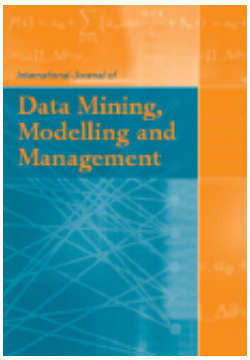


International Journal of Data Mining, Modelling and Management

ISSN online: 1759-1171 - ISSN print: 1759-1163
https://www.inderscience.com/ijdmmm

Hybrid classifier model for big data by leveraging map reduce framework

V. Sitharamulu, K. Rajendra Prasad, K. Sudheer Reddy, A.V. Krishna Prasad, M. Venkat Dass

DOI: 10.1504/IJDMMM.2024.10057054

Article History:

Received:	14 December 2022
Last revised:	29 March 2023
Accepted:	07 May 2023
Published online:	22 January 2024

Hybrid classifier model for big data by leveraging map reduce framework

V. Sitharamulu

Department of Computer Science and Engineering,
GITAM School of Technology,
GITAM (Deemed-to-be-University),
Hyderabad, India
Email: vsitaramu.1234@gmail.com

K. Rajendra Prasad

Department of CSE,
Institute of Aeronautical Engineering,
Hyderabad, India
Email: krprgm@gmail.com

K. Sudheer Reddy*

Department of Information Technology,
Anurag University,
Hyderabad, India
Email: sudheercse@gmail.com
*Corresponding author

A.V. Krishna Prasad

Department of Information Technology,
MVSR Engineering College,
Hyderabad, India
Email: kpvambati@gmail.com

M. Venkat Dass

College of Engineering,
Osmania University,
Hyderabad, India
Email: venkatdass@osmania.ac.in

Abstract: Big data technology is popular and desirable among many users for handling, analysing, and storing large data. However, clustering the large data has become more complex due to its size. In recent years, several techniques have been presented to retrieve the information from big data. The proposed hybrid classifier model CSDHAP, the hybridised form of sun flower optimisation (SFO) and deer hunting optimisation (DHO) algorithms with

adaptive pollination rate using MapReduce framework. The CSDHAP is a data classification technique that performed using classifiers. The results of the presented approach are evaluated over the extant approaches using various metrics namely, F1-score, specificity, NPV, accuracy, FNR, FDR, sensitivity, precision, FPR, and MCC. It is pertinent to mention that, the proposed model is better than any of the traditional models. The proposed HC+CSDHAP model attained better precision value than other traditional models like RNN, SVM, CNN, Bi-LSTM, NB, LSTM, and DBN, correspondingly.

Keywords: big data classification; MapReduce framework; long short-term memory; LSTM; deep belief network; DBN; optimisation.

Reference to this paper should be made as follows: Sitharamulu, V., Prasad, K.R., Reddy, K.S., Prasad, A.V.K. and Dass, M.V. (2024) 'Hybrid classifier model for big data by leveraging map reduce framework', *Int. J. Data Mining, Modelling and Management*, Vol. 16, No. 1, pp.23–48.

Biographical notes: V. Sitharamuluis is currently working in the Department of Computer Science and Engineering at the School of Technology, GITAM University, Hyderabad, Telangana State, India. He has cumulative triumph record of 22 years of teaching and academic experience in various top ranked engineering colleges across the country. His research interests targeted in publishing five patents and published more than 20 journal articles in Scopus, SCI, WOS, and international journals. His keen areas of domain specialisations include Data mining, machine learning, artificial intelligence, and networks. At the outset, he has guided many more scholars and still guiding his researchers.

K. Rajendra Prasad graduated in CSE from JNTU, postgraduated in CSE from VTU Belgaum, and PhD from JNTUA Anantapur. He is currently working as a Professor in the Department of CSE, Institute of Aeronautical Engineering. He received a project from DST-SERB, worth of 37 lakhs and completed at 2020. He has published 60 publications in reputed journals. His research interests are data mining, and big data. He guided four PhD scholars. He is a member of IEEE and CSI professional societies. He is a BoS-member at university boards. He received foundation certifications from GCP and AWS.

K. Sudheer Reddy has about 20 years of experience in education and research. He has obtained his Doctoral degree in Computer Science and Engineering. He has published 46 research papers in reputed journals and conferences. He has successfully completed several funded research projects granted by various funding agencies.

A.V. Krishna Prasad is working as an Associate Professor in the Dept. of Information Technology, Maturi Venkata Subba Rao Engineering College, Hyderabad. He has over 40 publications to his credit, which are published in renowned journals and conferences.

M. Venkat Dass is currently working as an Associate Professor in the Dept. of Computer Science and Engineering, College of Engineering, Osmania University, Hyderabad, India. He has over 25 years of experience in the filed of computer science. He has published over 25 articles in reputed journals.

1 Introduction

The knowledge discovery and data processing are not a simple task in the field of big data. This has become a critical component of organisations' ongoing efforts gain and to extend their intangible resources by analysing the data received from various sources. A wide range of real-world challenges are the real source of number of useful data-mining methods (Visuwasam and Raj, 2020; Venkatesh et al., 2019). In the last two decades, their remark has taken on a different significance. Due to the massive increase in data volume all through the time period, the development in organisations' expectations and needs for gaining the competitive advantage, understanding the commonalities across various sectors is a crucial method to approach data processing (products, events, customers, etc.). These techniques comes at a considerable cost in terms of algorithmic instability and processing time. As a consequence, finding similarities require the emergence of new reliable and efficient procedures (Khalemsky and Gelbard, 2019). With the rapid growth of global data, the 'big data' is widely used to analyse the massive amounts of data and advantages of collecting (Baldán and Benítez, 2018).

Big data is known as an quantity of data which surpasses the processing capability of a specific approach in relation to memory usage and/or time (Hernández et al., 2019). Big data is rapidly growing in a variety of industries including healthcare, government, social media, network applications, as well as financial services (Dessì et al., 2018; Fan et al., 2020). This is due to the gradual accumulation of a massive data quantity that has become more availability of distributed platforms with readily accessible. With these tools, novel prospect on identifying new big data sets is explored, assisting in the discovery of hidden values while posing new obstacles. These issues emerge throughout the data duration and collecting process, including classification, preservation, validation, content generation, and transformation to visualise and data analysis (Kaur et al., 2022; Iyer et al., 2022). Most existing and traditional ML (Thangam et al., 2019; Shaik and Ganesh, 2020) and data mining techniques, including bio-inspired ones (Hababeh et al., 2019; Ma et al., 2021) has used to complete these tasks. As a consequence, new technologies and methodologies must be created and developed to improve the process optimisation (Thomas and Rangachar, 2018; Reddy et al., 2022; Hojage, 2021), decision-making, and insight discovery (Bovenzi et al., 2020; Cai et al., 2021).

A new breed of fault-tolerant resilient methods depending on parallel computing is emerged with the MapReduce method (Yang et al., 2021) serving as the most notable case. Many research articles have been published in this aspect that concentrate on the parallel processing of ML and data mining approaches using the MapReduce architecture. These approaches help in classification, clustering, dimensionality reduction, and mining, which has demonstrated that distributing data and processing in a parallel computing architecture is extremely effective to speed the information extraction method (Jiang and Li, 2019; Elkano et al., 2020). This problem arises when samples of one class far exceed those of the other (Xing and Bei, 2020). Evolutionary techniques for unbalanced massive datasets (Saki et al., 2020) have been developed to address this issue. In terms of technology, these MapReduce evolutionary techniques use resampling methods (Abdel-Hamid et al., 2018) to investigate the impact of modifying the classification performance. Particularly, a MapReduce PSO method (Beno et al., 2014) able to handle the demanding resources provided by the traditional PSO method's individual function assessment processes. The contributions are:

- Uses MapReduce framework for handling the bigdata.
- Implements a combined sunflower and deer hunting model with adaptive pollination rate (CSDHAP) algorithm for training the hybrid classifier via tuning the optimal weights.

The literature is presented in Section 2. The architecture and framework of the proposed methodology is given in Section 3. The implementation details, results and discussions are given in Section 4. Section 5 has conclusion remarks.

The goal of the proposed hybrid classifier model is to enhance the classification performance, weight of both DBN and LSTM (Yadav et al., 2023). The results of the presented approach are evaluated over the extant approaches using various metrics.

2 Literature review

In 2023, Kanani et al. proposed an AI-enabled ensemble method for rainfall forecasting using long short-term memory (LSTM) by using classification.

In 2023, Yadav et al. proposed a method on plant leaf disease detection using CNN with transfer learning and XGBoost by employing clustering.

In 2022, Reddy et al. have investigated and proposed a method called, hybrid generalised aversial network.

In 2020, Ravuri and Vasundra have investigated a method via spark framework for clustering. The method clusters in two phases, comprising feature selection as well as clustering in the first cluster nodes of the spark framework. Originally, the best features were chosen and inserted in the feature vector using the suggested MFO-Bat method, which was created through combining MFO as well as bat techniques. The chosen characteristics were then sent into Spark's final cluster nodes that performs optimum clustering using the sparse FCM approach. With a higher Dice coefficient, classification accuracy, and a Jaccard coefficient, the suggested MFO-Bat has surpassed other extant models.

In 2020, Selvi and Valarmathi have used intelligent approaches to construct the suggested strategy, which would be focused on a big data categorisation system. The parallel pool map reduce framework were utilised to handle large amount of data in this case. Feature extraction, optimum feature selection, and classification are the three key phases of the model. The well-known feature extraction techniques PCA, LDA, and linear square regression were employed for extracting the features. Because feature vectors tend to be long, selecting the best features was a difficult challenge. As a consequence, the suggested model selects the best features using the L-FF method, which would be an optimal features. Additionally, the suggested L-FF+NN result was validated over the proposed approach, demonstrating that it outperforms them. The suggested L-FF+NN model achieved better outcomes than the traditional methods while experimentation.

In 2020, Ducange et al. have looked at several implementation concerns with more recent fuzzy models for addressing big categorisation tasks. The suggested model has looked at multiple distributed deployments of learning models for producing the modelling approach of FDTs and FRBCs. The model was compared with the abovementioned distributed fuzzy classification methods in terms of scalability as well as

accuracy, by considering the results obtained on four common big data classification datasets.

In 2019, González et al. have developed a method for obtaining fuzzy rule classifiers employing a sequential model capable of processing a huge number of instances. Moreover, a sequential model could indeed be aggressive in terms of some problems as well as the ability of learning beside parallel processing plans on the MapReduce framework. Thus, the consecutive processing employs a 'batch-incremental learning approach' that allows every group of instances to be processed independently. The testing revealed that the progressive technique was competitive with a parallel framework introduced for dealing with big data categorisation employing fuzzy rules.

In 2019, CGCNBMRM technique (Banchhor and Srinivasu, 2019) used for classifying big data. As a consequence, the CG-CNB classifier was created by combining the CNB classifier with the recently found optimisation technique CGWO. The CGWO method was introduced by integrating the CS algorithm into GWO to improve the CNB framework through selecting the best model parameters. Furthermore, the suggested CGCNB-MRM technique uses the subsequent data probability and the index table probability to classify each sample data. In the suggested CGCNB-MRM technique, three metrics were used: specificity accuracy, as well as sensitivity. It achieved 76.9% specificity, 84.5% sensitivity, and 80.7 100% accuracy, demonstrating its usefulness in big data categorisation.

In 2018, Zhai et al. have created a promising technique depending on MapReduce, ensemble learning, and oversampling to cope with the challenge of categorising binary unbalanced huge data. With MapReduce, every positive instance's enemy closest neighbour was discovered, and positive examples were randomly created in its enemy nearest neighbour hypersphere with uniform distribution. The suggested result was validated successfully with three similar methods: MR-V-ELM, SMOTE, and SMOTE+RF-Bigdata. Further, the suggested algorithm surpassed the other three different approaches, with respect to the statistical analysis as well as testing findings.

In 2018, García-Gil et al. have studied the first appropriate noise filter in big data environments, wherein higher dimensional space and significant instance redundancy issues offer new hurdles to traditional noise pre-processing techniques. Several data sets were used to evaluate the appropriateness of these suggested strategies in terms of data reduction rate, running durations, and accuracy improvement.

In 2018, Dagdia has focused on the attention of bio-inspired classifier, a DCA due to its limitations while dealing with larger data sets related to big data. Sp-DCA seems as dispersed bio-inspired dendritic cell model for large-scale data categorisation created underneath the Spark architecture. Further, the Spark architecture has provided an effective framework for parallelising the dendritic cell algorithm's operation, enabling it to address storage as well as runtime constraints. The test findings suggest that the adopted distributed method can improve the DCA's performance, allowing it to used for solving huge data classification challenges.

Table 1 summarises the review on big data classification approaches. Originally, the CGCNB-MRM model (Banchhor and Srinivasu, 2019) provides accuracy, higher sensitivity and maximum specificity; however, unsuitable for performance maximisation. The MFO-Bat algorithm (Ravuri and Vasundra, 2020) provides better dice coefficient, and maximal classification accuracy. The application development is insufficient and it requires update. The L-FF algorithm (Selvi and Valarmathi, 2020) offer increased precision, no or less mechanisms to handle noisy data. The fuzzy classification models

(Ducange et al., 2020) produce better accuracy and higher interpretability. The model needs to revisit various areas. MR-FI-ELM (Zhai et al., 2018) has F-measure, higher precision and recall. There is no such concept to handle multiple classifications. The NSLV-AR (González et al., 2019), better running time, accuracy and better scalability. The threshold parameter was significant. HTE-BE (García-Gil et al., 2018) that offer less computing time. There is no such approach for removal of the noisy instances. The Sp-DCA (Dagdia, 2018), has high accuracy, but missing sensitivity. Thus, the challenges were considered while big data classification in the current work.

Table 1 A comparative study

Author	*Methodology*	*Features*	*Challenges*
Banchhor and Srinivasu (2019)	CGCNB-MRM approach	• Better performance	• Unsuitable for maximising the performance.
Ravuri and Vasundra (2020)	MFO-Bat algorithm	• Better dice coefficient • Maximal classification accuracy	• The application development is insufficient.
Selvi and Valarmathi (2020)	L-FF algorithm	• Increasing the preciseness.	• No or less mechanisms to handle the noisy data.
Ducange et al. (2020)	Fuzzy classification models	• Better accuracy • Higher interpretability	• Various areas are to be revisited.
Zhai et al. (2018)	MR-FI-ELM model	• F-measure • Higher precision and recall	• There is no such concept called multiple classifications.
González et al. (2019)	NSLV-AR model	• Running time • Better accuracy and scalability	• Threshold parameter was insignificant.
García-Gil et al. (2018)	HTE-BE algorithm	• Computing time	• There is no such approach for removal of the noisy instances.
Dagdia (2018)	Sp-DCA model	• High accuracy	• The sensitivity is missing.

3 Methodology

The proposed CSDHAP has three phases such as:

1 pre-processing

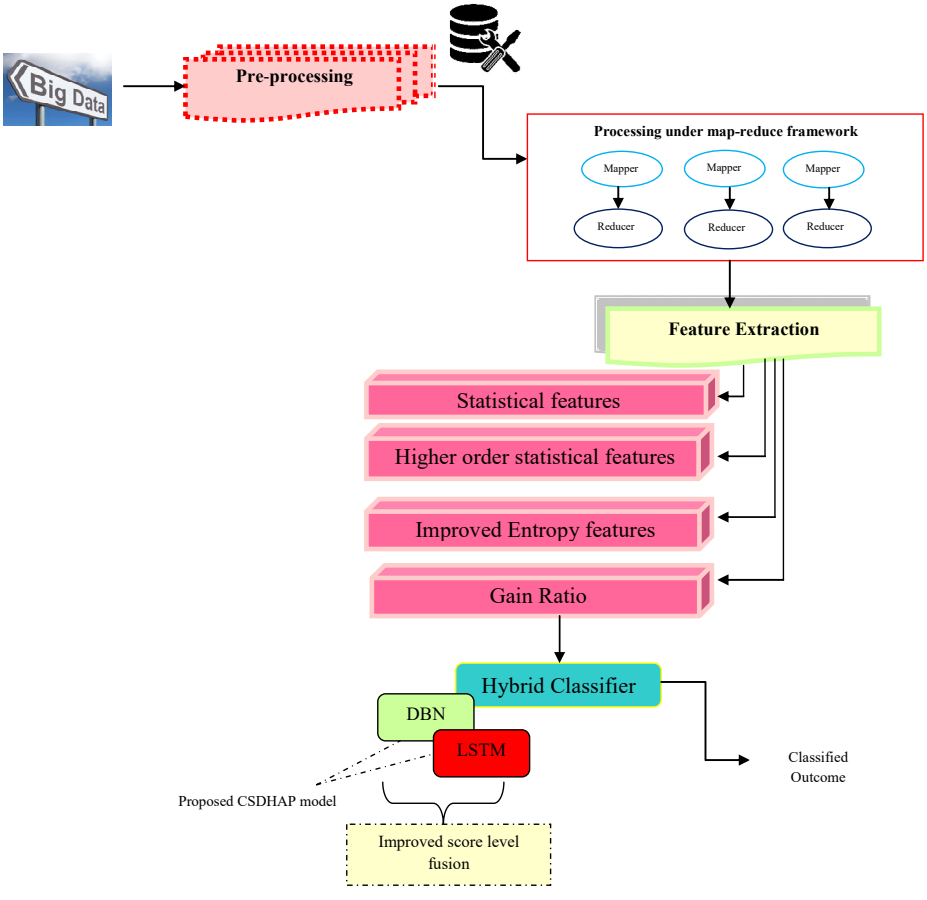
2 MapReduce framework

3 data classification.

The input data is processed in the pre-processing process. The data are then subjected under MapReduce framework (parallel pool). Subsequently, the feature extraction is

performed, in which higher order statistical features, an improved entropy, statistical features, and gain ratio are taken out. Moreover, the extracted features are subjected for classification through LSTM and DBN model. Finally, the output of LSTM and DBN are given to an improved score level fusion. To enhance the classification performance, the weight of both DBN and LSTM are optimised via a proposed CSDHAP algorithm. Figure 1 illustrates the framework of adopted scheme.

Figure 1 The framework (see online version for colours)



3.1 Pre-processing, MapReduce framework and feature extraction phase

3.1.1 Pre-processing phase

The preprocessing is performed under data cleaning process. Moreover, the null and the duplicate values are removed in this preprocessing phase.

3.1.2 MapReduce framework

Many recent 'big data' applications rely on the MapReduce framework (parallel pool). MapReduce (https://in.mathworks.com/help/matlab/import_export/process-big-data-in-

matlab-using-mapreduce.html) is a technique for 'divide and conquer' big data issues. MapReduce used in MATLAB includes three input arguments:

- A datastore used for reading the data.
- 'Mapper' function is expressed a subset of the data to work with. The map function produces a partial computation. MapReduce involves the mapper function for each chunk in the data centre and each call operating independently.
- 'Reducer' function also provided the mapper function's aggregated outputs. The reducer function completes the calculation started by the mapper function and returns the final result.

To a certain extent, this is an oversimplification due to the result of a call to the mapper function could be shuffled and merged in more ways before being handed to the reducer function.

- The 'mapper' function calculates the greatest value from the datastore in each chunk.
- The 'reducer' function calculates the largest value between all maximum determined by the mapper function calls.

To begin, reset the datastore and reduce the variables only to the single column that are interested on. The MapReduce's usage of keys is a crucial and effective feature of MapReduce (parallel pool). Every invocation to the mapper function stores intermediate results with one or more specified 'buckets', known as keys. The number of chunks in the datastore correlates to the number of calls to the mapper function via MapReduce. If the mapper function inserts values to several keys, then the reducer function calls multiple times only on the one key's intermediate values. The MapReduce function handles data flow among the map and reduces stages of the method.

3.1.3 Feature extraction

The pre-processed data has to extract the following features:

1 *Improved entropy*: The conventional entropy is given in equation (1).

$$E = \sum_{t=1}^{u(A_1)} p_t \log p_t \tag{1}$$

The new improved entropy is given in equation (2).

$$IE(m) = -\hat{W} * \sum m(A) \log_2 \left(\frac{m(A)}{2^{|A|} - 1} e^{(|A|-1)/|Y|} \right) \tag{2}$$

where $\hat{W}$ is calculated using tent chaotic map, Y is the frame of discernment (FOD), i.e., loss of information, $|A|$ denotes the cardinality of focal element A, $|Y|$ indicates the number of elements in FOD, and $m(A)$ refers to the mass function.

2 *Statistical features*: The mean, median, variance and std. deviation.

   a *Mean (average)* (https://en.wikipedia.org/wiki/Statistic): The mean can be computed as:

$$\bar{G} = \frac{1}{k}\sum_{q=1}^{k} G_q \tag{3}$$

In equation (3), G is the observed value, k – the count of values, and symbol of sample mean is represented using $\bar{G}$.

b *Median* (https://en.wikipedia.org/wiki/Statistic): The median can be calculated as:

$$Median = \begin{cases} G\left(\frac{k}{2}\right) & \text{if } k \text{ is odd} \\ \frac{G\left(\frac{k-1}{2}\right) + G\left(\frac{k+1}{2}\right)}{2} & \text{if } k \text{ is even} \end{cases} \tag{4}$$

c *SD*: The SD (https://en.wikipedia.org/wiki/Standard_deviation) can be calculated using the given equation (5). σ indicated SD.

$$\sigma = \sqrt{\frac{1}{k-1}\sum_{q=1}^{k}\left(G_q - \bar{G}\right)^2} \tag{5}$$

d *Variance*: It can be computed as given in equation (6).

$$Variance = \frac{\sum\left(G_q - \bar{G}\right)^2}{k-1} \tag{6}$$

The statistical features *SF*, as defined:

$$SF = \bar{G} + Median + \sigma + variance \tag{7}$$

3 Higher order features: Skewness and kurtosis are considered as higher order:

a Skewness (https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm#:~:text=Skewness%20is%20a%20measure%20of,relative%20to%20a%20normal%20distribution):

$$Skewness = \frac{\sum_{q=1}^{k}\left(G_q - \bar{G}\right)^3 \Big/ k}{\sigma^3} \tag{8}$$

In equation (8), $G_q = G_1, G_2, \ldots, G_k$, $\bar{G}$ – mean value, σ is SD, and data points k.

b *Kurtosis* (https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm#:~:text=Skewness%20is%20a%20measure%20of,relative%20to%20a%20normal%20distribution): The formula of kurtosis for univariate data such as $G_1, G_2, \ldots, G_k$, is expressed in equation (9).

$$Kurtosis = \frac{\sum_{q=1}^{k}\left(G_q - \bar{G}\right)^4 \Big/ k}{\sigma^4} \tag{9}$$

The higher order statistical features *HF*, is given as:

$$HF = Skewness + Kurtosis \tag{10}$$

4 *Gain ratio*: Information gain (https://en.wikipedia.org/wiki/Information_gain_ratio) is the minimisation in entropy attained from dividing a group of attributes and identifies the optimal candidate which produces the largest value.

$$IG(g,b) = h(g) - h\left(g \mid b \mid\right) \tag{11}$$

where g is a random variable and $h(g|b|)$ is the entropy of g given the value of attribute b. Moreover, the information gain ratio is defined as the ratio among the information gain to the split information value.

$$IGR(g,b) = IG(g,b) / Split\ Information(g) \tag{12}$$

The overall extracted features are indicated as *FE*, and it is given in equation (13).

$$FE = IE(m) + SF + HF + IGR(g,b) \tag{13}$$

Big data classification by hybrid classifier: combining deep belief network and LSTM

The extracted features are given to the classification phase through optimised HC comprising DBN and LSTM. In this model, the extracted features are given directly to both to have a concurrent computation. The results are processed under improved score level fusion.

3.1.4 Optimised DBN

DBN (Wang et al., 2016) with multiple layers consists of a visible and hidden neuron in each layer. The visible neurons are fully interconnected with the hidden neurons. In nature, the stochastic neuron's outcome is probabilistic in the Boltzmann networks. The DBN framework is an architectural model that includes hidden and visible neurons, as well as numerous layers that make up the output layer.

The output *out* is attained by probability function $L(\psi)$ in equation (14).

$$out = \begin{Bmatrix} 1 & \text{with } L(\psi) \\ 0 & \text{with } 1 - L(\psi) \end{Bmatrix} \tag{14}$$

$$L(\psi) = \frac{1}{1 + e^{\frac{-\psi}{s}}} \tag{15}$$

DBN approach is determined in equation (16).

$$\lim_{s \to 0^+} L(\psi) = \lim_{s \to 0^+} \frac{1}{1 + e^{\frac{-\psi}{s}}} = \begin{cases} 0 & \text{for} \quad \psi < 0 \\ \frac{1}{2} & \text{for} \quad \psi = 0 \\ 1 & \text{for} \quad \psi > 0 \end{cases} \tag{16}$$

The feature processing is done by a set of RBM layers in DBN structure and the classification occurs via MLP. The Boltzmann machine energy of binary state z given in

(17) and (18). Where $W_{\hat{a},\hat{b}}$ indicates the weights between the neurons that is optimally tuned via the novel CSDHAP approach and $\gamma_{\hat{a}}$ specifies the biases.

$$C(z) = -\sum_{\hat{a}<\hat{b}} z_{\hat{a}} W_{\hat{a},\hat{b}} - \sum_{\hat{a}} \gamma_{\hat{a}} z_{\hat{a}} \tag{17}$$

$$\Delta C\left(z_{\hat{a}}\right) = \sum_{\hat{b}} z_{\hat{a}} W_{\hat{a},\hat{b}} + \gamma_{\hat{a}} \tag{18}$$

Energy based on joint composition in hidden and visible neurons (c, f) is determined using (19) to (21), where $f_{\hat{a}}$ and $c_{\hat{a}}$ denotes the binary state of hidden unit, $\hat{b}$ and $\hat{a}$ are the visible unit. $W_{\hat{a},\hat{b}}$ specifies the weight and $U_{\hat{a}}$ and $V_{\hat{b}}$ refers to the biases.

$$C(c, f) = -\sum_{(\hat{a},\hat{b})} W_{\hat{a},\hat{b}} c_{\hat{a}} f_{\hat{b}} - \sum_{\hat{a}} U_{\hat{a}} c_{\hat{a}} - \sum_{\hat{b}} V_{\hat{b}} f_{\hat{a}} \tag{19}$$

$$\Delta C\left(c_{\hat{a}}, f\right) = \sum_{\hat{b}} W_{\hat{a},\hat{b}} f_{\hat{b}} + U_{\hat{a}} \tag{20}$$

$$\Delta C\left(c, f_{\hat{b}}\right) = \sum_{\hat{a}} W_{\hat{a},\hat{b}} c_{\hat{a}} + V_{\hat{b}} \tag{21}$$

The dispersed probabilities and the resultant weight allocations as given (22):

$$W_{(\hat{a})} = \max_{W} \prod_{c \in I} \hat{E}(c) \tag{22}$$

The probability distribution for the vectors pair (c, f)

$$\hat{E}(c) = \frac{1}{X} e^{-C(c,f)} \tag{23}$$

The partition function X as:

$$X = \sum_{c,f} e^{-C(c,f)} \tag{24}$$

The steps in CD method are as follows.

- Select f samples and fix it to visible neurons.
- Compute the probability of hidden neurons $\hat{E}_f$ is given in equation (25), λ is activation function.

$$\hat{E}\left(f_{\hat{b}} \to 1 \mid c\right) = \lambda\left(V_{\hat{b}} + \sum_{\hat{a}} \hat{b}_d W_{\hat{a},\hat{b}}\right) \tag{25}$$

- Probabilities are determined from the hidden state.
- Compute the exterior vectors product z and $\hat{E}_f$ as positive gradient $\varphi^{+} = c.\hat{E}_f^{s}$.
- The restoration of d visible states from f hidden states is defined in equation (26). Furthermore, it is essential to analyse the hidden states f' from c' restoration.

$$\hat{E}\left(c_{\hat{b}} \to 1 \mid f\right) = \lambda\left(U_{\hat{a}} + \sum_{\hat{a}} z_{\hat{b}} W_{\hat{a},\hat{b}}\right) \tag{26}$$

- Examine the exterior product c' and f', using its negative gradient $\varphi^{-} = c'.f'\ s$.
- The weight update is given in (27). β refers to the learning rate.

$$\Delta W = \beta\left(\varphi^{+} - \varphi^{-}\right) \tag{27}$$

- (28) is used for the weight update.

$$W'_{\hat{a},\hat{b}} = \Delta W_{\hat{a},\hat{b}} + W_{\hat{a},\hat{b}} \tag{28}$$

In the MLP model, assign $(\hat{B}^{\tilde{s}}, \hat{A}^{\tilde{s}})$ training patterns. $1 \le \tilde{s} < O$, $\tilde{s}$ refers to the number of training patterns, $\hat{A}^{\tilde{s}}$ the predicted output and $\hat{B}^{\tilde{s}}$ actual output. Moreover, the error evaluation is given (29). The output of DBN is denoted as CL_{DBN}.

$$e^{\tilde{s}} = \hat{B}^{\tilde{s}} - \hat{A}^{\tilde{s}} \tag{29}$$

3.1.5 LSTM

The extracted features (*FE*) are subjected to the LSTM. The LSTM is an effectual approach to elucidate the issues of gradient desertion through implementing a linear connection and gate control unit. Thus, the LSTM network captured the strong dependence among the time-series data.

Yan et al. (2020) present sequences of persistent LSTM cells, which has three units that indicates the 'forget gate, input gate, and output gate'. This element permits the LSTM memory cells to suggest the information for extensive time duration and to stock up. Consider H and S as the hidden and cell state. (H_j, S_j) represent the output and (Z_j, S_{j-1}, H_{j-1}) represents the input layers. At time j the output and input gates, forget gate are denoted as O_j, $\hat{I}_j$, $\widehat{G}_j$ correspondingly. LSTM primarily uses $\widehat{G}_j$ to filter the data. The $\widehat{G}_j$ is given as:

$$\widehat{G}_j = \kappa\left(w_l Z_j + K_l + w_J H_{j-1} + K_J\right) \tag{30}$$

Here, (w_J, K_J) and (w_l, K_l) specifies the bias parameter and weight matrix. Thus, the activation function of gate (κ) is elected as sigmoid operation. Next, the LSTM cell makes use of the input gate to combine the proper data as determined in equations (31), (32) and (33). Where (w_X, K_X) and (w_Y, K_Y) are the weight matrices and bias parameters, that map the input and hidden layers to cell gate. (w_P, K_P) and (w_Q, K_Q) is the weight and bias parameters which map the hidden and input layers to I_j.

$$\hat{U}_j = \tanh\left(w_{\bar{Y}} Z_j + K_{\bar{Y}} + w_{\bar{X}} H_{j-1} + K_{\bar{X}}\right) \tag{31}$$

$$I_j = \kappa\left(w_Q Z_j + K_Q + w_P H_{j-1} + K_P\right) \tag{32}$$

$$S_j = \widehat{G}_j S_{j-1} + I_j \hat{U}_j \tag{33}$$

Finally, LSTM obtains hidden layer (output) from output gate as:

$$T_j = \kappa\left(w_{\bar{B}} Z_j + K_{\bar{B}} + w_{\bar{M}} H_{j-1} + K_{\bar{M}}\right) \tag{34}$$

$$H_j = T_j \tanh\left(S_j\right) \tag{35}$$

where (w_M, K_M) and (w_B, K_B) indicates the weight and bias parameters for mapping the input and hidden layers to T_j. The LSTM output is CL_{LSTM}.

- *Improved score level fusion*: The output of both DBN and LSTM are processed under improved score level fusion. Here, the outcomes are separately computed for each classifier, and the results are mixed into a multimodal system for enhancing the entire system outcomes. The score vectors in the classification phase for both classifiers are calculated separately and normalised. Moreover, the improved score level fusion is determined in equation (36) and equation (37).

$$SC_i = \frac{SC_i - \min_{SC_i}}{\max_{SC_i} - \min_{SC_i}} \tag{36}$$

$$F_{sr} = \sum_{i}^{\hat{m}} \left(W * SC_{1i} + w * SC_{2i}\right) \tag{37}$$

where $\max_{SC_i}$ and $\min_{SC_i}$ indicates the maximum and minimum value of score vector of sample i, SC_i indicates the score normalisation of two classifiers (DBN and LSTM) of sample i, SC_{1i} and SC_{2i} refers to the score value of classifiers, and $\hat{m}$ specifies the number of classifiers.

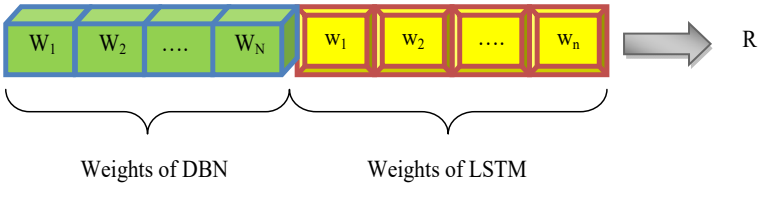
3.2 Weight optimisation of DBN and bi-LSTM via combined sunflower and deer hunting model with adaptive pollination rate

3.2.1 Solution encoding and objective function

The weights of both DBN and LSTM are optimally tuned by the proposed CSDHAP method. Figure 2 presents the input solution to the proposed CSDHAP. Here, the total amount of weights in DBN is indicated as N, and the total number of weighs in LSTM as n. Moreover, the predicted output of both DBN and LSTM are provided to an improved score level fusion to obtain an overall outcome. From this outcome, the accuracy is calculated. Further, the error function is determined as *error* = (1 – *accuracy*). The objective or fitness function of the implemented approach is given (38).

$$Obj = \min(error) \tag{38}$$

Figure 2 Solution encoding (see online version for colours)



3.2.2 DHSFO

Though, the existing sun flower optimisation (SFO) (Gomes et al., 2019) scheme solves the multidimensional and multi-modal problems, the SFO model is required to be sophisticated with certain parameters to obtain enhanced computational outcomes. To overcome this scenario, the deer hunting optimisation (DHO) (Brammya et al., 2019) is integrated with SFO in this work named as CSDHAP model. Generally, the hybrid optimisation approaches are assumed to be more suitable for search problems (Beno et al., 2014).

SFO is a population-based heuristic optimisation model for multi-modal and multidimensional issues. Moreover, the movement of sunflowers for catching solar radiation inspired the SFO. The sunflower cycle is always similar: every day, they accompany and awaken the sun like the clock needles. They travel in opposite direction at night to wait for their departure of the next morning. Sunflowers' peculiar habit is considered by the researchers for best orientation towards the sun. Pollination occurs randomly among the flower a and flower $a + 1$ that has least distance.

The inverse square law radiation is the focus of the nature-based optimisation. According to this law, "the intensity of radiation is inversely proportional to the square of the distance". If the plant is closer to the sun, the radiation is maximised, and it tend to stabilise in the particular region. Furthermore, the plant distance is larger from the sun; the heat is less as it focuses to take higher steps to reach near to the global optimum as feasible. Further, the amount of heat randomly among the flower arriving by every plant is determined in equation (39).

$$B_a = \frac{y}{4\pi r_a^2} \tag{39}$$

In equation (39), r_a the distance between the best and the current plant a and y indicates the power of the source.

The direction of sunflower to the sun is determined in equation (40).

$$x_a = \frac{R^* - R_a}{\left\| R^* - R_a \right\|}, \qquad a = 1, 2, ..., NM. \tag{40}$$

The step of the sunflowers direction is specified in equation (41).

$$D_a = \delta \times y_a \left(\left\| R_a + R_{a-1} \right\| \right) \times \left\| R_a + R_{a-1} \right\| \tag{41}$$

where $\neg\delta$ indicates the constant value, $y_a(\|R_a + R_{a-1})$ specifies the probability of pollination. Individuals near to the sun perform tiny steps of a local refinement, but those far away from the individuals move more often. It is vital to restrict the maximum step supplied via each individual for avoiding the skipping regions. Further, the maximum step is given in equation (42).

$$D_{\max} = \frac{\left\| R_{\max} - R_{\min} \right\|}{2 \times \tilde{N}_{pop}} \tag{42}$$

In equation (42), $R_{\min}$ and $R_{\max}$ are the lower and upper bounds values, $\tilde{N}_{pop}$ specifies count of plants in entire population. The new plantation is determined in equation (43).

$$R_{a+1} = R_a + D_a \times x_a \tag{43}$$

The algorithm starts by creating an individual's population. Moreover, this population is either random or even. Each individual's evaluation permits us to select that which gets changed to the sun.

Conventionally, M(%) plants are removed far away from the sun. As per the proposed CSDHAP model, the position in SFO is updated with DHO as given in equation (44).

$$R_{a+1} = R^{lead} - \tilde{T}.v.\left|\tilde{L} \times R^{lead} - R_a\right| + Levy \tag{44}$$

where R_a indicates the position of current iteration, R_{a+1} denotes the position at upcoming iteration, $\tilde{T}$ and $\tilde{L}$ indicates the coefficient vectors, R^{lead} indicates the leader position, and v denotes a random number along with the wind speed from 0 to 2.

As per the proposed CSDHAP model, the adaptive pollination rate is calculated as per equation (45).

$$\tilde{P}_r = 0.5 * \left(1 - \frac{iter}{iter_{\max}}\right) \tag{45}$$

As per the proposed CSDHAP model, the Cauchy mutation is performed. The Cauchy mutation is concerned to improve the capability of jumping out the local optima, as the Cauchy distribution is less with the higher search space. For creating the Cauchy random number, the Cauchy distribution function is given in equation (46).

$$\tilde{y} = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\zeta}{\tilde{g}}\right) \tag{46}$$

In equation (46), $\tilde{g}$ is similar to 1, $\tilde{y}$ indicates the uniformly distributed number within [0, 1], and $\zeta = \tan(\pi(\tilde{y} - 0.5))$.

The density function of Cauchy distribution is expressed in equation (47).

$$\tilde{f}_{Cauchy}\left(0, \tilde{g}\right) = \frac{1}{\pi}\frac{\tilde{g}}{\tilde{g}^2 + \tilde{y}^2} \tag{47}$$

The pseudo code of the adopted CSDHAP scheme.

Pseudo code of adopted CSDHAP model
Population initialisation of *NM* flowers
Identify the sun best solution s^*
Adjust all plants in the sun
while ($k < MaxDays$)
Compute the orientation vector
Proposed position of SFO is updated using DHO as per equation (44).
Compute the step for each plant
Greatest b plants pollinated in the sun

Assess the novel individuals
If a novel individual is a global greatest, then update it
end while
Best solution is obtained

4 Results and discussion

4.1 Simulation procedure

The proposed HC+CSDHAP was executed in Python programming language. The outcomes of the adopted HC+CSDHAP scheme were computed to the extant approaches such as Gomes et al. (2019), Brammya et al. (2019), Banchhor and Srinivasu (2019), Fouad (2015), Moosavi and Bardsiri (2019) and Selvi and Valarmathi (2020). The data was collected from WUSTL-IIOT-2018 Dataset (Shaik and Ganesh, 2020). The dataset was built using SCADA (Devarajan et al., 2022; Rabie et al., 2022; Khadidos et al., 2022a, 2022b; Shitharth et al., 2021; Alhalabi et al., 2023; Karthikeyan et al., 2023) test bed. The purpose of the test bed was to emulate industrial systems. They focus on reconnaissance attacks, where the network is scanned for possible vulnerabilities for later attacks. The performance was computed by changeable learning percentages 60, 70, 80, 90, for dissimilar performance metrics namely, FPR, sensitivity, specificity, accuracy, NPV, precision, FNR, F-measure, recall, and MCC, correspondingly.

4.2 Performance analysis

The performance analysis of the HC+CSDHAP is compared with HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, as illustrated in Figures 3, 4, and 5. Moreover, the accuracy of the adopted HC+CSDHAP scheme for learning percentage 70 is superior (~0.92) than other existing models like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, correspondingly as shown in Figure 3(b). This proves that the classification accuracy of adopted work is higher than the extant schemes. Likewise, the adopted HC+CSDHAP model attains higher sensitivity (~0.1) for learning percentage 60 in Bigdata classification than other existing schemes like HC+SFO (~0.9), HC+DHO (~0.5), HC+CGWO (~0.88), HC+SSO (~0.91), HC+PRO (~0.87), and HC+L-FF(~0.86), respectively in Figure 3(a). In addition, the proposed HC+CSDHAP model holds maximum specificity for learning percentage 70 when compared to the learning percentage 65 as given in Figure 3(d). Also, the precision on Bigdata classification model by proposed work is maximum when computed to existing scheme like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, respectively for learning percentage 80 in Figure 3(c). This performance on Bigdata classification model was better due to the appropriate training of HC with optimal factors for minimising the errors.

Figure 3 Comparative study of the HC+CSDHAP (see online version for colours)

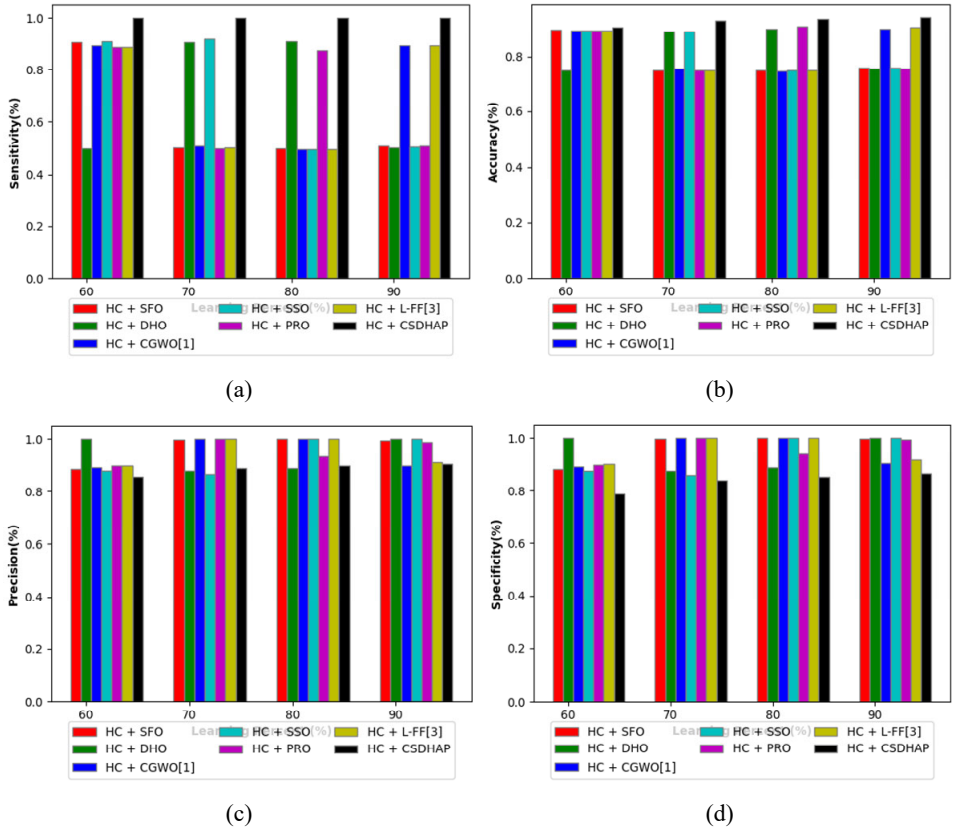


Figure 4 Comparative study of the HC+CSDHAP (see online version for colours)

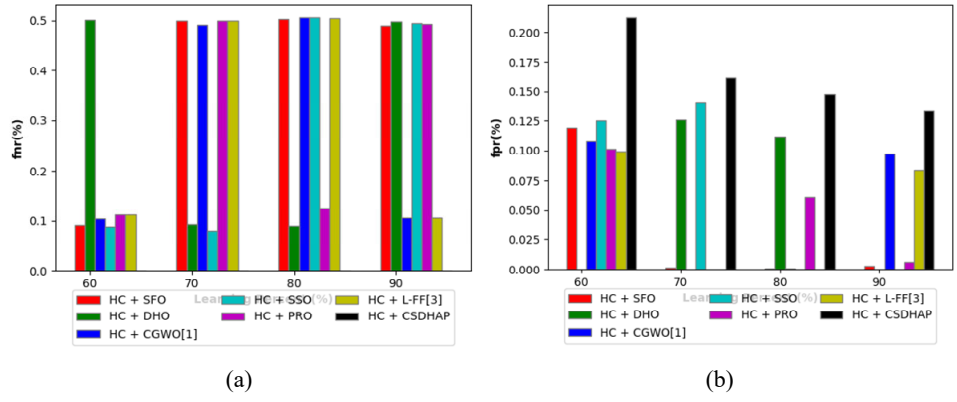


Figure 4 represents the negative metrics like FPR and FNR of the adopted HC+CSDHAP model over other conventional schemes. The adopted HC+CSDHAP model holds minimum FNR (~0.1) value for learning percentage 90 in Figure 4(a) than other traditional models like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and

HC+L-FF, respectively. Likewise, the FPR of the adopted HC+CSDHAP model attains lower value for learning percentage 90 with better performance when compared to the FPR value of other learning percentages in Figure 4(b). Thus, it is proved that the adopted HC+CSDHAP algorithm minimises the detection error that leads to accurate detection.

Figure 5 Performance analysis of the HC+CSDHAP (see online version for colours)

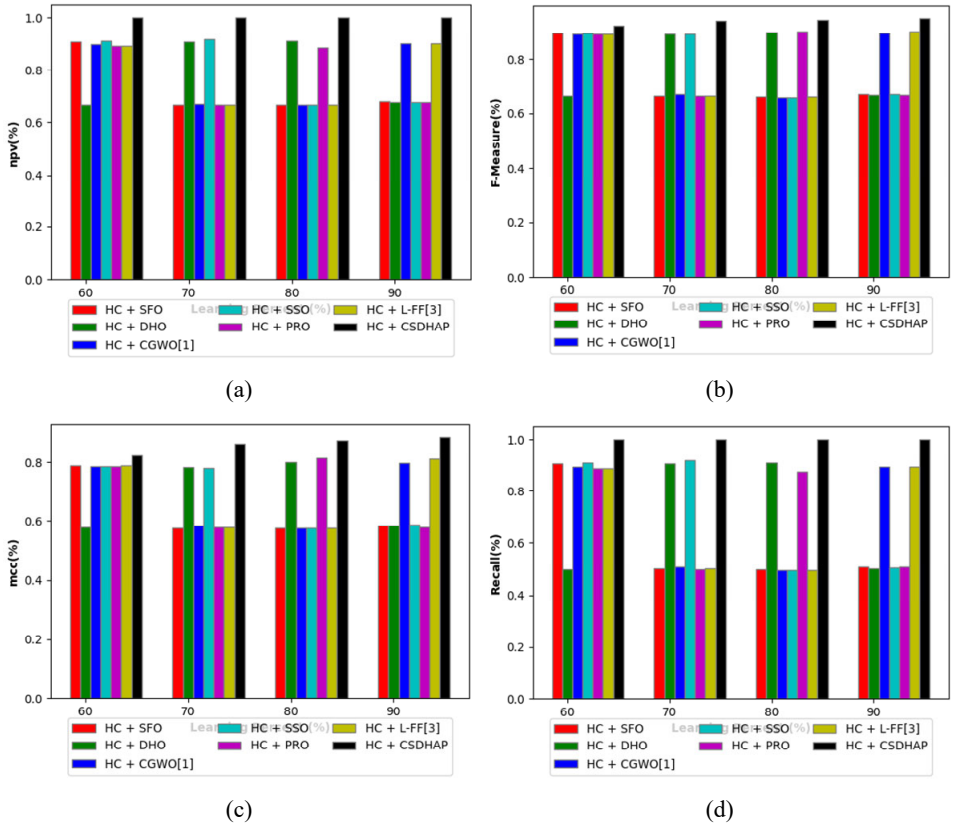


Figure 5 illustrates the other measures analysis. The F-measure of the proposed technique is superior to another traditional scheme like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, as shown in 5(b). Further, it is shown that the NPV of the proposed one holds a higher value (~1.0) for learning percentage 60; whereas, HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, respectively attains lower values as per Figure 5(a). Likewise, the adopted HC+CSDHAP model attains highest MCC (~0.98) for learning percentage 90 when compared to the learning percentage 60 in Figure 5(c). In addition, the Recall of adopted HC+CSDHAP model for learning percentage 80 in Figure 7(b) is better than another traditional scheme like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, correspondingly. Thus, the performance of presented HC+CSDHAP model has shown its enhancement over other existing schemes. This proves that the proposed model is more sufficient to handle the big data classification with the hybrid model.

Table 2 Analysis on classifier performance: proposed and traditional approaches

Metrics	*Classifiers*							
	RNN (Kao and Chiu, 2020)	*SVM (Avci, 2009)*	*CNN (LeCun et al., 2010; Shitharth and Winston D, 2017; Yadav et al., 2023)*	*Bi-LSTM (Zhou et al., 2019)*	*NB (Sunil Babu and Vijayalakshmi, 2019)*	*LSTM (Yan et al., 2020)*	*DBN (Wang et al., 2016)*	*Proposed HC+CSDHAP model*
Sensitivity (%)	0.700972	0.74053	0.685216	0.776735	0.779752	0.915186	0.998994	1
Accuracy (%)	0.8195	0.793	0.7945	0.8095	0.7925	0.898833	0.749167	0.929
Precision (%)	0.916301	0.825177	0.874251	0.829277	0.798216	0.885214	0.664882	0.887592
Specificity (%)	0.936692	0.844879	0.902552	0.841896	0.805104	0.882665	0.502154	0.838407
Recall (%)	0.700972	0.74053	0.685216	0.776735	0.779752	0.915186	0.998994	1
F-measure (%)	0.794302	0.780565	0.768277	0.802146	0.788876	0.899951	0.798392	0.940449
MCC (%)	0.656741	0.588822	0.602614	0.620084	0.585087	0.798151	0.57638	0.862649
NPV (%)	0.760086	0.767078	0.743583	0.792265	0.787103	0.913237	0.998024	1
FPR (%)	0.063308	0.155121	0.097448	0.158104	0.194896	0.117335	0.497846	0.161593
FNR (%)	0.299028	0.25947	0.314784	0.223265	0.220248	0.084814	0.001006	0

4.3 Analysis based on classifier performance

The performance analysis of the presented HC+CSDHAP model over other existing classifiers for different metrics is shown in Table 2. From the table, the adopted HC+CSDHAP model have proven its classification ability for all node setup than other conventional models such as RNN, SVM, CNN, Bi-LSTM, NB, LSTM, and DBN, respectively. Furthermore, the proposed HC+CSDHAP model attains maximum accuracy values (~0.929) when compared to the existing models. Also, the proposed HC+CSDHAP model attains better precision value than other traditional models like RNN, SVM, CNN, Bi-LSTM, NB, LSTM, and DBN, correspondingly. The adopted HC+CSDHAP model holds highest MCC value that is superior to other traditional models like RNN, SVM, CNN, Bi-LSTM, NB, LSTM, and DBN, respectively in Table 2. The outcomes have summarised that the proposed optimisation assisted hybrid model can classify the big data more precisely than the conventional classifiers with the involvement of map-reduce architecture.

4.4 Statistical analysis

The comparative study is presented in Table 3. The results of the proposed HC+CSDHAP method is compared with the existing models. The mean performance of the adopted HC+CSDHAP approach holds better results than SFO, DHO, CGWO, SSO, PRO, and L-FF. The proposed HC+CSDHAP model attains ~0.0588 in best case. The proposed HC+CSDHAP model has proved its improvement by precisely classifying the big data almost in all cases. This also proves the performance of proposed optimisation strategy on identifying the optimal solutions for better classification without stuck onto the local optima.

Table 3 Comparative study with respect to accuracy

Metrics	*Best*	*Worst*	*Mean*	*Median*	*SD*
HC+SFO (Gomes et al. 2019)	0.105125	0.24925	0.211094	0.245	0.061271
HC+DHO (Brammya et al., 2019)	0.10025	0.249125	0.17551	0.176333	0.0709
HC+CGWO (Banchhor and Srinivasu, 2019)	0.101	0.25075	0.175688	0.1755	0.071998
HC+SSO (Fouad, 2015)	0.106875	0.2505	0.177344	0.176	0.068742
HC+PRO (Moosavi and Bardsiri, 2019)	0.093	0.2485	0.172938	0.175125	0.073245
HC+L-FF (Selvi and Valarmathi, 2020)	0.0945	0.25	0.174542	0.176833	0.074646
Proposed HC+ CSDHAP model	0.058875	0.093375	0.072031	0.067937	0.013047

4.5 Analysis on feature

The analysis of adopted work based on feature is represented in Table 4. In addition, the proposed HC+CSDHAP model with an improved entropy features hold superior precision (~1) than other traditional models like proposed HC+CSDHAP model, and Proposed model without optimisation, respectively. Further, the Proposed HC+CSDHAP model with an improved entropy features has shown lower FPR with better performance when compared to other existing models such as proposed HC+CSDHAP model, and

proposed model without optimisation, respectively. This has attained that the proposed HC+CSDHAP model with an improved entropy features helps to classify the big data models more accurately (https://www.cse.wustl.edu/~jain/iiot/index.html, https://sites.google.com/view/iot-network-intrusion-dataset/home).

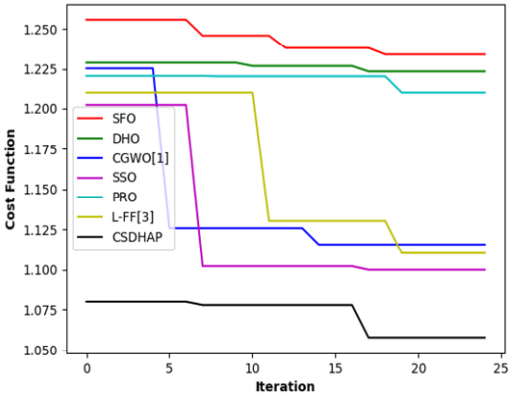
Table 4 Analysis of proposed work with different feature selection combinations using confusion matrix

Metrics	*Proposed HC+ CSDHAP model*	*Proposed model without optimisation*	*Proposed HC+ CSDHAP model with an improved entropy features*
MCC (%)	0.862649	0.819666	0.578051
Specificity (%)	0.838407	0.925091	1
FPR (%)	0.161593	0.074909	0
F-measure (%)	0.940449	0.90776	0.66622
Recall (%)	1	0.894066	0.499497
Precision (%)	0.887592	0.92188	1
Accuracy (%)	0.929	0.909667	0.751167
NPV (%)	1	0.898294	0.668958
Sensitivity (%)	1	0.894066	0.499497
FNR (%)	0	0.105934	0.500503
TP	1.789	1.89	1.21
TN	3.12	3.34	2.81
FP	1.3	0.91	0.30
FN	0	1	0.46

4.6 Convergence analysis

The CSDHAP model is compared with existing ones by varying the iteration counts. Figure 6 presents the analysis. The CSDHAP method achieves the minimum cost function as given in equation (38). The CSDHAP has achieved the lower cost function with grander results.

Figure 6 Convergence analysis (see online version for colours)



5 Conclusions

This paper has presented a big data classification model. Originally, the pre-processing process was done. Then, the data were then subjected under MapReduce framework. Subsequently, the feature extraction was performed, in which statistical functions; an improved entropy, higher order statistical features, and gain ratio were taken out. Furthermore, the data classification wad performed by employing classifiers. For enhancing the performance of classification results, the weight of both DBN (https://www.cse.wustl.edu/~jain/iiot/index.html, https://sites.google.com/view/iot-network-intrusion-dataset/home) and LSTM (Shitharth and Winston D, 2017) were optimised via a proposed CSDHAP model that hybridised the SFO and DHO. The proposed approach evaluated using various metrics namely, F1-score, specificity, NPV, accuracy, FNR, FDR, sensitivity, precision, FPR, and MCC, respectively. From the graph, the adopted HC+CSDHAP model attained higher sensitivity (~0.1) for learning percentage 65 in big data classification than other existing schemes like HC+SFO (~0.9), HC+DHO (~0.5), HC+CGWO (~0.88), HC+SSO (~0.91), HC+PRO (~0.87), and HC+L-FF(~0.86), respectively. In addition, the Recall of adopted HC+CSDHAP model for learning percentage 85 was better than other traditional scheme like HC+SFO, HC+DHO, HC+CGWO, HC+SSO, HC+PRO, and HC+L-FF, correspondingly. Also, the proposed HC+CSDHAP model attained better precision value than other traditional models like RNN, SVM, CNN, Bi-LSTM, NB, LSTM, and DBN, correspondingly.

References

Abdel-Hamid, N.B., ElGhamrawy, S., Desouky, A.E. et al. (2018) 'A dynamic spark-based classification framework for imbalanced big data', *J. Grid Computing*, Vol. 16, pp.607–626, https://doi.org/10.1007/s10723-018-9465-z.

Alhalabi, W., Al-Rasheed, A., Manoharan, H., Alabdulkareem, E., Alduailij, M., Alduailij, M. and Selvarajan, S. (2023) 'Distinctive measurement scheme for security and privacy in internet of things applications using machine learning algorithms', *Electronics*, Vol. 12, p.747, https://doi.org/10.3390/electronics12030747.

Avci, E. (2009) 'A new intelligent diagnosis system for the heart valve diseases by using genetic-SVM classifier', *Expert Systems with Applications*, September, Vol. 36, No. 7, pp.10618–10626.

Baldán, F.J. and Benítez, J.M. (2018) 'Distributed fastshapelet transform: a big data time series classification algorithm', *Information Sciences*, 19 October, Vol. 496, pp.451–463 (Cover date: September 2019).

Banchhor, C. and Srinivasu, N. (2019) 'Integrating cuckoo search-grey wolf optimization and correlative naive Bayes classifier with map reduce model for big data classification', *Data & Knowledge Engineering*, 27 December, Vol. 127, Article 101788 (Cover date: May 2020).

Beno, M.M., Valarmathi, I.R., Swamy, S.M. and Rajakumar, B.R. (2014) 'Threshold prediction for segmenting tumour from brain MRI scans', *International Journal of Imaging Systems and Technology*, Vol. 24, No. 2, pp.129–137, DOI: https://doi.org/10.1002/ima.22087.

Bovenzi, G., Aceto, G., Ciuonzo, D., Persico, V. and Pescapé, A. (2020) 'A big data-enabled hierarchical framework for traffic classification', *IEEE Transactions on Network Science and Engineering*, 1 October–December, Vol. 7, No. 4, pp.2608–2619, DOI: 10.1109/TNSE.2020.3009832.

Brammya, G., Praveena, S., Preetha, N.S.N., Ramya, R., Rajakumar, B.R. and Binu, D. (2019) 'Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm', *The Computer Journal*, https://doi.org/10.1093/comjnl/bxy133.

Cai, Z., Wang, J. and Ma, M. (2021) 'The performance evaluation of big data-driven modulation classification in complex environment', *IEEE Access*, Vol. 9, pp.26313–26322, DOI: 10.1109/ACCESS.2021.3054756.

Dagdia, Z.C. (2018) 'A scalable and distributed dendritic cell algorithm for big data classification', *Swarm and Evolutionary Computation*, 1 September, Vol. 50, Article 100432 (Cover date: November 2019).

Dessì, D., Fenu, G., Marras, M. and Recupero, D.R. (2018) 'Bridging learning analytics and cognitive computing for big data classification in micro-learning video collections', *Computers in Human Behavior*, 3 March, Vol. 92, pp.468–477 (Cover date: March 2019).

Devarajan, D. et al. (2022) 'Cervical cancer diagnosis using intelligent living behavior of artificial jellyfish optimized with artificial neural network', in *IEEE Access*, Vol. 10, pp.126957–126968, DOI: 10.1109/ACCESS.2022.3221451.

Ducange, P., Fazzolari, M. and Marcelloni, F. (2020) 'An overview of recent distributed algorithms for learning fuzzy models in big data classification', *J. Big Data*, Vol. 7, p.19, https://doi.org/10.1186/s40537-020-00298-6.

Elkano, M., Sanz, J.A., Barrenechea, E., Bustince, H. and Galar, M. (2020) 'CFM-BD: a distributed rule induction algorithm for building compact fuzzy models in big data classification problems', in *IEEE Transactions on Fuzzy Systems*, January, Vol. 28, No. 1, pp.163–177, DOI: 10.1109/TFUZZ.2019.2900856.

Fan, Y., Bai, J. and Tan, G. (2020) 'Privacy preserving based logistic regression on big data', *Journal of Network and Computer Applications*, 3 August, Vol. 171, Article 102769, (Cover date: 1 December 2020).

Fouad, A. (2015) *Social Spider Optimization Algorithm*, DOI: 10.13140/RG.2.1.4314.5361.

García-Gil, D., Luengo, J. and Herrera, F. (2018) 'Enabling smart data: noise filtering in big data classification', *Information Sciences*, 3 December, Vol. 479, pp.135–152 (Cover date: April 2019).

Gomes, G.F., da Cunha, S.S. and Ancelotti, A.C. (2019) 'A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates', *Engineering with Computers*, Vol. 35, pp.619–626, https://doi.org/10.1007/s00366-018-0620-8.

González, A., Pérez, R. and Romero-Zaliz, R. (2019) 'An incremental approach to address big data classification problems using cognitive models', *Cogn. Comput*., Vol. 11, pp.347–366, https://doi.org/10.1007/s12559-019-09655-x.

Hababeh, I., Gharaibeh, A., Nofal, S. and Khalil, I. (2019) 'An integrated methodology for big data classification and security for improving cloud systems data mobility', *IEEE Access*, Vol. 7, pp.9153–9163, DOI: 10.1109/ACCESS.2018.2890099.

Hernández, G., Zamora, E. and Furlán, F. (2019) 'Hybrid neural networks for big data classification', *Neurocomputing*, 21 October, Vol. 390, pp.327–340 (Cover date: 21 May 2020).

Hojage, A. (2021) 'Race detection using mutated SALP swarm optimization algorithm based DBN from face shape features', *Multimedia Research*, Vol. 4, No. 2, pp.147–162.

Iyer, S.S., Jain, A. and Wang, J. (2022) *Handbook of Research on Lifestyle Sustainability and Management Solutions Using AI, Big Data Analytics, and Visualization*, February, p.411 [online] https://books.google.co.in/books?id=r59lEAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false (accessed 24 May 2023).

Jiang, C. and Li, Y. (2019) 'Health big data classification using improved radial basis function neural network and nearest neighbor propagation algorithm', *IEEE Access*, Vol. 7, pp.176782–176789, DOI: 10.1109/ACCESS.2019.2956751.

Kanani, S., Patel, S., Gupta, R.K., Jain, A. and Lin, J.C-W. (2023) 'An AI-enabled ensemble method for rainfall forecasting using long-short term memory', *Mathematical Biosciences and Engineering*, Vol. 20, No. 5, pp.8975–9002.

Kao, L-J. and Chiu, C.C. (2020) 'Application of integrated recurrent neural network with multivariate adaptive regression splines on SPC-EPC process', *Journal of Manufacturing Systems*, Vol. 57, pp.109–118.

Karthikeyan, R., Sundaravadivazhagan, B., Cyriac, R., Balachandran, P.K. and Shitharth, S. (2023) 'Preserving resource handiness and exigency-based migration algorithm (PRH-EM) for energy efficient federated cloud management systems', *Mobile Information Systems*, Vol. 2023, Article ID 7754765, 11 pages, 2023. https://doi.org/10.1155/2023/7754765

Kaur, P., Jain, S., Jain, A. and Morato, J. (2022) 'Fuzzy based model for students debar policy in Indian Engineering Institutes', *The European Conference on Education 2022: Official Conference Proceedings*, ISSN: 2188-1162, https://doi.org/10.22492/issn.2188-1162.2022.33.

Khadidos, A.O., Khadidos, A.O., Manoharan, H., Alyoubi, K.H., Alshareef, A.M. and Selvarajan, S. (2022a) 'Integrating industrial appliances for security enhancement in data point using SCADA networks with learning algorithm', *International Transactions on Electrical Energy Systems*, Vol. 2022, Article ID 8685235, 11pp, https://doi.org/10.1155/2022/8685235.

Khadidos, A.O., Manoharan, H., Selvarajan, S., Khadidos, A.O., Alyoubi, K.H. and Yafoz, A. (2022b) 'A classy multifacet clustering and fused optimization based classification methodologies for SCADA security', *Energies*, Vol. 15, p.3624, https://doi.org/10.3390/en15103624.

Khalemsky, A. and Gelbard, R. (2019) 'A dynamic classification unit for online segmentation of big data via small data buffers', *Decision Support Systems*, 7 September, Vol. 128, Article 113157, (Cover date: January 2020).

LeCun, Y., Kavukvuoglu, K. and Farabet, C. (2010) 'Convolutional networks and applications in vision', in *International Symposium on Circuits and Systems*, pp.253–256.

Ma, Z., Yang, L.T. and Zhang, Q. (2021) 'Support multimode tensor machine for multiple classification on industrial big data', *IEEE Transactions on Industrial Informatics*, May, Vol. 17, No. 5, pp.3382–3390, DOI: 10.1109/TII.2020.2999622.

Moosavi, S.H.S. and Bardsiri, V.K. (2019) 'Poor and rich optimization algorithm: a new human-based and multi populations algorithm', *Engineering Applications of Artificial Intelligence*, 26 September, Vol. 86, pp.165–181, (Cover date: November 2019).

Rabie, O.B.J., Balachandran, P.K., Khojah, M. and Selvarajan, S. (2022) 'A proficient ZESO-DRKFC model for smart grid SCADA security', *Electronics*, Vol. 11, p.4144, https://doi.org/10.3390/electronics11244144.

Ravuri, V. and Vasundra, S. (2020) 'Moth-flame optimization-bat optimization: map-reduce framework for big data clustering using the moth-flame bat optimization and sparse fuzzy C-means', *Big Data*, Vol. 8, No. 3, DOI: 10.1089/big.2019.0125.

Reddy, A.M., Reddy, K.S., Jayaram, M., Lakshmi, N.V.M., Aluvalu, R., Mahesh, T.R., Kumar, V.V. and Alex, D.S. (2022) 'An efficient multilevel thresholding scheme for heart image segmentation using a hybrid generalized adversarial network', *Journal of Sensors*, Vol. 2022, Article ID 4093658, 11pp, https://doi.org/10.1155/2022/4093658.

Reddy, M.A., Reddy, S.K., Kumar, S.C.N. and. Reddy, S.K (2022) 'Leveraging bio-maximum inverse rank method for iris and palm recognition', *International Journal of Biometrics (IJBM)*, Published Online: 11 July, Vol. 14, Nos. 3/4, pp.421–438, https://doi.org/10.1504/IJBM.2022.124681.

Saki, M., Abolhasan, M. and Lipman, J. (2020) 'A novel approach for big data classification and transportation in rail networks', *IEEE Transactions on Intelligent Transportation Systems*, March, Vol. 21, No. 3, pp.1239–1249, DOI: 10.1109/TITS.2019.2905611.

Selvi, R.S. and Valarmathi, M.L. (2020) 'Optimal feature selection for big data classification: firefly with lion-assisted model', *Big Data*, Vol. 8, No. 2, DOI: 10.1089/big.2019.0022.

Shaik, J.B. and Ganesh, V. (2020) 'Deep neural network and social ski-driver optimization algorithm for power system restoration with VSC-HVDC technology', *Journal of Computational Mechanics, Power System and Control*, Vol. 3, No. 1, pp.1–9.

Shitharth, S. and Winston D, P. (2017) 'Comparison of PRC based RVM classification versus SVM classification in SCADA network', *Journal of Electrical Engineering*, Vol. 17, No. 1, pp.318–331.

Shitharth, S., Meshram, P., Kshirsagar, P.R., Manoharan, H., Tirth, V. and Sundramurthy, V.P. (2021) 'Impact of big data analysis on nanosensors for applied sciences using neural networks', *Journal of Nanomaterials*, Vol. 2021, Article ID 4927607, 9pp, https://doi.org/10.1155/2021/4927607.

Sunil Babu, M. and Vijayalakshmi, V. (2019) 'An effective approach for sub-acute ischemic stroke lesion segmentation by adopting meta-heuristics feature selection technique along with hybrid naive Bayes and sample-weighted random forest classification', *Sens. Imaging*, Vol. 20, p.7, https://doi.org/10.1007/s11220-019-0230-6.

Thangam, T., Muthuvel, K. and Kazem, H.A. (2019) 'SFOA: sun flower optimization algorithm to solve optimal power flow', *Journal of Computational Mechanics, Power System and Control*, Vol. 2, No. 4, pp.10–18.

Thomas, R. and Rangachar, M.J.S. (2018) 'Hybrid optimization based DBN for face recognition using low-resolution images', *Multimedia Research*, Vol. 1, No. 1, pp.33–43.

Venkatesh, R., Balasubramanian, C. and Kaliappan, M. (2019) 'Development of big data predictive analytics model for disease prediction using machine learning technique', *J. Med. Syst*., Vol. 43, p.272, https://doi.org/10.1007/s10916-019-1398-y.

Visuwasam, L.M.M. and Raj, D.P. (2020) 'A distributed intelligent mobile application for analyzing travel big data analytics', *Peer-to-Peer Netw. Appl*., Vol. 13, pp.2036–2052, https://doi.org/10.1007/s12083-019-00799-z.

Wang, H.Z., Wang, G.B., Li, G.Q., Peng, J.C. and Liu, Y.T. (2016) 'Deep belief network based deterministic and probabilistic wind speed forecasting approach', *Applied Energy*, Vol. 182, pp.80–93.

Xing, W. and Bei, Y. (2020) 'Medical health big data classification based on KNN classification algorithm', *IEEE Access*, Vol. 8, pp.28808–28819, DOI: 10.1109/ACCESS.2019.2955754.

Yadav, D., Gupta, A., Jain, A. and Yadav, A.K. (2023) 'Plant leaf disease detection using CNN with transfer learning and XGBoost', *International Journal of Data Analysis Techniques and Strategies*, 9 January, Vol. 14, No. 3, pp.244–265, https://doi.org/10.1504/IJDATS.2022.128273.

Yan, H., Qin, Y. and Chen, H. (2020) 'Long-term gear life prediction based on ordered neurons LSTM neural networks', *Measurement*, 11 July, Vol. 165, Article 108205 (Cover date: 1 December 2020).

Yang, L-H., Liu, J., Wang, Y-M. and Martínez, L. (2021) 'A micro-extended belief rule-based system for big data multiclass classification problems', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, January, Vol. 51, No. 1, pp.420–440, DOI: 10.1109/TSMC.2018.2872843.

Zhai, J., Zhang, S., Zhang, M. et al. (2018) 'Fuzzy integral-based ELM ensemble for imbalanced big data classification', *Soft Comput*., Vol. 22, pp.3519–3531, https://doi.org/10.1007/s00500-018-3085-1.

Zhou, X., Lin, J., Zhang, Z., Shao, Z. and Liu, H. (2019) 'Improved itracker combined with bidirectional long short-term memory for 3D gaze estimation using appearance cues', *Neuro Computing*, in press, corrected proof, 20 October.

Nomenclature

Abbreviation	*Description*
DBN	Deep belief network
FPR	False positive rate
PSO	Particle swarm optimisation
ML	Machine learning
SFO	Sun flower optimisation
DCA	Dendritic cell algorithm
FNR	False negative rate
L-FF	Lion-based firefly
DL	Deep learning
NPV	Net predictive value
DHO	Deer hunting optimisation
LSTM	Long short-term memory
MCC	Matthews correlation coefficient
CGCNBMRM	CG-CNB and MapReduce model
MFO-Bat	Moth-flame optimisation-based bat
GWO	Grey Wolf optimiser
CS	Cuckoo search
CGWO	Cuckoo-Grey wolf based optimisation
SSO	Social spider optimisation
FCM	Fuzzy C-means
CSDHAP	Combined sunflower and deer hunting model with adaptive pollination rate
CG-CNB	Cuckoo-Grey wolf based correlative naive Bayes classifier
PRO	Poor and rich optimisation