Section A-Research paper
ISSN 2063-5346

# MAPREDUCE BASED HYPER PARAMETER OPTIMISED EXTREME LEARNING MACHINE FOR BIG DATA CLASSIFICATION

**Issa Mohammed Saeed Ali[1] and Dr. D. Hariprasad[2]**
[1]Research Scholar, Sri Ramakrishna College of Arts and Science, India
[2]HOD of BCA, Sri Ramakrishna College of Arts and Science, India
*Corresponding author's: Issa Mohammed Saeed Ali: issamohammed.cs@gmail.com

**Abstract**

The challenging problem of classifying big data, exemplified by volume, velocity, variety, value, and veracity, is very relevant in all fields of study. Additionally, the real-time datasets also suffer from another problem of imbalanced class data, which is common among various application-based data such as healthcare, intrusions, fraud detection, stock prediction, weather forecasting, etc. To handle big imbalanced data, this paper presents an advanced parallel programming-based classifier using an optimised Extreme Learning Machine (ELM) classifier. The proposed MapReduce-based Hyper-parameter Optimized ELM (MR-HPOELM) is developed by integrating the MapReduce framework into the classification model along with optimising the ELM configurations by tuning the hyperparameter values using Black Widow Optimization (BWO). The proposed MR-HPOELM also uses a hybrid feature selecting method by combining the Information Gain (IG) and Population and Global Search Improved Squirrel Search Algorithm (PGS-ISSA). This hybrid feature selection approach reduces the feature dimension using information gain and then selects the optimal feature subset using PGS-ISSA. Finally, the MR-HPOELM obtains these selected features and classifies the data accurately, irrespective of the class imbalance problem. Implemented in MATLAB, this hybrid model is tested on benchmark datasets where outcomes demonstrated that for the voluminous class unbalanced datasets, the suggested IG-PGS-ISSA and MR-HPOELM based classification models produced excellent classification accuracies with reduced computations.

**Keywords:** Big data classification, high dimensional data, Class imbalance problem, Information gain, Improved Squirrel Search Algorithm, MapReduce, Optimized Extreme Learning Machine.

## 1. INTRODUCTION

Fast developments in information technologies have given rise to five V's of big data. [1], making it complex to handle voluminous data effectively. Big data are extremely complicated and voluminous, making it challenging to handle them using traditional methods of data processing [2]. Feature selection and data classification are considered the major tasks, and there are numerous algorithms in the literature for both tasks. Selecting the most suitable approach for each task is the primary objective for performing efficient big data classification [3]. Machine Learning (ML) algorithms are considered the most efficient algorithms for data mining applications. With the large volume of big data, ML algorithms still possess better performance [4]. Although Deep Learning (DL) algorithms have been much more accurate in big data classification, the requirement of large training data and subsequent longer training time are the disadvantages of DL [5]. Another major problem faced in most big data classification approaches is the class imbalanced data problem [6]. In such cases, the samples in each class of the dataset are different, and mostly, the difference in the numbers of samples is huge. Likewise, the number of features is limited compared to the sample size, leading to an imbalanced data problem. To overcome this problem, different sampling methods were utilised in traditional approaches. Although they are efficient, the sampling methods tend to provide biased selection and lead to erroneous decisions in the classification stage [7]. The modern classifiers have been developed with an additional constraint of handling the imbalanced classification. The traditional algorithms, such as decision trees, logistic regression, etc., have often lacked this characteristic to handle the class imbalance problems [8].

Classifiers such as Support Vector Machines and ELM can provide high classification performance for big datasets [9]. Yet, the ELM classifier also suffers from this class imbalance problem, but optimally selecting its weights to model the weighted ELM can solve the class imbalance problem and provide highly accurate classification [10]. However, handling such big data is still a problem when the processing is carried out on a single system as the storage and handling of the big data increase the computation complexity. Therefore, the need for developing classifiers for the faster processing of big data is being increasingly explored in the research field. Improving the speed of the algorithms can be a viable solution. But, it might not be sufficient for all types of datasets and might also lead to premature convergence. To overcome such limitations, parallel processing strategies and frameworks are often used to implement the classification algorithms such that the speed of processing is improved without impacting the outcomes [11]. MapReduce is one of the most efficient parallel processing frameworks that can be used to enhance the processing speed of any task. Although developed for Hadoop architecture, this framework has been adopted in all implementation types due to its wide range of applications [12].

This paper aims to design advanced parallel programming-based classifiers using MapReduce and ELM classifiers. The proposed MR-HPOELM is developed by optimising the ELM configurations by tuning the hyper-parameter values using BWO. In the proposed approach, the MapReduce framework is employed to provide parallel programming of the larger data to minimise the processing time. For feature selection, a novel hybrid optimisation algorithm is developed by merging the filter and wrapper methods of IG and PGS-ESSA. This hybrid feature selection performs feature dimension reduction using IG and selects the top feature subsets optimally using PGS-ISSA. HPOELM is introduced by tuning the hyper-parameter values of the ELM classifier. The hidden layers, hidden nodes and learning rate parameters of ELM are optimally tuned by BWO to obtain the optimal configuration of ELM with better training time and less complexity. The rest of the article is divided into two sections: "Related works" in section 2, and "Explanation of the proposed methodology using MR-HPOELM" in section 3. The results of the experiments are in Section 4, and suggestions for future work are in Section 5.

## 2. RELATED WORKS

Many MLAs and DLTs have been utilised for classification problems in big data. Zhai et al. [13] proposed the ELM ensemble (MR-FI-ELME) was made possible by the MapReduce architecture for balancing big data categorisation. Non-iterative learning and oversampling were achieved by employing the nearest neighbour technique in this model. The MR-FI-ELME then classifies the data with an accuracy rate of 94.21 %. The ELM parameters can still be tuned using this manner, Yet, this method has limitations in tuning the ELM parameters. Lakshmanaprabu et al. [14] developed a hybrid technique using RFC (Random forest classifier) and MapReduce for big data classification. The scheme selected optimal attributes using IDA (Improved Dragonfly Algorithm). This model's accuracy was tested against the e-health dataset and it was at 94%. However, the method's random decision trees added to complexities while training BRNN (bidirectional recurrent neural networks) for classifications of voluminous unbalanced data. Hassib et al. [15] suggested WOA (Whale optimisation algorithm) for classifying Iris dataset instances but even with smaller feature subsets their suggested WOA-BRNN models executed for seven hours though they achieved 94% accuracy.

Kadkhodaei et al. [16] developed heterogeneous ensemble classifiers based on MapReduce to classify the big datasets. The ensemble classifiers of heterogeneous boosting (HBoost) and distributed heterogeneous boosting (DHBoost) algorithms are used in this study. The evaluations on seven big datasets showed that the DHBoost achieved high accuracy. For the Higgs dataset, it achieved 75.78% and the CEN dataset with 96.73%. However, thig method consumes more training time.Narayana et al. [17] developed ACSODeep RNN (Ant Cat Swarm Optin 2zation-enabled Deep Recurrent Neural Networks) enabled by the MapReduce framework. Black hole entropy fuzzy clustering based on Pearson correlations and Deep RNNs were tailored in the study to perform well with ACSO. This model achieved 89.3% accuracy, 90% sensitivity and 88.4% specificity on the Cleveland dataset. But this model requires high training time to learn all the temporal features.Akhtar et al. [18] presenteda dragonfly rider optimisation algorithm-based recurrent neural networks called DROA-RNN for big data classifications. This model achieved 99.6% accuracy, 99.5% sensitivity and 99.5% specificity on benchmark datasets.

M Mujeeb et al. [19] suggested DBN (Deep Belief Networks) feature selection and AEBA (Adaptive Exponential Bat algorithm) for voluminous data categorizations. On UCI datasets, this model achieved a higher accuracy of 89.98% and a higher TPR of 91.44%. This model, however, was unable to pick up on a number of crucial traits, which lowered accuracy across all datasets. For the categorization of huge data, Hassanat et al. [20] suggested the MF (Magnetic Force) Classifier. It was used to analyse 28 datasets and has good accuracy ratings of 98%. On Higgs data, it has a 60% accuracy rate, but it is also quite sensitive to class skewness. AHSHO-IELM (Archerfish Hunter Spotted Hyena Optimisation-based Improved

Extreme Learning Machine) classifier-based MapReduce framework for Big Data Classification was created by Chidambaram and Gowthul Alam[21]. This model used ELM in combination with PCA (Principal Component Analysis) to develop solutions to multicollinear problems. The accuracy, specificity, and sensitivity of this model were 99 %, 99 %, 98.3 %, and 99 %,99 percent, and 98 %, respectively, were examined using the datasets for dermatology and rotten tomatoes movie reviews. The model's ability to handle the data from the unbalanced classes, however, also has como traints. For identifying the large, highdimensional datasets, He et al. [22] introduced a Bayesian attribute bagging-based extreme learning machine (BAB-ELM). On benchmark examples, this model had regression error that was 96% lower and classification accuracy that was 96% higher. However, this model fails to handle the skewed class data.

From the literature, few limitations of the big data classification algorithms have been identified. The major limitations are the higher training time, high model complexity and computational complexity of the classifiers along with the inability to handle the class imbalanced data. Also, it is inferred that the parallel processing using MapReduce has provided better processing speed for most methods. This research focuses on solving such problems for the classification of different datasets using the MapReduce framework and advanced ELM classifier.

## 3. METHODS

This work's suggested schema for classifying big data includes stages of preprocessing, IG and PGS-ISSA based feature selection and MR-HPOELM based classification methods. Figure 1 displays the flow of the suggested model.

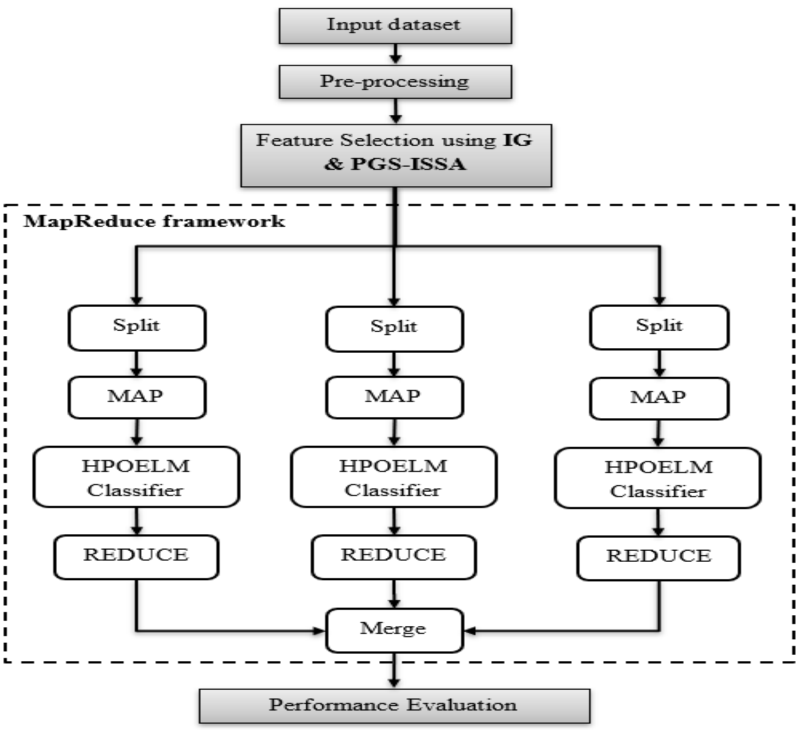


**Fig.1. Functional flow diagram of MR-HPOELM based big data classification**

### 3.1. Datasets and Pre-processing

This works classification model was tested using UCI library's six benchmark datasets. the datasets were sized based on the quantity of occurrences where Higgs dataset (With 1100000) was the largest. Table 1 contains information about the datasets used in this study.

**Table.1. Datasets details**

| Datasets | Number of attributes | Number of instances |
|---|---|---|
| Higgs | 28 | 1100000 |
| Glass | 9 | 214 |
| Diabetes | 20 | 768 |
| Cleveland | 14 | 303 |
| Vehicle | 18 | 846 |
| Wine | 13 | 178 |

The pre-processing methods of noise identification, data cleaning and data normalization are performed as a standard procedure. After pre-processing, the features are extracted from the given datasets. Then features are selected for obtaining most informative features.

### 3.2. Feature selection using IG and PGS-ISSA

Feature selection in this study is performed using a hybrid method by combining the Information gain (IG) method [23] and PGS-ISSA [24]. PGS-ISSA algorithm has been developed as an improved version of the standard Squirrel Search Algorithm. By collaboratively using these two methods, the features are selected faster with high efficiency. In this hybrid model, the IG is used inside the PGS-ISSA algorithm and is used as the metric to estimate the importance of the features. Then the PGS-ISSA based optimisation strategy ranks the features based on the IG values and returns the top-ranked features for the classification.

Initially, the squirrels population is assembled and the initial parameters are defined along with the maximum number of iterations. Here each squirrel will resemble a feature subset. Then the IG is computed for these individuals using Eq. (3) and the process will be terminated if the maximum IG value is obtained. IG was used to calculate feature's effectiveness in voluminous data to obtain higher classification accuracy. It is estimated as the amount of information via the reduction in entropy. Entropy measures attribute variety and aids in identifying information impurities, allowing one to assess the degree of uncertainty associated with predictions made using a specific variable. So, in order to compute the IG, the entropy is first defined. Let y denote the discrete random variable attribute with two possible outcomes i.e. relevant ($R$) and irrelevant ($\bar{R}$) to the ideal features. The binary function H can be expressed as a logarithmic value.

$$H(y) = -p(R)\log p(R) - p(\bar{R})\log p(\bar{R}) \quad (1)$$

Here $(R, \bar{R})$ denotes the possible classes- relevant and irrelevant, $p(R)$ denote the probability of the sample being $y \in (R)$ and $p(\bar{R})$ denote the probability of the sample being $y \in (\bar{R})$. Conditional entropy defines the quantity of the uncertainties of each feature in the decision process and it is computed between two events X and Y where X has the value of feature $x$,

$$H(Y|X) = \sum_{x \in X} p_x(x)\, H(Y|X = x)$$

$$= \sum_{x \in X} p_x(x) \; \sum_{y \in Y} p(y|x) \log p_y(y|x)$$

$$= \sum_{x \in X} \sum_{y \in Y} p_{xy}(x, y) \log p_y(y|x) \tag{2}$$

The smaller values of the impurity will result in more skewed class distributions. The values of entropy and the misclassification errors will be the highest when the class distribution is uniform and the minimum value of entropy is obtained when all the samples belong to the same class.

The IG of $y$ can be computed using the entropy and conditional entropy from a feature $x$ as

$$IG(y|x) = H(y) - H(y|x) \tag{3}$$

Larger IG defines the higher discriminative power for the decision process and determines the relevance of the features with respect to the classification problem. In order to rank these features based on their relevance, the optimisation strategy of PGS-ISSA is applied. The feature selection process is summarised in Algorithm 1.

**Algorithm 1: IG and PGS-ISSA based feature selection**

Define input parameters, iterations

Map the available feature subsets as food sources

Generate chaotic locations for the inflight squirrels

Estimate the IG for each inflight squirrel

Rank the squirrels based on IG

Determine the locations of the squirrels

Remove the bottom-ranked squirrels

Rearrange the squirrels' population

Recalculate the IG for each squirrel

Compare the IG and determine the best squirrel locations

Update the positions of squirrels moving to normal, acorn and hickory trees

Section A-Research paper
ISSN 2063-5346

Compute the seasonal constant

Check the stopping criteria

Return the best locations of the squirrels (feature subsets)

### 3.3. Classification using MR-HPOELM

The ELM configuration was optimised and implemented on MapReduce frameworks in MR-HPOELM where the distributed programming technique (MapReduce) uses

concurrent distributed clusters of nodes to analyze enormous volumes of data. Mapping and Reducing are two phases of the processes where mapping stages are transformation phases, while reduction stages are summarization stages. The proposed HPOELM reformulates classifier training from a single training network to a number of tiny networks with equivalent training using the MapReduce architecture.

The training data, which contains the data batch and labels, is sent to the mappers as input. The label is the ground truth label for each instance, while the batch is the training data. Initially, a set of ideal weights that reflect the network's hidden layers is established using BWO. The system is then trained using the input data samples after each mapper optimises the network weights. The set of recently trained weights contained in the set of optimum weights determined by the BWO [25] is the output. The reduction, which combines the weights, takes these outputs as input. The average weight is then calculated by adding together all the overweights and dividing it by the batch's training case count. The results become the base for weight updates which are then transmitted to mappers.

Using the BWO method, the proposed HPOELM is optimised by selecting the hyperparameters' ideal values. The number of hidden layers, number of hidden layer nodes, and learning rates are the hyperparameters that are best chosen together with weight and bias using the BWO. Using the mapping function, the input featurs obtained from the IG and PGS-ISSA technique are transferred to the hidden layer $H = a(Wx + b)$ where W denotes the input weight matrix, $a(.)$ denotes the activation function and b represent the bias vector. The hidden layer is plotted into the remodelled input vector $\hat{x} = a(WH + b)$.By reducing the remodelling mistakes between the real input and encoded results, the HPOELM parametes are trained. The estimated function may be discovered by computing the output weights for the N training features with input X and output O of different dimensions. Two procedures make up the training phase: teast-squares constraint solving and random mapping. The buried layer is constructed using random neurons that are mapped using the sigmoid function during the random mapping procedure and shown below

$$H(x) = \{1 + \exp[-(W^T x + b)]\}^{-1} \quad (4)$$

The goal is to choose an ELM configuration that minimises error and improves precision without affecting the complexity of any tested data [26]. In order to accomplish this goal, the multi-objective optimisation problem is formulated as

$$\min F(X) = |f_1(x), f_2(x)\ | \tag{5}$$

Where $f_1(x) = Error$; $f_2(x) = Complexity$

For minimising the optimisation problem, the initial values of weights, bias, and the number of hidden layers, hidden nodes and learning rate are optimally tuned. It is formulated as $(W, b, H, \xi, \alpha\ )$. BWO selects the optimal values for this model. BWO begins with the establishment of a spider population. After then, these original spiders mate off in an effort to spawn a new generation of spiders. During the mating process, the female spiders eat the male spider. The female releases the eggs after transporting the sperm to them. Sperm will

cause egg laying, and new spiderlings will be produced. These spiderlings live together on their mother's web, and the stronger spiderlings are eaten by the mother spiders (cannibalism) in order to get rid of them and save the stronger spiders first. The BWO procedure is described in Algorithm 2.

**Algorithm 2: BWO algorithm**

Population spiders are initialised using a D-dimensional array ().

Change Iteration to 0

Based on the reproduction rate, ascertain the quantity of reproductions (number of solutions).

Choose the best options for the population and store them in subpopulation 1/

// Engage in cannibalism and reproduction

For each $i = 0\ to\ nr$ do

From the sub-population-1, selecting two parents$(x_1, x_2)$

Create D offspring $(y_1, y_2)$ i.e. new archived solutions using

$$\begin{cases} y_1 = \theta \times x_1 + (1-\theta) \times x_2 \\ y_2 = \theta \times x_2 + (1-\theta) \times x_1 \end{cases} \tag{6}$$

Destroy the father's solution

Estimate fitness $f(y) = \min(W, b, H, \xi, \alpha\ )$

Eliminate a few weak kids (new remedies) based on the cannibalization rate

Store the remaining solutions in sub-population-2

End for

//Perform mutation

Calculate the number of offspring with mutations ($nm$) using the mutation rate.

For $i = 0\ to\ nm$ do

Choose a solution from sub-population-2.

Utilize random mutation to provide a new solution.

Store the new solution in sub-population-3

End for

Update population = sub-population-2 + sub-population-3

Until the maximum number of iterations

Return the optimal response from the population

End

In this model, the different values of $(W, b, H, \xi, \alpha)$ are given as input to the BWO and the fitness is evaluated. The best sequence $(W, b, H, \xi, \alpha)$ that achieved low error and low complexity is chosen and stored by the BWO. Based on these optimal values, the hidden layers are formulated.

In this hidden layer, the output vector is modelled as $H(x) = R^{N\times\xi}$, where r is the dimension of variables and $\xi$ denotes the number of hidden nodes. The output can be modelled as

$$\hat{o}_n = H(x_n)\gamma, \quad n = 1,2, \dots N \tag{7}$$

Here $\gamma$ is the output weight of the hidden layer. It can be achieved by reducing the cost function $\Gamma_{ELM}$

$$\min_{\gamma \in R^{H\times r}} \Gamma_{ELM} = \left\|O - \hat{O}\right\|^2 = \|O - H\gamma\|^2 \tag{8}$$

The output weight can be modelled using least-squares constraints solving to obtain the modified output weights $W'$.

$$\gamma = H^{\dagger}O \tag{9}$$

Where $H^{\dagger}$ denote the Moore–Penrose generalised inverse of H. The training process of the HPOELM is performed which uses the samples $(X, Y)$ and employs the fully connected multi-layered network structure.

Similar to the SVM classifier, the proposed MR-HPOELM also utilises different kernel functions to improve the feature learning process. It adaptively selects the best kernel function based on the given problem.

## 4. PERFORMANCE EVALUATION

The proposed MR-HPOELM is implemented using MATLAB R2016b over six benchmark datasets. The performance of the proposed MR-HPOELM classifier is evaluated and compared with the classifiers from the literature namely MR-FI-ELME [13], WOA-BRNN [15], DHBoost [16], DROA-RNN [18] and AHSHO-IELM [21]. The complete datasets were given as inputs, and the training and testing data ratio was set as 70:30. The proposed model also used the IG + PGS-ISSA for feature selection. Hence, the performance of the proposed method was evaluated with and without this feature selection method. The performance metrics used are accuracy, recall, f-measure, precision, specificity and time. The obtained results are given in Table 2 for 20 independent runs.

**Table.2. Performance comparison**

| Metrics | Dataset | MR-FI-ELME | WOA-BRNN | DH Boost | DROA-RNN | AHSHO-IELM | Proposed MR-HPOELM | IG+PGS-ISSA with MR-HPOELM |
|---|---|---|---|---|---|---|---|---|
| **Accuracy (%)** | Higgs | 90.21 | 92.67 | 82.67 | 96.5 | 97.43 | 98.5 | **99.32** |
| | Glass | 94.22 | 93.66 | 88.78 | 96.11 | 97.59 | 97.36 | **98.38** |
| | Diabetes | 90.13 | 91.56 | 89.23 | 95.65 | 96.91 | 96.89 | **97.57** |
| | Cleveland | 93.67 | 95.79 | 86.98 | 96.5 | 97.95 | 98.16 | **99.01** |
| | Vehicle | 94.5 | 96.18 | 90.35 | 94.2 | 97.27 | 97.51 | **98.03** |
| | Wine | 94.12 | 97.25 | 92.12 | 98.11 | 99.12 | 99.4 | **99.81** |
| **Precision (%)** | Higgs | 90.88 | 93.67 | 86.75 | 93.78 | 94.11 | 95.38 | **96.12** |
| | Glass | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Diabetes | 91.5 | 93.76 | 89.55 | 94.67 | 95.89 | 96.99 | **98.58** |
| | Cleveland | 92.67 | 94.89 | 90.2 | 95.8 | 96.27 | 99.1 | **100** |
| | Vehicle | 91.75 | 94.88 | 91.56 | 96.91 | 97.88 | 97.97 | **98.21** |
| | Wine | 92.8 | 93.14 | 90.48 | 96.5 | 97.33 | 98.89 | **99.5** |
| **Recall (%)** | Higgs | 91.88 | 94.67 | 85.87 | 95.88 | 94.67 | 96.5 | **97.64** |
| | Glass | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Diabetes | 90.77 | 92.44 | 87.71 | 93.91 | 94.11 | 94.67 | **95.21** |
| | Cleveland | 91.1 | 93.65 | 88.65 | 92.5 | 95.78 | 96.83 | **98.0** |
| | Vehicle | 92.61 | 90.35 | 86.45 | 94.78 | 96.11 | 97.36 | **98.24** |
| | Wine | 94.29 | 94.17 | 89.53 | 95.88 | 96.26 | 97.75 | **98.95** |
| **F-measure (%)** | Higgs | 91.38 | 94.17 | 86.31 | 94.82 | 94.39 | 95.94 | **96.87** |
| | Glass | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Diabetes | 91.13 | 93.09 | 88.62 | 94.29 | 94.99 | 95.82 | **96.87** |
| | Cleveland | 91.88 | 94.27 | 89.42 | 94.12 | 96.02 | 97.95 | **98.99** |
| | Vehicle | 91.72 | 92.55 | 89.63 | 96.39 | **97.8** | 97.10 | 97.34 |
| | Wine | 93.54 | 93.66 | 90.0 | 96.19 | 96.79 | 98.32 | **99.22** |
| **Specificity (%)** | Higgs | 95.78 | 94.76 | 91.22 | 94.28 | 94.91 | 96.34 | **97.11** |
| | Glass | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Diabetes | 93.89 | 95.67 | 90.72 | 94.88 | 94.16 | 96.03 | **97.62** |
| | Cleveland | 95.78 | 96.81 | 92.44 | 95.76 | 97.21 | 98.83 | **100** |
| | Vehicle | 95.71 | 96.44 | 91.5 | 96.52 | 97.99 | 98.5 | **99.5** |
| | Wine | 94.66 | 96.67 | 90.27 | 94.76 | 95.91 | 97.59 | **98.53** |
| **Time (seconds)** | Higgs | 87.78 | 78.67 | 98.5 | 77.29 | 75.5 | 61.8 | **51.09** |
| | Glass | 4.88 | 4.21 | 5.01 | 3.85 | 3.79 | 3.28 | **3.07** |
| | Diabetes | 12.76 | 11.65 | 13.55 | 11.84 | 11.5 | 11.02 | **10.76** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Cleveland | 4.98 | 4.67 | 5.21 | 4.32 | 4.44 | 3.95 | **3.74** |
| | Vehicle | 7.83 | 7.61 | 8.39 | 7.14 | 7.32 | 6.67 | **6.08** |
| | Wine | 2.86 | 2.91 | 3.08 | 2.68 | 2.79 | 2.53 | **2.211** |

From the above table, it can be seen that the proposed MR-HPOELM has achieved 98.5% accuracy for Higgs data while utilising the IG+PGS-ISSA feature selection the accuracy increased to 99.32%. It has improved big data classification more than the other models. Compared with the other models, the proposed MR-HPOELM with the hybrid feature selection has increased accuracy, recall, f-measure, precision, specificity and reduced the execution time. For the huge Higgs dataset, MR-HPOELM with IG+PGS-ISSA feature selection has a high accuracy which is 0.82%, 1.89%, 2.82%, 16.65%, 6.65% and 9.11% higher than the proposed MR-HPOELM without feature selection, AHSHO-IELM, DROA-RNN, DHBoost, WOA-BRNN and MR-FI-ELME models, respectively. Likewise, MR-HPOELM with IG+PGS-ISSA feature selection has improved the precision, recall and f-measure values. It also consumed less processing time of 51.09 seconds for Higgs data for the big data classification which is lesser than all the other compared models. Similar results have also been obtained for the other benchmark datasets. This indicates that the MR-HPOELM classifier with IG+PGS-ISSA feature selection has much better efficiency for the diverse big data problems.

## 5. CONCLUSION

In this paper, a parallel computing framework based classification model is presented to reduce the processing time in handling the large datasets. To achieve this objective, the MapReduce framework was integrated with the optimised ELM classifier for reducing the complexity issues. In the optimised ELM model, the hyper parameters along with the weights are optimally tuned to eliminate the impact of the class imbalance problem. The proposed MR-HPOELM is also equipped with hybrid feature selection using the IG+PGS-ISSA model which combines the benefits of the filter and wrapper methods. Using these strategies, the classification of big datasets was performed accurately and with less complexity. This model was evaluated on benchmark datasets and the results showed that the proposed MR-HPOELM classifier achieved high accuracy, recall, f-measure, precision, specificity and less processing time than the state-of-the-art methods. In future, the possibility of integrating the online evaluation models for ELM will be investigated. Additionally, the ability of the proposed algorithms to handle the multi-source and multi-format big data will also be analysed in future.

## REFERENCES

1. Buhl, H. U., Röglinger, M., Moser, F., & Heidemann, J. (2013). Big data. *Business & Information Systems Engineering*, *5*(2), 65-69.
2. Jadhav, D. K. (2013). Big data: the new challenges in data mining. *International Journal of Innovative Research in Computer Science & Technology*, *1*(2), 39-42.
3. Pramanik, P. K. D., Mukhopadhyay, M., & Pal, S. (2021). Big data classification: Applications and challenges. In *Artificial Intelligence and IoT* (pp. 53-84). Springer, Singapore.

4. Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350-361.

5. Yang, M., Nazir, S., Xu, Q., & Ali, S. (2020). Deep learning algorithms and multicriteria decision-making used in big data: a systematic literature review. *Complexity*, *2020*.

6. Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, *5*(1), 1-30.

7. Rendon, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., & Granda-Gutierrez, E. E. (2020). Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, *10*(4), 1276.

8. Madasamy, K., & Ramaswami, M. (2017). Data imbalance and classifiers: Impact and solutions from a big data perspective. *International Journal of Computational Intelligence Research*, *13*(9), 2267-2281.

9. Wang, X., Liu, X., & Matwin, S. (2014). A distributed instance-weighted SVM algorithm on large-scale imbalanced datasets. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 45-51). IEEE.

10. Wang, Z., Xin, J., Yang, H., Tian, S., Yu, G., Xu, C., & Yao, Y. (2017). Distributed and weighted extreme learning machine for imbalanced big data learning. *Tsinghua Science and Technology*, *22*(2), 160-173.

11. Tsai, C. F., Lin, W. C., & Ke, S. W. (2016). Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. *Journal of Systems and Software*, *122*, 83-92.

12. Ahlawat, K., Chug, A., & Singh, A. P. (2019). Empirical evaluation of map-reduce based hybrid approach for problem of imbalanced classification in big data. *International Journal of Grid and High Performance Computing (IJGHPC)*, *11*(3), 23-45.

13. Zhai, J., Zhang, S., Zhang, M., & Liu, X. (2018). Fuzzy integral-based ELM ensemble for imbalanced big data classification. *Soft Computing*, *22*(11), 3519-3531.

14. Lakshmanaprabu, S. K., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. *International journal of machine learning and cybernetics*, *10*(10), 2609-2618.

15. Hassib, E., El-Desouky, A., Labib, L., & El-kenawy, E. S. M. (2020). WOA+ BRNN: An imbalanced big data classification framework using Whale optimisation and deep neural network. *Soft Computing*, *24*(8), 5573-5592.

16. Kadkhodaei, H., Moghadam, A. M. E., & Dehghan, M. (2021). Big data classification using heterogeneous ensemble classifiers in Apache Spark-based on MapReduce paradigm. *Expert Systems with Applications*, *183*, 115369.

17. Narayana, S., Chandanapalli, S. B., Rao, M. S., & Srinivas, K. (2021). Ant Cat Swarm Optimization-Enabled Deep Recurrent Neural Network for Big Data Classification Based on MapReduce Framework. *The Computer Journal*.

18. Akhtar, M. M., Ahamad, D., & AlamHameed, S. (2021). Optimisation algorithm-based recurrent neural network for big data classification. *International Journal of Intelligent Information and Database Systems*, *14*(2), 153-176.

19. Md Mujeeb, S., Praveen Sam, R., & Madhavi, K. (2021). Adaptive Exponential Bat algorithm and deep learning for big data classification. *Sādhanā*, *46*(1), 1-15.

20. Hassanat, A. B., Ali, H. N., Tarawneh, A. S., Alrashidi, M., Alghamdi, M., Altarawneh, G. A., & Abbadi, M. A. (2022). Magnetic Force Classifier: A Novel Method for Big Data Classification. *IEEE Access*, *10*, 12592-12606.

21. Chidambaram, S., & Gowthul Alam, M. M. (2022). An Integration of Archerfish Hunter Spotted Hyena Optimization and Improved ELM Classifier for Multi-collinear Big Data Classification Tasks. *Neural Processing Letters*, 1-29.

22. He, Y., Ye, X., Huang, J. Z., & Fournier-Viger, P. (2022). Bayesian Attribute Bagging-Based Extreme Learning Machine for High-Dimensional Classification and Regression. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *13*(2), 1-26.

23. Azhagusundari, B., & Thanamani, A. S. (2013). Feature selection based on information gain. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *2*(2), 18-21.

24. Ali, I. M. S., & Balakrishnan, M. (2021) Population and Global Search Improved Squirrel Search Algorithm for Feature Selection in Big Data Classification. International Journal of Intelligent Engineering and Systems, 14(4), 177-189.

25. Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimisation algorithm: a novel meta-heuristic approach for solving engineering optimisation problems. *Engineering Applications of Artificial Intelligence*, *87*, 103249.

26. Huang, G. B., Ding, X., & Zhou, H. (2010). Optimisation method based extreme learning machine for classification. *Neurocomputing*, *74*(1-3), 155-163.