65, no. 1, pp. 393–399, 2018.

[3] L. Hbibi and H. Barka, "Big data: Framework and issues," in *Proc. of Int. Conf. on Electrical and Information Technologies (ICEIT 2016)*, Tangier, Morocco, pp. 485–490, 2016.

[4] M. Assefi, E. Behravesh, G. Liu and A. P. Tafti, "Big data machine learning using apache spark MLlib," in *Proc. of the 2017 IEEE Int. Conf. on Big Data*, Boston, MA, USA, pp. 3492–3498, 2017.

[5] A. Abbasi, S. Sarker and R. H. Chiang, "Big data research in information systems: Toward an inclusive research agenda," *Journal of the Association Information Systems*, vol. 17, no. 2, pp. 1–33, 2016.

[6] A. Alexopoulos, G. Drakopoulos, A. Kanavos, P. Mylonas and G. Vonitsanos, "Two-step classification with SVD preprocessing of distributed massive datasets in Apache Spark," *Algorithms*, vol. 13, no. 71, pp. 1–24, 2020.

[7] D. J. Hand, H. Mannila and P. Smyth, *Principles of Data Mining*. Cambridge, I Edition, MA, USA: MIT Press, 2001.

[8] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[9] C. Lian, S. Ruan and T. Denoeux, "An evidential classifier based on feature selection and two-step classification strategy," *Pattern Recognition*, vol. 48, no. 7, pp. 2318–2327, 2015.

[10] H. Kadkhodaei, A. M. E. Moghadam and M. Dehghan, "Classification of big datasets using heterogeneous ensemble classifiers in Apache Spark based on MapReduce paradigm," *Expert Systems with Applications*, vol. 183, no. 11, pp. 115369, 2021.

[11] W. C. Sleeman and B. Krawczyk, "Multi-class imbalanced big data classification on Spark," *Knowledge-Based Systems*, vol. 212, no. 4, pp. 106598, 2021.

[12] C. Fernandez-Basso, M. D. Ruiz and M. J. Martin-Bautista, "Spark solutions for discovering fuzzy association rules in Big Data," *International Journal of Approximate Reasoning*, vol. 137, no. 3, pp. 94–112, 2021.

[13] C. Gong, Z. G. Su, P. H. Wang, Q. Wang and Y. You, "Evidential instance selection for K-nearest neighbor classification of big data," *International Journal of Approximate Reasoning*, vol. 138, no. 1, pp. 123–144, 2021.

[14] Y. Xu, H. Liu and Z. Long, "A distributed computing framework for wind speed big data forecasting on Apache Spark," *Sustainable Energy Technologies and Assessments*, vol. 37, pp. 100582–100596, 2020.

[15] C. C. Lin, J. R. Kang, Y. L. Liang and C. C. Kuo, "Simultaneous feature and instance selection in big noisy data using memetic variable neighborhood search," *Applied Soft Computing*, vol. 112, no. 1, pp. 107855, 2021.

[16] S. Salloum, R. Dautov, X. Chen, P. X. Peng and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics*, vol. 1, no. 3, pp. 145–164, 2016.

[17] Z. Sadeghian, E. Akbari and H. Nematzadeh, "A hybrid feature selection method based on information theory and binary butterfly optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 97, no. 5, pp. 104079–104093, 2021.

[18] L. Bi, G. Hu, M. M. Raza, Y. Kandel, L. Leandro *et al.,* "A gated recurrent units (GRU)-based model for early detection of soybean sudden death syndrome through time-series satellite imagery," *Remote Sensing*, vol. 12, no. 21, pp. 3621–3640, 2020.

[19] A. Chakrabarty, C. Danielson, S. A. Bortoff and C. R. Laughman, "Accelerating self-optimization control of refrigerant cycles with bayesian optimization and adaptive moment estimation," *Applied Thermal Engineering*, vol. 197, pp. 117335–117354, 2021.