Pertanika J. Sci. & Technol. 32 (5): 2033 - 2053 (2024)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Learning Discriminative Features Using ANN-based Progressive Learning Model for Efficient Big Data Classification

Nandita Bangera[1,2]* and Kayarvizhy[1]

[1]*B.M.S College of Engineering, Bangalore, Karnataka 560019, India*
[2]*RV Institute of Technology and Management, Bangalore 560076, India*

ABSTRACT

Progressive techniques encompass iterative and adaptive approaches that incrementally enhance and optimize data processing by iteratively modifying the analysis process, resulting in improved efficiency and precision of outcomes. These techniques contain a range of components, such as data sampling, feature selection, and learning algorithms. This study proposes the integration of an Artificial Neural Network (ANN) with a Progressive Learning Model (PLM) to enhance the efficacy of learning from large-scale datasets. The SMOTE and Pearson Correlation Coefficient (PCC) methods are commonly employed in imbalanced dataset handling and feature selection. The utilization of progressive weight updating is a notable strategy for improving performance optimization in neural network models. This approach involves the incremental modification of the network's progressive weights during the training phase rather than relying on gradient values. The proposed method gradually obtains the localization of discriminative data by incorporating information from local details into the overall global structure, effectively reducing the training time by iteratively updating the weights. The model has been examined using two distinct datasets: the Poker hand and the Higgs. The performance of the suggested method is compared with that of classification algorithms: Population and Global Search Improved Squirrel Search Algorithm (PGS-ISSA) and Adaptive E-Bat (AEB). The convergence of Poker's is achieved after 50 epochs with ANN-PLM; however, without PLM, it takes 65 epochs. Similarly, with the Higgs, convergence is achieved after 25 epochs with PLM and 40 without PLM.

Keywords: Artificial neural network, big data classification, data imbalance, Pearson correlation coefficient-based feature selection, progressive learning model, weight updating

ARTICLE INFO

Article history:
Received: 22 August 2023
Accepted: 01 February 2024
Published: 08 August 2024

DOI: https://doi.org/10.47836/pjst.32.5.06

E-mail addresses:
nanditamanohar@gmail.com (Nandita Bangera)
kayarvizhyn.cse@bmsce.ac.in (Kayarvizhy)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

Information technologies have achieved extraordinary growth in data. Large amounts of data from various applications are combined as big data, which has resulted in the complexity of dealing with big data (Wang et al., 2021) and enhancing convergence. Big data is either structured or unstructured. The number of data created is represented as volume (Dubey et al., 2021), data's creation speed is defined as velocity and structured and unstructured characteristics are represented as data's variety (Jain et al., 2022; Kantapalli & Markapudi, 2023). Big data gathers huge attention in numerous areas, such as electronic commerce, online social networks, the Internet of Things, bioinformatics, and e-health because those applications have progressively achieved an enormous amount of raw data (Brahmane & Krishna, 2021; Hassib et al., 2020; Park et al., 2021; Xing & Bei, 2020).

Big data applications have revolutionized various industries by providing unprecedented opportunities to extract valuable insights from massive and complex datasets. However, the volume and complexity of big data often pose challenges in terms of response time, as processing such large-scale data can be time-consuming and resource-intensive. Many techniques have emerged as promising approaches to reduce response time and improve the efficiency of big data applications in addressing these challenges,

Data preparation techniques can reduce processing time in large-scale data applications. Preprocessing refers to a set of operations that increase the quality and usability of data, such as data cleansing, transformation, and integration. Preprocessing processes that are executed efficiently can decrease unnecessary computational overhead, resulting in faster processing time. Sampling approaches are an alternate method for dealing with the issue of time limits. Sampling is a statistical strategy in which a representative subset of data is chosen for examination rather than the complete dataset. It is feasible to achieve large savings in computing complexity and processing time while receiving important insights using a smaller sample size. It is also critical to ensure that the sampling approach maintains the statistical traits and characteristics of the initial dataset. In machine learning, data classification is an extensive operation that involves understanding the targeted data to predict the class of unseen data (Banchhor & Srinivasu, 2021). The existence of prominent redundancy of information in data is required to be noted while examining the openly accessible tabular big data issues because these redundant features cause an impact on storage and scalability (Basgall et al., 2020). The training of an efficient learning system is difficult in data mining when the given class distribution is imbalanced in a training data set. Moreover, the classification of rare objects is more complex than that of general objects in most data mining approaches (Abhilasha & Naidu, 2022). The imbalance (Juez-Gil et al., 2021) decreases the classifier's generalization abilities and makes it inefficient for minority classes (Sleeman & Krawczyk, 2021). Therefore, feature selection is combined with progressive learning to improve big data classification in this work. Feature selection

is choosing appropriate features and eliminating redundant features from the dataset. Moreover, preserving the strong features makes the predictive model highly discriminative, which enhances performance (Al-Thanoon et al., 2021; BenSaid & Alimi, 2021).

The timely completion of data processing and machine learning model training is critical for generating efficient and timely results. Numerous approaches have been proposed in scholarly publications, including feature selection, dimensionality reduction, ensemble learning, approximation, transfer learning, progressive sampling algorithms, mini-batch learning, and online learning to overcome this barrier. Using these strategies, researchers can effectively reduce training time while maintaining acceptable levels of precision, allowing for faster and more effective application of machine learning models.

The use of iterative and incremental strategies in model construction and data analysis distinguishes progressive approaches in machine learning. These strategies aim to gradually improve the accuracy of machine learning models by incorporating new data and modifying model parameters. Researchers and data scientists can iteratively improve the precision and efficacy of their models by implementing progressive methodologies. Progressive techniques have been developed to decrease the temporal complexity of training time, resulting in faster learning. This advancement is particularly significant for big data applications. However, it is important to note that these techniques also have certain drawbacks. Progressive networks present a model framework that enables transfer through lateral connections to characteristics of previously acquired columns. This mechanism mitigates the issue of catastrophic forgetting by establishing a distinct neural network, referred to as a column, for each task being performed. During training, the system maintains a reservoir of pre-trained models and leverages lateral connections from existing models to extract valuable characteristics for novel tasks. The network's last layer, along with its associated weights, increases in size as each new class is introduced. All these models necessitate additional overhead in establishing new connections and incorporating additional columns to retain the acquired data. The trade-off involves an increase in model complexity, which refers to including a greater number of parameters to be trained for each extra column. If a new class is introduced, it becomes necessary to retrain the entire model. The implementation of progressive learning paradigms necessitates a fundamental alteration in the arrangement of layers and neurons, augmenting the process's intricacy.

Progressive learning is a concept in artificial neural networks (ANNs) that refers to incrementally improving a neural network's performance over time. There are many progressive learning techniques for ANN and CNN. The techniques differ in the way the network carries out the learning. Recent literature has focused on developing progressive learning algorithms that are more efficient, robust, and flexible. Some approaches include incremental, transfer, lifelong, and meta-learning. Incremental learning methods gradually learn new tasks while preserving previously learned knowledge. Transfer learning

approaches leverage knowledge from previously learned tasks to improve learning on new tasks. Lifelong learning methods learn continuously over an extended period while maintaining a growing knowledge base. Meta-learning methods aim to learn how to learn, facilitating faster and more efficient learning.

This research uses the ANN with PLM to perform big data classification without changing the overall structure of the ANN and maintaining the process of traditional ANN. In this proposed system, the incremental learning of weights according to the batches of data is termed a progressive learning model. This research highlights feature selection, data imbalance, and progressive learning methods, which collectively reduce the training time of neural networks.

The contributions of this work are concise as follows—the ANN-PLM approach is used to localize discriminative data from local details to the global structure, which is further used to enhance classification by combining the output of the last multiple stages and progressively updating the probabilities of the weight. PLM helps the neural network learn from the data effectively and efficiently, leading to better accuracy and faster training times.

The possible research questions that arise and which have been addressed in the paper are:

- RQ1: How does combining a Progressive Learning Model (PLM) with an Artificial Neural Network (ANN) impact learning effectiveness from extensive datasets?
- RQ2: What is the specific impact of progressive weight updating on reducing the training time of the ANN-PLM model, and how does this compare to traditional gradient-based weight updating methods?
- RQ3: What is the comparative performance of the ANN-PLM model about traditional classification algorithms like Population and Global Search improved Squirrel search Algorithm (PGS-ISSA) and Adaptive E-Bat (AEB)?

RELATED WORK

We discuss further the related work on big data classification and progressive learning, along with its advantages and limitations.

Du et al. (2022) developed a progressive training approach that operated in a zooming-out manner to perform fine-grained visual classification. This progressive training was executed in various steps to accomplish the feature learning and acquire the essential complementary characteristics between various granularities. The Category-Consistent Block Convolution (CCBC) was proposed, which integrated the operation of block convolution with the feature Category-Consistency Constraint (CCC). This CCC was used to overcome the overfitting issue and confirmed that the acquired multi-granularity regions are expressive and related to classes. The classification of developed progressive training mainly depends on the block numbers of convolutional layers.

Rebuffi et al. (2017) proposed a method for incremental learning that addresses the problem of catastrophic forgetting. It uses a combination of exemplar-based rehearsal and feature expansion to learn new tasks while preserving old ones. All these methods have the overhead of storing the old, learned data.

A newly developed learning method that could help learn new classes while keeping information from older courses was proposed by Venkatesan and Er (2016). The number of classes did not bind it. The neural network structure is automatically reconstructed by enabling new neurons and interconnections when a new class that is not native to the knowledge obtained so far is encountered, and the parameters are computed so that the knowledge learned thus far is kept. This approach is suited for real-world applications where it is necessary to learn online using real-time data and where the number of classes is frequently uncertain. The consistency and intricacy of the progressive learning method are studied. The proposed method used the ELM technique, where the output layer structure changes every time a new class is introduced.

Chatterjee et al. (2017) created an approach for systematically creating a large artificial neural network employing a progression property in this paper. The systematic design handles network size selection and parameter regularization. A network's number of nodes and layers grows over time to constantly lower a reasonable cost. Each layer is optimized individually, with optimal parameters learned via convex optimization. Certain weight matrices' random occurrences reduced the number of parameters to learn. However, instead of utilizing a back propagation-based learning strategy, they applied a nonlinear modification at each layer.

In this Progressive method, a deep network is developed unsupervised by PSL, a progressive stage-wise learning framework for unsupervised visual representation learning (Li et al., 2021). Early learning stages concentrate on simple tasks, whereas later learning stages are guided to glean deeper knowledge from more challenging tasks. They have used the gradient flow concept from one step to the next.

The suggested network architecture prevents prior knowledge from being forgotten and allows previously learned knowledge to be leveraged through lateral connections to previously learned classes and their attributes (Siddiqui & Park, 2021). Furthermore, the suggested technique is scalable and does not necessitate structural changes to the network trained on the old task; both are critical qualities in embedded systems, but this proposed method requires a pool of pre-trained models. Progressive Neural Networks (ProgNN) is a method for incremental learning (Rusu et al., 2016). ProgNN adds new neural networks to the architecture to solve new tasks while retaining knowledge from previous tasks. Each new network is trained on the new task and connected to the previous networks, forming a chain of expertise.

This work is based on a cross-entropy loss to learn the new classes and a distillation measure to retain the knowledge from the old classes (Castro et al., 2018). It requires extra memory space to store the old class data.

Movassagh et al. (2021) suggest a Hierarchical Convolutional Neural Network (HCNN) for image classification in this paper, which consists of numerous subnetworks utilized to categorize images progressively. The images with the revised weights are utilized to train the following sub-networks. If the prediction confidences in a sub-network are above a certain threshold, the results are output immediately. Otherwise, the following sub-networks must acquire deeper visual properties sequentially until a reliable image classification result or the last sub-network is reached. Otherwise, the following sub-networks must acquire deeper visual properties one after the other until a reliable image classification result or the last sub-network is reached. The model's accuracy is relatively high; however, it necessitates the maintenance of subnetworks, which adds overhead. All the research on progressive learning depicts the stupendous effort to retain knowledge, creating an overhead in time and memory.

The remaining portion of the associated study is based on data imbalance and classification techniques used for ANN. Smote plays an important role in solving the issues related to data imbalance. Sleeman and Krawczyk (2021) presented Apache Spark, including a SMOTE, to overcome spatial restrictions in big data analytics. The developed multi-class sampling approaches, i.e., SMOTE, under- and oversampling, were augmented with informative sampling and partitioning for SMOTE in Spark nodes. Therefore, clustering-based data partitioning was used to avoid the issue of the absence of spatial coherence between the instances from each class because of the random data splitting between the nodes. The probability of generating erroneous artificial instances was minimized by using the SMOTE in Spark. For an effective classification, feature selection was required to be considered for selecting the optimal features.

Ali and Balakrishnan (2021) developed the Population and Global Search Improved Squirrel Search Algorithm (PGS-ISSA) for feature selection. The developed PGS-ISSA was used to overcome the issue of local optimum and minimize the convergence rate in the conventional squirrel search algorithm. The main modification of this PGS-ISSA was the development of chaos theory to improve the population initialization, which is used to maximize the search space. The optimal features were chosen according to the minimum error rate used in the fitness to enhance the classification. The classification's SVM does not perform better when processed with larger datasets.

Mujeeb et al. (2021) presented the optimization-based MapReduce framework (MRF) for dealing with imbalanced data using the deep learning approach in classification. An adaptive E-Bat (AEB) approach was used to select the feature using the mappers in the MRF. The developed AEB integrated the Exponential Weighted Moving Average (EWMA) and the Bat algorithm (BA). The AEB was used to modify the update expression of E-Bat by creating an adaptive one for handling real-time data. The Deep Belief Network was used to classify the features according to the chosen ones. The developed AEB required

many training data to provide better classification accuracy. In most of the work mentioned above, the categorization of the generated progressive training was mostly based on the convolutional layer block counts, and a large amount of training data was needed.

Zhou et al. (2021) presented the region proposal and progressive learning, namely PRP-Net, for recognizing vegetable disease under a complex background. The attention proposal subnetwork, APN, was developed to acquire the disease's key regions from the background. The developed APN provided highly discriminative data for extracting the features. Next, these acquired regions were integrated with progressive learning to support the model in concentrating on fine-grained areas to obtain multiscale features. The channel attention mechanism was used to estimate the features for classification. The database's knowledge information was required to be enhanced to assist the training process; only then can it process the data from various times and planes. Hassanat et al. (2021) developed a supervised machine learning Magnetic Force (MF) classifier for big data classification according to iron-filling attraction to magnetic force. Here, the class was denoted by certain magnets, and iron filings denoted the unknown data points required to be categorized in big data classification. The inverse square law was applied to computing each class's force over each point in feature space. The developed MF was sensitive to the information skewed by the class.

MATERIALS AND METHODS

With a progressive learning model, ANN is developed to improve big data classification in this research. The important processes of the proposed method are (1) data acquisition, (2) class imbalance processing using SMOTE, (3) feature selection using PCC, and (4) classification using ANN-PLM. Here, the PCC is used to choose the optimal features from the feature vector, which leads to improving the classification. The localization of discriminative data from local details to the global structure is used to perform an effective classification. Figure 1 shows the block diagram of the proposed method.

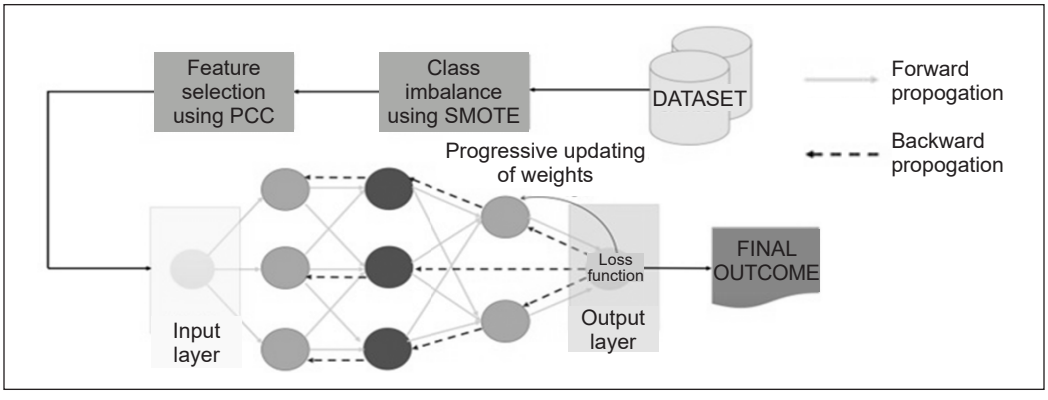


Figure 1. Block diagram of the proposed method

Dataset Acquisition

In this research, two datasets, the Poker hand dataset and the Higgs dataset, are taken from the UCI and Kaggle machine repository. The links for the dataset are: https://archive.ics.uci.edu/ml/datasets/Poker+Hand; and https://archive.ics.uci.edu/ml/datasets/HIGGS.

Few research methodologies have used these datasets, considering their voluminous structure, which consumes much processing time. The proposed method experimented with the PLM training method on this dataset to achieve considerably good results.

The Poker hand dataset includes the 1025010 instances and 11 attributes with categorical and integer features. The nature of this Poker hand dataset is multivariate.

Poker is a 5-card poker hand used in each instance of the dataset, with each card having two attributes (suite and rank) and the poker-hand label. It is an all-categorical trait and highly imbalanced dataset, with the first two classes representing 90% of the samples in both the training and testing sets. In the Higgs dataset, the number of attributes and instances are 28 and 1100000, respectively. Monté Carlo simulations were used to generate the data. The first column is the class label (s for the signal for background), followed by the 28 features (21 low-level features, then 7 high-level features The first 21 features (columns 2–22) are kinematic attributes measured by the accelerator's particle detectors. The final seven features are functions of the first 21 features. These are high-level features developed by physicists to aid in distinguishing between the two groups.

Class Imbalance Processing Using SMOTE

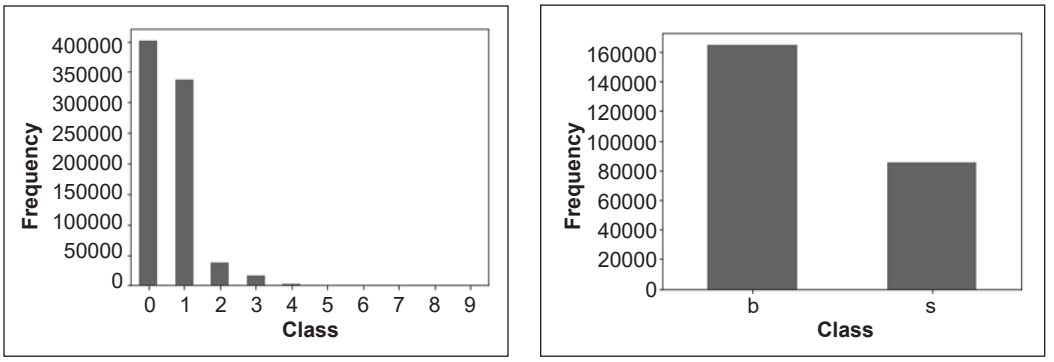
The data acquired from the datasets are processed using the synthetic minority over-sampling technique (SMOTE) approach to avoid issues related to imbalanced data. SMOTE is a classical oversampling in that the number of samples of the minority class is maximized in proportion to the majority class. The main principle of SMOTE is to include new data at random places among the minority data and its neighbors. Initially, the K-nearest neighbors are investigated using minority-class data. Equation 1 shows the interpolation expression of SMOTE.

$$D_i^{'} = D + rand(0,1) \times (NN_i - D) \quad [1]$$

The data sample of minority class samples is denoted as D; the random number between [0,1] is denoted as $rand(0,1)$; the i the nearest neighbors are denoted as NN_i, and the interpolated sample is denoted as. D'_i.

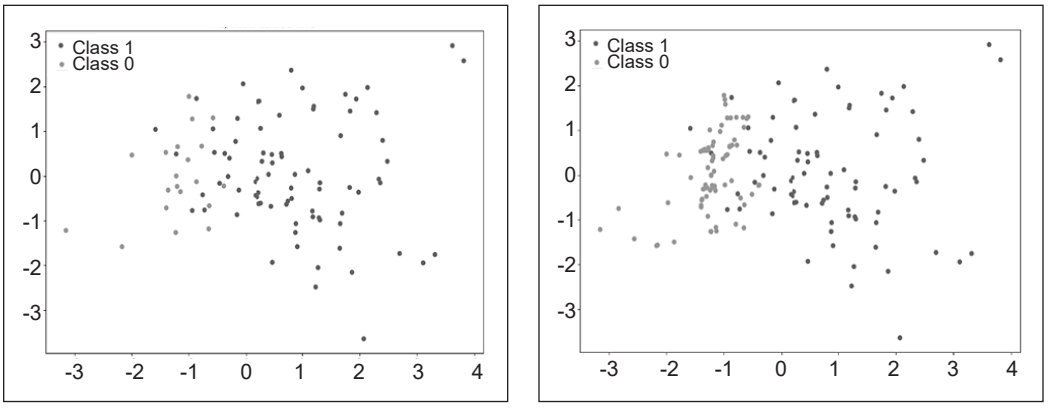
Figure 2 shows the imbalance property of both datasets, whereas Figure 3 depicts the class distribution before and after applying SMOTE in the Higgs Dataset.

The Higgs dataset contained imbalanced data, with the background class representing almost 90% of data and the signal class around 50% after the application of SMOTE. Oversampling equally distributed the classes.



(a) (b)

Figure 2. Depiction of total imbalance distribution of classes in: (a) Poker dataset; and (b) Higgs dataset before applying SMOTE



(a) (b)

Figure 3. Distribution of classes in Higgs dataset: (a)Before; and (b) after applying SMOTE

Feature Selection Using Pearson Correlation Coefficient

Pearson Correlation Coefficient (PCC) is a linear dependence degree measured between the two random features, i.e., real-valued vectors obtained from the dataset. PCC of two variables, D'_1 and D'_2, is generally defined as the ratio between the covariance (COV) of the two variables and the standard deviation's product expressed in Equation 2.

$$\rho_{D_1^{'},D_2^{'}} = \frac{COV(D_1^{'},D_2^{'})}{\sigma_{D_1^{'}}\sigma_{D_2^{'}}} \quad [2]$$

Where the PCC is denoted as $\rho_{D_1^{'},D_2^{'}}$; standard deviations of D'_1 and D'_2 are denoted as $\sigma_{D_1^{'}}$ and $\sigma_{D_2^{'}}$; Hence, the relevant features are selected based on derived PCC, and it is processed further in the ANN with PLM for big data classification. The coefficient correlation value less than 0.5 was not considered for the training dataset.

Classification Using ANN-PLM

PLM Process

The training process, which is performed using PLM, starts from a lower stage and progressively updates the learning weights of all stages to perform the overall training. The phenomena considered in progressive learning is the computation of progressive weight based on cross-entropy loss. This cumulative progressive weight is updated to the previous layer via backpropagation. In normal backpropagation, the gradient value is updated as feedback to the previous layer, but ANN-PLM updates the progressive weight to the previous layer. This progressive weight-based feedback helps to achieve trained layers with optimal performance in an earlier stage compared to the conventional ANN. The PLM is required to obtain the discriminative data from local details to overcome the lower stage's restriction of representation capacity and receptive field. Here, the representation capacity denotes the data training capacity of neurons in an ANN layer, and the receptive field represents the response attainable from the neurons according to the previous stage output. The ANN steadily discovers discriminative data from local (i.e., layer) details to the global structure along with the increment of stages, where the global structure is cumulative of all hidden layers.

In general, the ANN output is the trained weights of the hidden layers, whereas the ANN-PLM's output is the weight of the global structure. The changes in the local layer's weight create an impact on the adjacent layers. Consequently, the global structure varies as a result of progressive learning.

Steps in the PLM Process

The flowchart of the proposed system is shown in Figure 4.

The main objective is to develop progressive training to reduce classification loss in various intermediate stages. Therefore, the convolution block of B_l^{conv} considers the output of stage S_l as input and minimizes vector depiction. $V_l = B_l^{conv}(S_l)$ The classification module B_l^{class} is defined for predicting the probability distribution, whereas B_l^{class} has exponential linear units, batch norm, and two fully connected stages. The probability distribution $y_l = B_l^{class}(V_l)$ is obtained by giving V_l as input to the classification.

The cross-entropy loss L_{CE} expressed in Equation 3 is adopted in PLM for reducing the distance among the label of ground truth y and distribution of prediction probability y_l

$$L_{CE}(y_l, y) = -\sum_{i=1}^{m} y^i \times \log(y_l^i) \quad [3]$$

Where the number of categories is represented as m; the probability that the input X of the category i and stage l is represented as y_l^i.The outputs of multiple previous stages are combined, as shown in Equation 4, to enhance the classification.

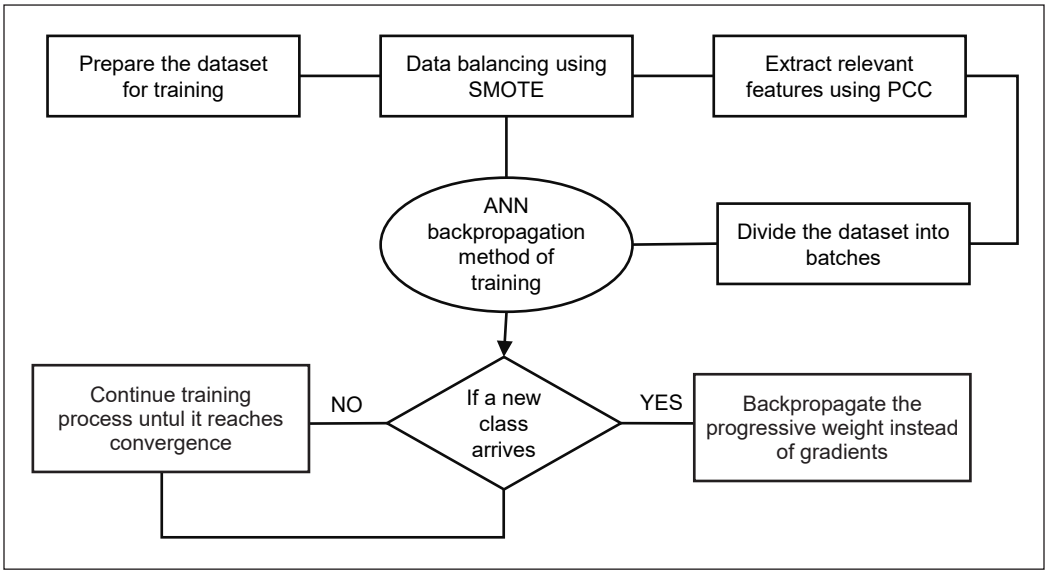


Figure 4. Flowchart of proposed ANN-PLM method

$$V_{concat} = concat[V_{L-S+1}, \ldots, V_{L-1}, V_L] \quad [4]$$

Where the amount of the last stages is represented as S, and it is followed by the classification, $y_{concat} = H_{concat}^{class}(V_{concat})$, where H is an output. Subsequently, the PLM is optimized using the cross-entropy loss, expressed in Equation 5.

$$L_{CE}(y_{concat}, y) = -\sum_{i=1}^{m} y^i \times \log(y_{concat}^i) \quad [5]$$

The parameters used in the current estimation are optimized and are updated in the previous step to help every stage in the PLM operate together in the ANN. The probability distribution of discovery, such as y_l and y_{concat}, is obtained in PLM. The outcome of PLM is derived as Equation 6 when it only uses the y_{concat} in the discovery.

$$C = argmax\,(y_{concat}) \quad [6]$$

The identifications of each stage are complementary and unique; therefore, all outcomes are integrated to obtain the final prediction, as shown in Equation 7, which is modified from Equation 6.

$$C = argmax\left(\sum_{l=L-S+1}^{L} y_l + y_{concat}\right) \quad [7]$$

Pseudocode for ANN-PLM with an Experimental Setup

Input: Initialize the hyperparameters of the network such as learning rate =0.01, hidden layers =10, Number of neurons = 30, Maximum number of epochs =100, batch size =8, Test ratio = 20%, Train ratio=80%and Activation function = Sigmoid.

- m = number of classes
- B_l^{conv} = convolution block
- V_l = vector
- L_{CE} = cross loss entropy
- log = the natural log
- y = ground truth label for i-th sample
- y_l = predicted label for i-th sample
- V_{concat} = outputs of multiple previous stages
- y_{concat} = predicted labels for the concatenated output
- Δy^i = new class sample

Preprocess the input data for classification.

For epochs 1, *N* do # *N* defines the number of epochs

With probability *p* with random learning weight

Calculate $L_{CE}(y_l, y)$ for each class of data

$L_{CE}(y_l, y) = -\sum_{i=1}^{m} y^i \times \log(y_l^i)$ // Calculate loss for each batch of training data as in Equation 4.

Calculate $L_{CE}(y)$ for each class

Repeat

Calculate Δy^i

$L_{CE}(y_l, \Delta y^i) = -\sum_{i=1}^{m}(y^i + \Delta y^i) \times \log(y_l^i + \Delta y^i)$

If ($L_{CE}(y_l, y^i)$=<$L_{CE}(y_l, \Delta y^i)$) // Check, new class data arrived.

$L_{CE}(y_{concat}, y) = -\sum_{i=1}^{m} y^i \times \log(y_{concat}^i)$ // Update learning weights and back-propagate the progressive weight

//Find the loss probabilities

Else continue

End if

End For

Evaluate prediction for test data in the trained model

Compute performance measures.

Output: Classified information of big data.

The input data must be pre-processed before classification during the initialized training phase. The learning weights are randomly initialized, and the training data loss in every batch is calculated using Equation 4. For several epochs, this process is repeated

for every class, and the loss of the training data is evaluated and checked for new class data. If new class data arrives to calculate the new learning weights, the learning weights are updated using Equation 6. Else, continue the epochs. The trained model is evaluated based on the test data prediction, and the performance measure is calculated, resulting in big data classification.

RESULTS AND DISCUSSION

The design and simulation of the proposed method are performed in Python 3.7. The system configurations used to run this big data classification are an i5 processor, 16 GB RAM, and 6 GB GPU. The datasets used to analyze the big data classification using the proposed method are the Poker hand and the Higgs datasets. Of these datasets, 80% were taken for training and 20% for testing. The performance metrics such as accuracy, precision, recall, F-measure, and specificity are used in this study. The ROC curve, Confusion Matrix, was also derived for both datasets. Training validation accuracy and loss are calculated for 100 epochs. Finally, the convergence rate of the datasets concerning accuracy and epochs is also derived to provide more insights into the proposed model.

Performance Analysis of the Proposed Method

Higgs Dataset

Figure 5 shows the ROC Curves for the Higgs dataset. The ROC curve is the reference point for evaluating the classifier's performance. A ROC curve is a graph that displays the performance of the classification model at different classes (0 to 9). Figure 5 observes that the ROC curve of class 9 (area 1.00) reaches a stable point of 1.0 to achieve superior results for the Higgs dataset. Figure 6 displays that training accuracy reaches 0.01820 at 100 epochs, while validation accuracy achieves 0.01860 at 100 epochs. Figure 7 shows the graphical representation of training and validation loss for the Higgs dataset. Training loss values stabilize at 5.020 for 100 epochs, while the validation loss stabilizes at 5.009.

Figure 8 shows the early convergence of the ANN-PLM method as compared to ANN. The epochs required to attain convergence is 25 compared to the conventional ANN method, which takes 40 epochs with a uniform accuracy rate.

Figure 8 also shows the Graphical representation of accuracy performance for the Higgs dataset. It can be observed that the proposed ANN-PTM with SMOTE achieved

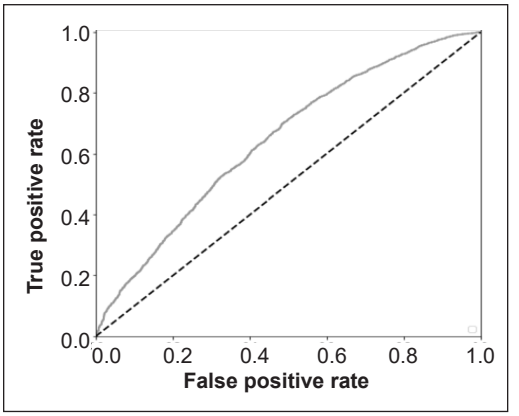


Figure 5. ROC characteristics of Higgs dataset

better accuracy, 0.98, at a cut-off range of 25 epochs, where the accuracy starts to stabilize and is maintained the same till it reaches 100 epochs. While considering the ANN process, it achieved an accuracy of 0.95 at a cut-off range of 40 epochs. The performance evaluation of the proposed method with the Higgs dataset with and without PCC and SMOTE is shown in Table 1. The ANN-PLM performs better with and without PCC and SMOTE than the ANN, KNN, and SVM.

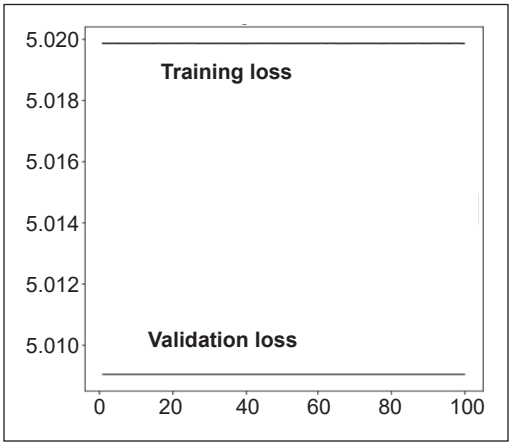


Figure 6. Epochs vs. training-validation loss

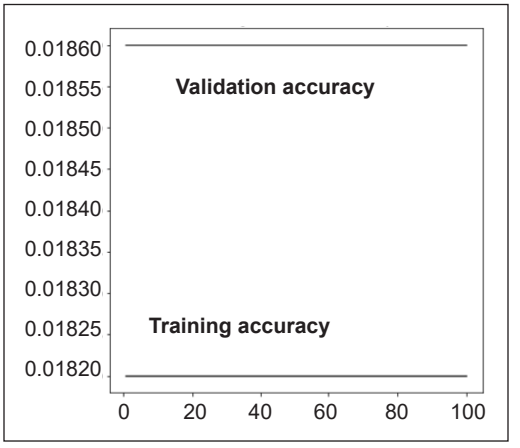


Figure 7. Epochs vs. training-validation accuracy

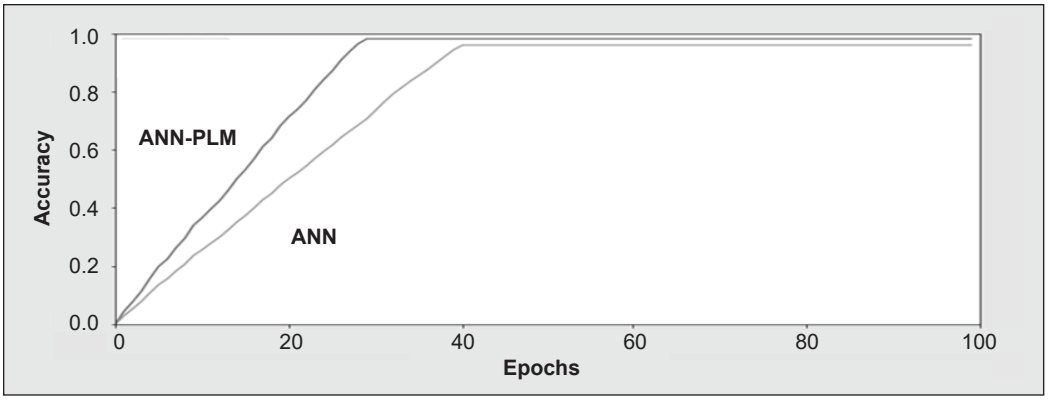


Figure 8. Convergence graph of Higgs dataset with ANN and ANN-PLM

Table 1
Performance evaluation of the proposed method for the Higgs dataset

Feature selection	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	Fmeasure (%)	Specificity (%)
Without PCC	ANN	91.804	91.372	89.264	91.176	89.955
	KNN	93.882	94.217	94.153	94.386	96.357
	SVM	94.454	95.280	95.969	95.373	96.116
	ANN-PLM	97.220	95.166	96.671	96.092	96.104

Table 1 *(continue)*

Feature selection	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	Fmeasure (%)	Specificity (%)
With PCC	ANN	93.110	93.948	94.893	95.057	94.463
	KNN	96.170	97.006	95.487	96.533	95.382
	SVM	97.597	97.637	96.784	97.362	96.735
	ANN-PLM	99.329	99.121	99.004	99.536	99.668
Without SMOTE	ANN	90.60	70.40	80.39	78.50	80.39
	KNN	93.689	96.456	95.336	91.337	92.896
	SVM	92.081	96.189	94.667	95.321	88.542
	ANN-PLM	96.227	95.780	96.548	97.168	78.660
With SMOTE	ANN	93.80	77.4	87.1	82.0	86.10
	KNN	94.657	93.778	96.932	92.436	93.786
	SVM	93.180	97.005	95.457	96.879	89.865
	ANN-PLM	97.325	96.532	97.278	98.568	80.578

Pokers Dataset

In Figure 9, the ROC curve contains two constraints, (i.e.) True Positive Rate (TPR) and False Positive Rate (FPR). In general, a ROC of more than 0.9 is considered outstanding. Figure 9 shows that the ROC curve reaches the value of 0.98, which is closer to 1, i.e., it produces better classification results for the Poker hand dataset. The ROC curve signifies that all ten classes are properly classified within the range of 0.97 and 1.

Figure 10(a) shows the graphical representation of Training and Validation accuracy for the Higgs dataset. The training accuracy reaches 0.821 at 100 epochs, while the validation accuracy achieves 0.86 at 100. As shown in Figure 10 (b), training loss reaches -1.3 for 100 epochs, while the validation loss reaches -1.4 at 100 epochs.

Figure 11 shows the graphic representation of accuracy performance for the Poker hand dataset. Figure 11 shows that the proposed ANN-PTM achieved better accuracy, 0.92,

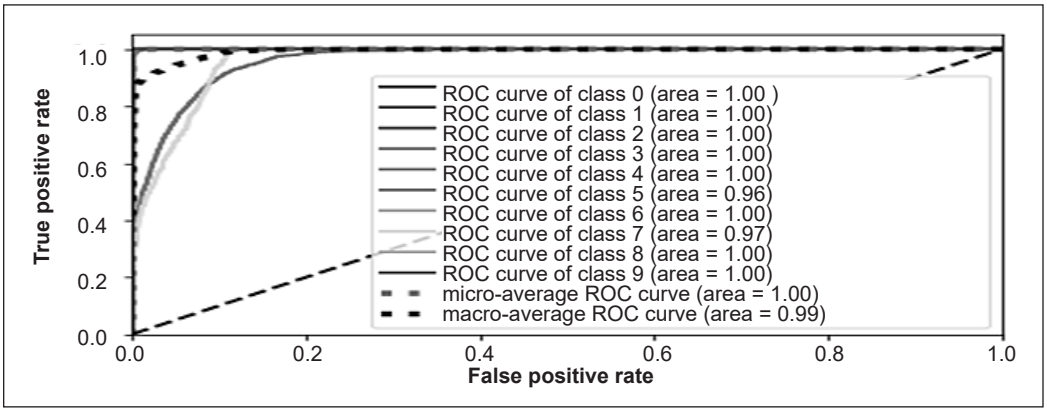
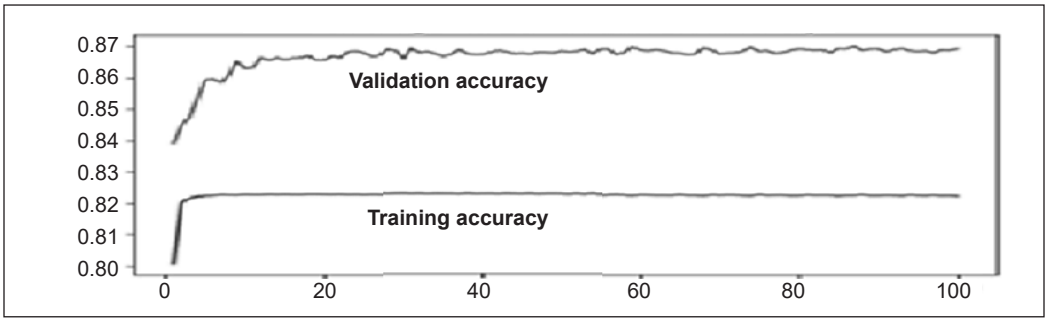
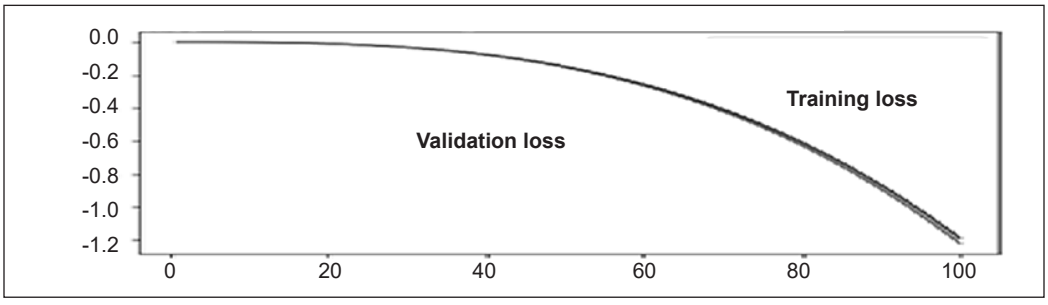


Figure 9. ROC characteristics for Pokers data



(a)



(b)

Figure 10. (a) Training and validation accuracy; and (b) Training and validation loss

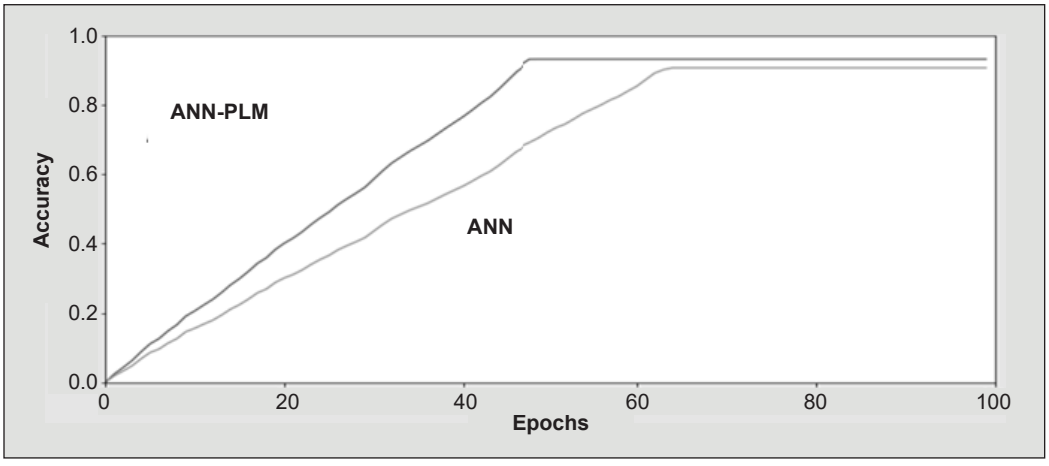


Figure 11. Convergence graph of Pokers dataset with ANN and ANN-PLM

at a cut-off range of 50 epochs, where the accuracy starts to stabilize and is maintained the same until it reaches 100 epochs. While considering the ANN process, it achieved an accuracy of 0.89 at a cut-off range of 65 epochs.

Table 2 shows the performance evaluation of the proposed method with the Pokers' dataset with and without PCC and SMOTE. The ANN-PLM performs better with and without PCC and SMOTE than the ANN, KNN, and SVM.

Table 2
Performance evaluation of the proposed method for the Poker hand dataset

Feature selection	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Specificity (%)
Without PCC	ANN	91.804	91.372	89.264	91.176	89.955
	KNN	93.882	94.217	94.153	94.386	96.357
	SVM	94.454	95.280	95.969	95.373	96.116
	ANN-PLM	97.220	95.166	96.671	96.092	96.104
With PCC	ANN	93.110	93.948	94.893	95.057	94.463
	KNN	96.170	97.006	95.487	96.533	95.382
	SVM	97.597	97.637	96.784	97.362	96.735
	ANN-PLM	99.329	99.121	99.004	99.536	99.668
Without Smote	ANN	90.60	70.40	80.39	78.50	80.39
	KNN	93.689	96.456	95.336	91.337	92.896
	SVM	92.081	96.189	94.667	95.321	88.542
	ANN-PLM	96.227	95.780	96.548	97.168	78.660
With Smote	ANN	93.80	77.4	87.1	82.0	86.10
	KNN	94.657	93.778	96.932	92.436	93.786
	SVM	93.180	97.005	95.457	96.879	89.865
	ANN-PLM	97.325	96.532	97.278	98.568	80.578

Comparative Analysis of Other Classification Methods Using Higgs and Poker Dataset

The existing research on big data classification, such as MF [28], PGS-ISSA [19], Genetic Programming, Multilayer feedforward Backpropagation, and AEB [20] are used to compare the proposed method. A comparison is made between the Poker hand and the Higgs datasets. In that, the PGS-ISSA [19] is analyzed for the Poker hand dataset, and AEB [20] is analyzed for the Higgs dataset, while MF [28] is used for both data set comparisons. Tables 3 and 4 show the comparative analysis of the proposed method with Higgs and Poker hand datasets, respectively. Tables 3 and 4 show that the proposed method provides better performance than the existing methods. The graphical representation is shown in Figures 12 and 13 for the Higgs and Pokers datasets, respectively.

Table 3
Comparative analysis of the Higgs dataset

Method	Accuracy (%)
MF	52
PGS-ISSA	64.72
Cartesian Genetic Programming Using Random Sampling	65
Proposed method	96.566

Table 4
Comparative analysis of Poker hand dataset

Method	Accuracy (%)
MF	50
AEB	89.93
Multilayer Feedforward Propagation Method	94
Proposed method	98.629

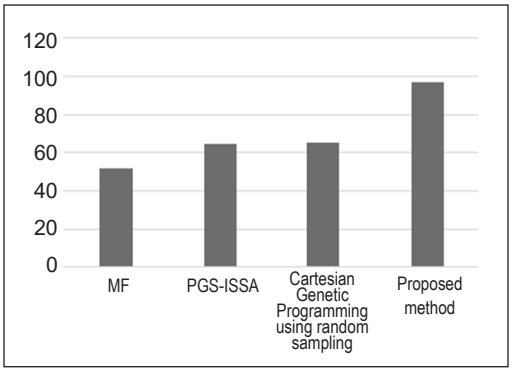


Figure 12. Accuracy comparison of Higgs dataset

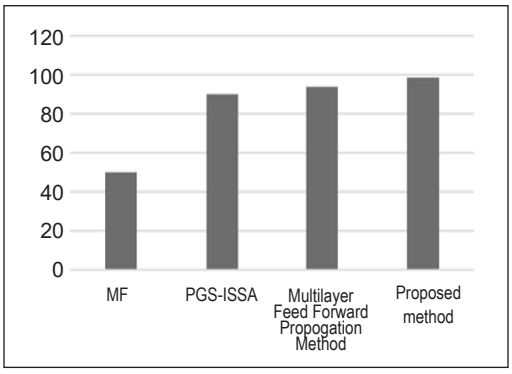


Figure 13. Accuracy comparison of Poker dataset

CONCLUSION

This research proposed a methodology for the classification of huge amounts of data. The strategy combines the use of the ANN-PLM technique, which stands for Artificial Neural Network with Progressive Learning Method, with feature selection based on the Pearson Correlation Coefficient (PCC). The Synthetic Minority Over-sampling Technique (SMOTE) is used as a means of data augmentation in the classification procedure to mitigate the issues related to class imbalance and overfitting. The PCC feature selection approach is utilized to identify the most pertinent features from the feature vector, improving the classification performance. The Pearson correlation coefficient (PCC) aids in the identification of the most suitable collection of features by evaluating their correlation with the target variable. This process enhances the classification model's ability to differentiate across classes.

In addition, we integrate the notion of discriminative data localization, which entails iteratively adjusting the weights of the neural network model by considering both local particulars and global structure. This localization methodology allows the network to concentrate on significant patterns and characteristics in the data, increasing the categorization accuracy. The experimental results demonstrate that the proposed ANN-PLM strategy exhibits superior performance compared to traditional ANN approaches in terms of convergence epochs and other classification performance criteria. The suggested method demonstrates significantly improved accuracy on the Higgs and Poker datasets when utilizing the PLM technique compared to currently available methods.

In summary, the efficacy of integrating ANN-PLM, PCC-based feature selection, SMOTE, and data localization approaches for the classification of huge data is demonstrated by our suggested strategy. The findings underscore the effectiveness of the suggested approach to precision, convergence speed, and overall classification performance. It underscores its potential as a reliable and efficient option for addressing classification issues involving large datasets.

In future research, it would be beneficial to evaluate the performance of this method on imbalanced data, as the current study only assessed its effectiveness on balanced data. This limitation can be further experimented with.

ACKNOWLEDGEMENTS

The author extends sincere gratitude to Dr. Kayarvizhy N for invaluable advice and constructive feedback throughout the development of this manuscript. Her expertise and guidance were instrumental in shaping the final version of this paper. The author also wishes to acknowledge the open-access resources and online communities that offered valuable insights and data, enabling the author to pursue this research independently.

REFERENCES

Abhilasha, A., & Naidul, P. A. (2022). Self-boosted with dynamic semi-supervised clustering method for imbalanced big data classification. *International Journal of Software Innovation*, *10*(1), 1-24. https://doi.org/10.1007/s11042-022-12038-4

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 & Systems, 14(*4), 177-189. https://doi.org/10.22266/ijies2021.0831.17

Al-Thanoon, N. A., Algamal, Z. Y., & Qasim, O. S. (2021). Feature selection based on a crow search algorithm for big data classification. *Chemometrics and Intelligent Laboratory Systems*, *212*, Article 104288. https://doi.org/10.1016/j.chemolab.2021.104288

Banchhor, C., & Srinivasu, N. (2021). Analysis of *Bayesian* optimization algorithms for big data classification based on map reduce framework. *Journal of Big Data*, *8(*1*)*, Article 81. https://doi.org/10.1186/s40537-021-00464-4

Basgall, M. J., Naiouf, M., & Fernández, A. (2021). FDR2-BD: A fast data reduction recommendation tool for tabular big data classification problems. *Electronics, 10(*15), Article 1757.https://doi.org/10.3390/electronics10151757

BenSaid, F., & Alimi, A. M. (2021). Online feature selection system for big data classification based on multi-objective automated negotiation. *Pattern Recognition*, *110*, Article 107629. https://doi.org/10.1016/j.patcog.2020.107629

Brahmane, A. V., & Krishna, B. C. (2021). Big data classification using deep learning and apache spark architecture. *Neural Computing and Applications, 33*(2),15253-15266. https://doi.org/ 10.1007/s00521-021-06145-w

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., & Alahari, K. (2018, September 8-14). *End-to-end incremental learning*. [Paper presentation]. Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany. https://doi.org/10.48550/arXiv.1807.09536

Chatterjee, S., Javid, A. M., Sadeghi, M., Mitra, P. P., & Skoglund, M. (2017). Progressive learning for systematic design of large neural networks. *arXiv,* Article 1710.08177. https://doi.org/10.48550/arXiv.1710.08177

Dubey, A. K., Kumar, A., & Agrawal, R. (2021). An efficient ACO-PSO-based framework for data classification and preprocessing in big data. *Evolutionary Intelligence, 14*, 909-922. https://doi.org/10.1007/s12065-020-00477-7

Du, R., Xie, J., Ma, Z., Chang, D., Song, Y. Z., & Guo, J. (2021). Progressive learning of category-consistent multi-granularity features for fine-grained visual classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(12), 9521-9535. https://doi.org/10.1109/TPAMI.2021.3126668

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. htttps://doi.org/10.1109/ACCESS.2022.3142888

Hassib, E. M., El-Desouky, A. I., Labib, L. M., & El-Kenawy, E. S. M. (2020). WOA+ BRNN: An imbalanced big data classification framework using whale optimization and deep neural network. *Soft Computing, 24*(8), 5573-5592. https://doi.org/10.1007/s00500-019-03901-y

Jain, D. K., Boyapati, P., Venkatesh, J., & Prakash, M. (2022). An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. *Information Processing Management, 59*(1), Article 102758. https://doi.org/10. 1016/j.ipm.2021.102758

Juez-Gil, M., Arnaiz-Gonzalez, A., Rodriguez, J. J., Lopez-Nozal, C., & Garcia-Osorio, C. (2021). Approx-SMOTE: Fast SMOTE for big data on *Apache* spark. *Neurocomputing, 464*, 432-437. https://doi.org/10.1016/j.neucom.2021.08.086

Kantapalli, B., & Markapudi, B. R. (2023). SSPO-DQN spark: Shuffled student psychology optimization based deep Q network with spark architecture for big data classification. *Wireless Networks, 29*(1),369-385. https://doi.org/10.1007/s11276-022-03103-9

Li, Z., Liu, C., Yuille, A., Ni, B., Zhang, W., & Gao, W. (2021, June 19-25). *Progressive stage-wise learning for unsupervised feature representation enhancement*. [Paper presentation]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, USA. https://doi.org/10.48550/arXiv.2106.05554

Movassagh, A. A., Alzubi, J. A., Gheisari, M., Rahimi, M., Mohan, S., Abbasi, A. A., & Nabipour, N. (2021). Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. *Journal of Ambient Intelligence and Humanized Computing*, *14*, 6017–6025. https://doi.org/10.1007/s12652-020-02623-6

Mujeeb, S. M., Sam, R. P., & Madhavi, K. (2021). Adaptive exponential bat algorithm and deep learning for big data classification. *Sādhanā, 46*(1), Article 15. https://doi.org/10.1007/ s12046-020-01521-z

Park, S. T., Kim, D. Y., & Li, G. (2021). An analysis of environmental big data through the establishment of emotional classification system model based on machine learning: Focus on multimedia contents for portal applications. *Multimedia Tools and Applications, 80*, 34459-34477. https://doi.org/ 10.1007/s11042-020-08818-5

Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 2001-2010). IEEE Publishing. https://doi.org/10.48550/arXiv.1611.07725

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu R. & Hadsell, R. (2016). Progressive neural networks. *arXiv,* Article 1606.04671. https://doi.org/10.48550/arXiv.1606.04671

Siddiqui, Z. A., & Park, U. (2021). Progressive convolutional neural network for incremental learning. *Electronics, 10*(16), Article 1879. https://doi.org/10.3390/electronics10161879

Sleeman IV, W. C., & Krawczyk B. (2021). Multi-class imbalanced big data classification on spark. *Knowledge-Based Systems, 212*, Article 106598. https://doi.org/10.1016/j.knosys. 2020. 106598

Venkatesan, R., & Er, M. J. (2016). A novel progressive learning technique for multi-class classification. *Neurocomputing*, *207*, 310-321. https://doi.org/10.1016/j.neucom.2016.05.006

Wang, H., Xiao, M., Wu, C., & Zhang, J. (2021). Distributed classification for imbalanced big data in distributed environments. *Wireless Networks*, *2021,* 1-12. https://doi.org/10.1007/s11276-021-02552-y

Xing, W., & Bei, Y. (2019). Medical health big data classification based on KNN classification algorithm. *IEEE Access, 8*, 28808-28819. https://doi.org/10.1109/ACCESS.2019.2955754

Zhou, J., Li, J., Wang, C., Wu, H., Zhao, C., & Wang, Q. (2021). A vegetable disease recognition model for complex background based on region proposal and progressive learning. *Computers and Electronics in Agriculture, 184*, Article 106101. https://doi.org/10.1016/j.compag.2021.106101