



## Development of Hybridized Algorithms for a Customer Information Profiling System

\*<sup>1</sup>Ayegbo, O.J., <sup>2</sup>Olabiyisi S.O., <sup>2</sup>Adeosun O.O., & <sup>3</sup>Olaleye O.J.

<sup>1</sup>Department of Computer Science, Auchi Polytechnic, Auchi, Edo State, Nigeria.

<sup>2</sup>Department of Computer Science, Ladoko Akintola University of Technology, Ogbomosho, Oyo State, Nigeria.

<sup>3</sup>Department of Computer Science, Bell University of Technology, Ota, Ogun State, Nigeria.

\*Corresponding author email: femiayegbo@yahoo.com

### Abstract

Accurate customer information profiling remains a critical challenge for banks and businesses due to the massive volume of customer data and the need for timely, targeted service delivery. The conventional Back Propagation Neural Network (BPNN) algorithm suffers from high computational time, limiting its efficiency in large-scale data analysis. To address this limitation, this study developed a hybridized MapReduce Back Propagation Neural Network (MRBPNN) technique for customer information profiling. A dataset comprising 2,000 customers' demographic and socioeconomic records, each with 35 attributes and 5 status updates, was obtained from selected banks in Edo State. The data were divided into training and testing sets and evaluated using a ten-fold cross-validation approach. The hybrid MRBPNN model was implemented in MATLAB R2023a and compared with the conventional BPNN algorithm based on false positive rate (FPR), precision, recall, F-measure, accuracy, and computational time. Experimental results revealed that MRBPNN achieved superior performance with an FPR of 5.83%, accuracy of 94.59%, precision of 99.31%, recall of 94.64%, F-measure of 96.92%, and computational time of 13.23 seconds. In contrast, BPNN recorded an FPR of 15.61%, accuracy of 84.97%, precision of 97.96%, recall of 85.04%, F-measure of 91.05%, and computational time of 26.69 seconds. The findings demonstrate that the developed MRBPNN significantly reduces computational time and enhances classification accuracy compared to BPNN, making it an efficient and scalable technique for customer information profiling in financial institutions.

**Keywords:** MapReduce, Hybridized Algorithms, Back Propagation, Neural Network, Machine Learning

### Introduction

In the contemporary data-driven economy, financial institutions, enterprises, and organizations increasingly acknowledge that effective customer information profiling constitutes a crucial determinant of sustainable competitive advantage (Annamalai & Muthiah, 2022). As markets grow more volatile and consumer expectations continually evolve, the capacity to derive actionable insights from extensive datasets has become a strategic imperative. Data analytics now represents a valuable "digital asset," empowering firms to enhance marketing precision, optimize the utilization of resources, strengthen customer engagement, and ultimately increase overall profitability (AlFayez & Bouhamed, 2023). Customer profiling entails the systematic development of a comprehensive representation of target customers by integrating demographic variables (such as age, gender, and geographic location), socioeconomic indicators (including income level, occupation, and educational attainment), and psychographic dimensions (such as lifestyle, attitudes, and aspirations) (Arockiam & Pushpanathan, 2023). This multidimensional analytical framework enables organizations to design and deliver personalized products and services that effectively correspond to customer preferences. However, ensuring the accuracy and timeliness of customer profiling remains a considerable challenge, primarily due to the exponential growth, diversity, and complexity of customer data, as well as the need for high-speed, scalable analytical solutions (Awasthy & Valivartha, 2023). Among existing computational approaches, the

Backpropagation Neural Network (BPNN), has been extensively utilized in customer information profiling because of its robust capability in pattern recognition and adaptive learning (Bergui *et al.*, 2024). Nonetheless, BPNN models are constrained by high computational time and scalability limitations when processing large, distributed datasets; this reduces their utilizations in real-time or large-scale customer analytics systems used by financial institutions.

To bridge this gap, hybrid solutions that combine distributed processing frameworks (MapReduce/Hadoop/Spark) with neural learning (BPNN) was developed to merge distributed efficiency with the robust learning capabilities of neural networks (Biswas *et al.*, 2022). The MapReduce-Backpropagation hybrid (MRBPNN) promises reduced wall-clock training time while keeping competitive predictive performance for large-scale customer profiling tasks (Cui *et al.*, 2022).

### Statement of the problems

Banks, mostly, deal with numerous customers' data, and to accurately and speedily identify active customers to offer the right products and services at the right time has become a major concern.

Slow convergence speed and high computational time are the major drawbacks limiting the existing Back Propagation neural network algorithm for customer information profiling (Chandrasekar & Ponnusamy, 2025). To solve this problem, this research developed a hybridized Map-Reduce Back Propagation Neural Network (MRBPNN) technique for customer information profiling.

### Aim and Objectives of the Study

The primary objective of this study is to design a hybridized Map-Reduce Back Propagation Neural Network (MRBPNN) technique for effective Customer Information Profiling. The specific goals are to:

- i. Design and formulate the MRBPNN technique for customer information profiling;
- ii. Implement the proposed technique using MATLAB (R2023a) software;
- iii. Assess the performance of the developed MRBPNN model based on key evaluation indicators such as false positive rate, accuracy, precision, recall, F-measure, and computational time and
- iv. Compare the performance outcomes of the MRBPNN technique with the conventional Back Propagation Neural Network (BPNN) to evaluate its overall efficiency and computational effectiveness.

### MapReduce Neural Network (MRNN)

The explosive growth of big data across finance, healthcare, social media, and IoT has necessitated distributed learning frameworks capable of processing massive datasets efficiently. The Map-Reduce paradigm provides a data-parallel model via Map and Reduce phases, offering scalability, fault tolerance, and ease of deployment on commodity clusters (Dampfhofer *et al.*, 2023). MRNN frameworks adapt standard neural network training by decomposing forward/backward passes into distributed Map tasks (local forward and gradient computation) and Reduce tasks (aggregation and parameter synchronization), enabling parallelization across many nodes and greater throughput on extremely large datasets.

Numerous studies and applied systems show Map-Reduce (and Map-Reduce-style frameworks such as Hadoop and Spark) used successfully to scale neural network workloads: examples include Map-Reduce-based convolutional algorithms, parallel Winograd approaches for convolution, distributed deep learning pipelines for image/medical/recommendation workloads, and IoT pipelines that use Map-Reduce for pre-processing and distributed model training. Hybrid MRNNs report substantial speedups in wall-clock time for large datasets though the tradeoffs include added communication and synchronization cost and challenges supporting low-latency streaming updates (Dong and Wang, 2025). Typical performance improvements are achieved when communication compression, asynchronous aggregation, or hierarchical aggregation schemes reduce synchronization overhead. Despite these benefits, MRNNs face research challenges: (i) communication/synchronization overhead for iterative gradient updates; (ii) data partitioning and load balancing across heterogeneous compute nodes; (iii) support for streaming or online learning; and (iv) efficient fault tolerance and straggler mitigation in production clusters (Gao *et al.*, 2023). Recent reviews emphasize moving from pure Map-Reduce to more flexible distributed frameworks (Spark, Ray, parameter server designs) or hybrid strategies that combine the simplicity of Map-Reduce batch processing with dedicated asynchronous gradient aggregation for iterative learning (Gazis & Katsiri, 2022).

### **Backpropagation Neural Network (BPNN)**

Backpropagation remains a foundational supervised learning algorithm for multilayer networks, using gradient-based optimization to update weights by propagating the error from outputs to inputs. BPNN variants and improvements (regularization, advanced optimizers, second-order methods, Bayesian approaches) have been proposed recently to improve convergence and generalization. Hybridizing BPNN with metaheuristics such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and other population-based methods continues to be an active area, particularly in domains where weight initialization and hyperparameter tuning are critical for performance (Gattoju and Nagalakshmi, 2023).

However, BPNN training on large datasets becomes computationally demanding: memory pressure, long epochs, vanishing/exploding gradients (in deep architectures), and sensitivity to hyper-parameters limit vanilla BPNN applications at scale (Hosseini et al., 2022). To mitigate these issues, researchers have developed distributed BPNN training paradigms (MapReduce-based, Spark-based, parameter-server and federated learning approaches) that demonstrate substantial speedups while introducing challenges such as stale gradients, consistency models, and communication bottlenecks. Hybrid strategies that incorporate gradient compression, asynchronous updates, and hardware acceleration (GPU/TPU) improve efficiency and real-time viability (Hamdi et al., 2024).

### **Research Gaps**

The existing algorithm for customer information profiling suffers from slow convergence speed and high computational time. Hence, to address these limitations, the research developed a hybridized Map-Reduce-Back-propagation Neural Network (MRBPNN) technique in order to improve the efficiency and the accuracy of customer information profiling.

### **Methodology**

#### **Research Approach**

In this research, a hybridized Map-Reduce Back Propagation Neural Network (MRBPNN) technique was developed for bank customer information profiling using selected banks in Edo State as a case study. The technique was implemented in five divisions, these include: data gathering, data pre-processing, data storage, technique development, data classification and decision making.

#### **Data Collection**

The data collection phase involved obtaining customer information from selected banking institutions. A total of 2,000 customer records were compiled, each containing 28 variables spanning socioeconomic, psychographic, and demographic categories, along with approximately five status updates per customer. The dataset was partitioned into **training** and **testing** subsets after preprocessing to ensure consistency, quality, and analytical readiness. The collected data encompassed various indicators of customer behavior, including activity-based statistics such as the number of likes, status posts, group memberships, tags, and events. Demographic and socioeconomic variables—such as age, gender, marital status, occupation, and account type—were incorporated due to their established influence on financial behavior (Jansen et al., 2024). Furthermore, egocentric network parameters (e.g., number of friends, network density, brokerage, and betweenness) were extracted to provide additional insight into customers' social interactions, which serve as valuable proxies for personality and behavioral profiling.

Figure 1 showed the raw bank customers' data.

W	X	Y	Z	AA	AB	AC	AD
TRANSACTION	TRANSACTION_DATE	NUMBER_OF_TRANSACTION	ACCOUNT_BALANCE	ACCOUNT_STATUS	LOAN_STATUS	LOAN_TYPE	LOAN_AMOUNT
DEPOSIT	13-09-2023		10	75,000 ACTIVE	ACTIVE	MORTGAGE	5,000,000
DEPOSIT	23-07-2024		37	300,000 ACTIVE	ACTIVE	MORTGAGE	2,000,000
WITHDRAW	23-01-2024		59	250,000 ACTIVE	PAID OFF	AUTOMOBILE	1,000,000
WITHDRAW	23-07-2023		20	100,000 ACTIVE	DEFAULTED	EDUCATION	500,000
TRANSFER	23-06-2024		25	500,000 INACTIVE	NO LOAN		
DEPOSIT	23-03-2023		45	700,000 ACTIVE	ACTIVE	MORTGAGE	5,000,000
DEPOSIT	23-06-2024		50	150,000 ACTIVE	ACTIVE	AUTOMOBILE	3,000,000
DEPOSIT	21-07-2023		60	400,000 ACTIVE	ACTIVE	MORTGAGE	1,000,000
DEPOSIT	21-07-2024		37	500,000 ACTIVE	NO LOAN		
DEPOSIT	20-08-2024		100	2,000,000 ACTIVE	ACTIVE	BUSINESS	8,500,000
WITHDRAW	13-09-2023		25	120,000 ACTIVE	PAID OFF	AUTOMOBILE	2,000,000
WITHDRAW	23-07-2024		30	180,000 ACTIVE	ACTIVE	AUTOMOBILE	1,000,000
TRANSFER	23-07-2023		70	600,000 ACTIVE	ACTIVE	PERSONAL	500,000
TRANSFER	23-07-2023		80	750,000 ACTIVE	ACTIVE	BUSINESS	3,500,000
TRANSFER	23-06-2024		40	30,000 ACTIVE	ACTIVE	AUTOMOBILE	4,000,000
TRANSFER	23-07-2024		35	800,000 ACTIVE	ACTIVE	EDUCATION	250,000
DEPOSIT	23-06-2023		24	1,500,000 ACTIVE	NO LOAN		
DEPOSIT	21-01-2024		150	1,200,000 ACTIVE	ACTIVE	MORTGAGE	5,000,000
DEPOSIT	24-06-2024		120	60,000 ACTIVE	PAID OFF	AUTOMOBILE	3,000,000
DEPOSIT	20-08-2023		46	45,000 ACTIVE	ACTIVE	EDUCATION	1,000,000

Figure 1: Sample of Raw Bank Customers' Data

**Data Pre-Processing**

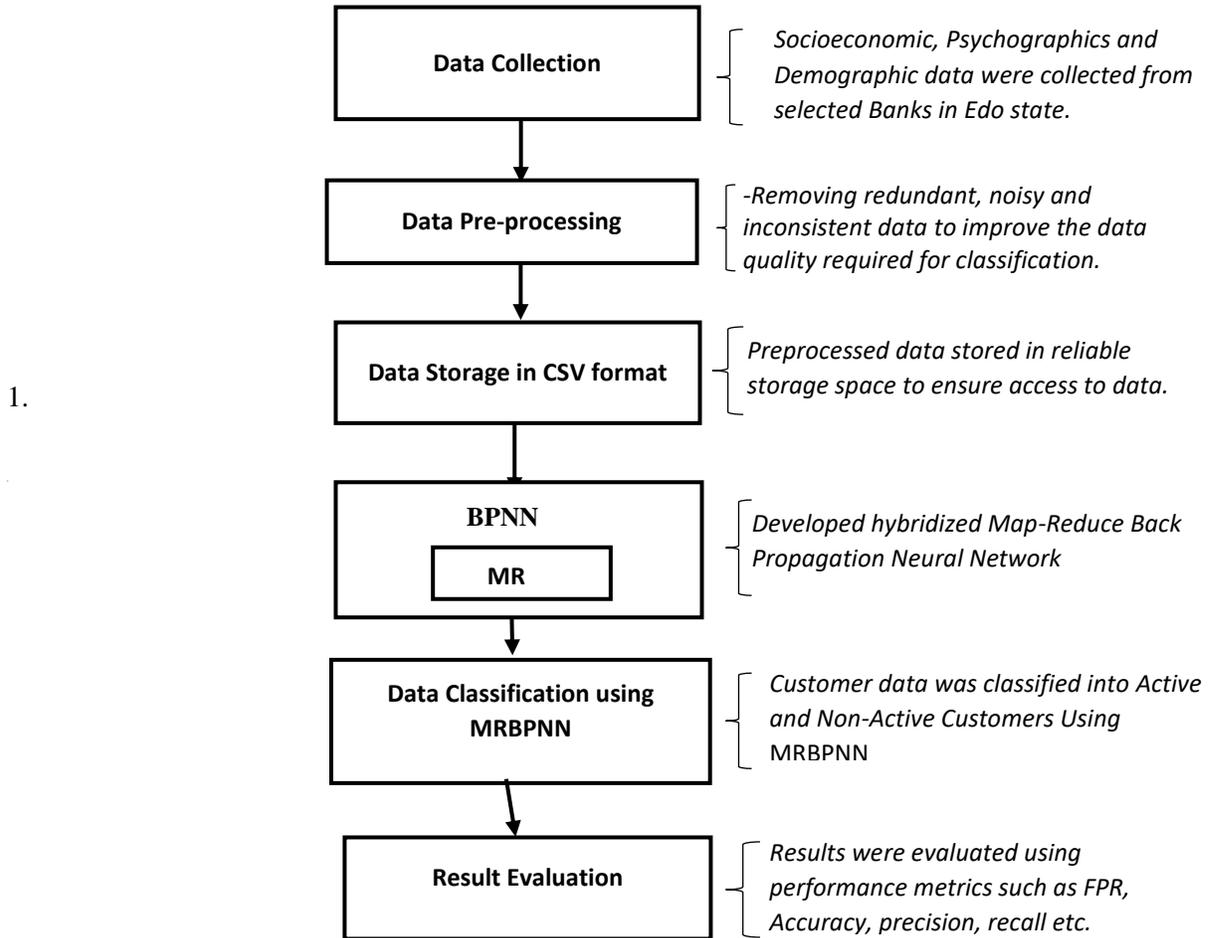
The second phase, data pre-processing, was undertaken to enhance data quality and ensure suitability for classification tasks. This phase involved eliminating redundant, noisy, and inconsistent records to improve analytical precision and optimize storage efficiency. The procedure comprised three key stages: Integration, in which the datasets were extracted, transformed into a standardized format, and loaded into a unified storage framework; Cleaning, where the data were validated for accuracy, completeness, and consistency, with irrelevant or erroneous entries corrected or removed; and Redundancy Elimination, which employed filtering and compression techniques to minimize duplication and enhance data integrity. The outcome of this process is illustrated in Figure 2, showing the pre-processed bank customer dataset.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1	GENDER	AGE	RELIGION	MARITAL	OCCUPATI	EMPLOYM	MONTHLY	ACCOUNT	EDUCATIC	TRANSACTION	NUMBER	ACCOUNT	ACCOUNT	LOAN_STA	LOAN_TYF	LOAN_AM	LOAN_TEF	MTHTLY	LC	LOAN_QUALIFICATION
2	1	1.230543	0	1	133	1	-0.89942	3	0	0	-1.13985	-0.63278	0	0	5	0.984301	1.767907	0.474247	0	
3	0	-0.21912	1	1	98	2	-0.29427	3	2	0	-0.52058	-0.54968	0	0	5	-0.21252	0.952467	-0.16017	1	
4	1	1.3802	0	1	118	2	-1.00045	0	6	2	-0.01598	-0.56815	0	3	0	-0.61145	-0.67841	-0.31878	0	
5	1	0.407428	0	1	55	2	-0.19221	2	7	2	-0.91049	-0.62355	0	1	2	-0.81092	-1.49385	-0.63599	0	
6	0	-1.31363	0	2	90	2	0.110883	3	5	0	-0.33709	-0.40194	0	0	5	0.984301	0.952467	0.474247	1	
7	0	-1.2388	0	2	103	0	-0.4953	3	0	0	-0.22241	-0.60508	0	0	0	0.186423	0.952467	-0.00157	1	
8	1	-0.93949	1	1	133	0	-1.00045	3	3	0	0.006953	-0.51274	0	0	0	5	-0.61145	-0.67841	-0.31878	1
9	1	1.904001	1	1	98	2	-0.29324	3	6	0	0.924394	0.078221	0	0	1	2.380587	0.952467	0.315642	1	
10	1	-1.38846	0	2	91	0	-0.59633	1	7	2	-0.79581	-0.61616	0	3	0	-0.21252	-0.67841	-0.16017	1	
11	0	-1.53812	0	2	133	0	-0.89942	0	2	2	-0.68113	-0.594	0	0	0	-0.61145	-0.67841	-0.31878	0	
12	0	-1.31363	0	1	45	0	-0.39427	0	0	1	0.236313	-0.43887	0	0	6	-0.81092	-1.49385	-0.63599	1	
13	1	-0.1912	0	1	67	2	-1.00045	0	11	1	0.465673	-0.38347	0	0	1	0.385893	0.952467	-0.00157	1	
14	1	0.781571	0	1	67	2	-0.19221	2	11	1	-0.45177	-0.6494	0	0	0	0.585362	0.952467	0.077735	1	
15	1	1.305372	0	1	63	2	-0.79839	0	11	1	-0.56645	-0.365	0	0	2	-0.31906	-2.3093	-0.63599	0	
16	0	0.108113	0	1	68	2	-0.4953	0	0	0	2.071195	-0.21726	0	0	5	0.984301	0.952467	0.474247	1	
17	0	0.706742	1	1	68	2	-1.00045	1	11	0	1.383114	-0.63832	0	3	0	0.186423	0.137026	-0.00157	1	
18	1	1.904001	2	1	134	2	-1.10148	0	2	0	-0.31415	-0.64386	0	0	2	-0.61145	-0.67841	-0.31878	1	
19	1	1.455029	0	0	134	2	-0.29324	2	0	2	-0.56645	-0.63278	0	3	6	-0.81092	-1.49385	-0.63599	0	
20	1	0.781571	0	1	122	2	0.110883	1	7	1	-0.56645	-0.56815	0	0	0	0.186423	-0.67841	-0.00157	1	
21	0	1.978829	0	0	97	2	-0.89942	2	11	0	-0.31415	-0.62355	0	0	5	-0.61145	-1.49385	-0.31878	1	
22	1	1.080886	0	1	47	2	1.626335	2	5	0	-0.56645	0.299833	0	0	6	-0.81092	-1.49385	-0.63599	1	
23	1	-1.16397	0	1	90	2	3.646936	2	3	0	-0.22241	1.001604	0	0	5	1.38324	0.952467	0.791456	1	
24	1	-0.215	0	1	63	2	-1.00045	2	2	0	0.006953	-0.60508	0	3	0	-0.01305	0.137026	-0.00157	0	
25	0	-0.49052	1	1	37	0	-0.19221	2	0	2	-0.79581	-0.51274	0	1	2	-0.77103	-1.49385	-0.47738	0	
26	0	-0.34086	1	1	36	2	-0.89942	2	0	2	0.924394	-0.47581	0	0	5	-0.81092	-1.49385	-0.63599	1	
27	1	-0.41569	1	1	42	2	-0.39427	3	7	1	-0.79581	0.078221	0	0	5	0.984301	0.952467	0.474247	1	
28	1	0.57085	0	1	50	0	0.211913	3	2	0	-1.25453	-0.43887	3	1	5	-0.81092	-1.49385	-0.63599	0	
29	1	-0.8564	0	1	56	2	-0.4953	3	11	1	-0.56645	-0.60508	0	3	0	0.186423	0.137026	-0.00157	0	
30	1	-1.46329	0	2	2	2	-1.00045	3	2	1	-0.81874	-0.43887	0	0	2	-0.61145	-0.67841	-0.31878	0	
31	0	0.931228	0	1	82	0	-0.89942	3	0	1	2.071195	-0.21726	0	0	6	-0.01305	-1.49385	-0.00157	1	

Figure 2: Sample of Preprocessed Customers' Data

**Data Storage**

In this data storage phase, preprocessed data were stored in reliable storage space in comma separated value(CSV) to ensure access to data storage. In this phase, data were transferred from the data source to the data center and transfer within the data center. Apart from storage of data, data center helps in collecting, organizing and managing data. Figure 3 illustrates the block diagram showing the processes involved in the classification of bank customers demographic data.



**Figure 3:** The Block Diagram of the Classification System

**Formulation of the Hybridized Map-Reduce Back Propagation Neural Network Technique**

The weight of the BPNN was initialized and Map-Reduce Computing Technique was introduced at the second step of the BPNN to select only the useful data features and remove the useless data features from the BPNN. Map-Reduce also minimizes the weights of the data features which has influence on the complexity and speed of BPNN. It is a hybrid technique. The input data was segmented into a number of data chunks which are processed by mappers in parallel and useful data features was selected for training and classification. The detailed explanation is given as follows.

The Map-Reduce framework consists of two primary components: the Map (mapper) and Reduce (reducer) functions. The mapper performs the core data processing task, producing intermediate outputs in the form of (key, value) pairs. The reducer then aggregates these intermediate outputs from multiple mappers, performing secondary operations such as sorting and merging based on key values (Kumar et al., 2023). The final computation results are generated by the Reduce function.

In this study, the pairs-based Map-Reduce approach was employed, as outlined in Algorithm 1. Typically, document identifiers and their corresponding contents serve as the input key-value pairs. The mapper processes each document and emits intermediate pairs, where each co-occurring word pair represents the key and the integer one (the frequency count) serves as the value.

This is implemented through two nested loops: the outer loop iterates through all words (serving as the left element of the pair), while the inner loop processes the neighboring words (the right element of the pair). Word neighbors can be defined using a sliding window or other contextual boundaries such as sentences. The Map-Reduce framework ensures that all values associated with identical keys are grouped within the reducer. The reducer subsequently sums the values corresponding to each co-occurring word pair to determine their total frequency within the corpus, emitting this as the final key-value pair. Each resulting pair represents a cell in the word co-occurrence matrix. This modified algorithm demonstrates the application of complex key structures to facilitate coordinated distributed computation (Li *et al.*, 2023).

### Training and Classification using Map-Reduce Back Propagation Neural Network

This study examined a network consisting of  $n$  input Map-Reduce data variables, two hidden layers ( $k_1$  and  $k_2$ ), and  $m$  output units, as outlined in Algorithm 1. The connection weight between an input variable  $i$  and the two hidden units  $j$  is denoted as  $w^1_{ij}$  and  $w^2_{ij}$ , respectively. Similarly, the weight connecting the second hidden unit  $i$  to the output unit  $j$  is represented as  $w^3_{ij}$ . Each unit's bias ( $\theta$ ) is implemented as the weight of an additional edge. Consequently, the input Map-Reduce vectors are extended by adding a constant 1 component, which is also applied to the hidden layer's output vector. The weight connecting this constant 1 to the two hidden units  $j$  is represented by  $w^1_{(n+1,k1)}$  and  $w^2_{(k1+1,k2)}$ , while the weight connecting the constant 1 to the output unit  $j$  is denoted as  $w^3_{(k2+1,m)}$ .

A Back Propagation Neural Network (BPNN) using the stochastic gradient descent (SGD) algorithm was employed to minimize the total squared error of the network's output, starting with weight initialization. The MAPPER function from the Map-Reduce framework is executed in parallel for each pair (keyed by *custid* and customer name) within the customer dataset. This operation generates a list of pairs (keyed by instances) corresponding to each customer's neighbors. The MAPPER processes input key-value pairs (such as customer ID and name), producing zero or more output key-value pairs. Input and output types may differ. In this work, the MAPPER function performs a word count on customer fields, splitting each line into words and emitting key-value pairs like (cust w, neighbor u), where each key represents a word and the value indicates the word's frequency in the field (Lisa *et al.*, 2025).

Next, the Map-Reduce framework aggregates all pairs with identical keys (instances) and groups them accordingly. The REDUCER function is applied in parallel to each group, producing a new set of values within the same domain. Typically, each REDUCER call outputs one key-value pair or none, though multiple outputs are possible. The framework invokes the REDUCER once per unique key in sorted order, allowing it to iterate through associated values and generate outputs. In this research, the REDUCER function sums the customer values ( $c$ ) and produces a single output for each field's word and its total sum ( $s$ ). Consequently, the Map-Reduce framework converts a list of customer key-value pairs ( $(p, c_1, c_2, \dots, c_n)$ ) into another list  $((p, s))$ , which is represented as  $x_j$ .

The Map-Reduce data  $x_j$  are then loaded into the BPNN, which is randomly divided into training and testing datasets. The BPNN architecture consists of three layers—input, hidden, and output. The number of input nodes corresponds to the dimensionality of the Map-Reduced data, and the hidden layer nodes are initialized randomly. All connection weights are randomly assigned within a specified range. A Sigmoid activation function is applied for nonlinear transformation. The network is trained on the training dataset using the developed MRBPNN model until the error rate falls below a predefined threshold or the specified number of training epochs is reached. The Stochastic Gradient Descent (SGD) method updates weights after each training instance based on the computed error. Once the network achieves satisfactory accuracy, the trained MRBPNN model is used to test unseen data to classify customers as loyal or non-loyal and to assess its overall performance (Lin & Dyer, 2022). Algorithm 1 outlines the step-by-step process of the MRBPNN model.

**Algorithm 1: Hybridized Map Reduce Back Propagation Neural Network**

**Step 1:** Initialize the weight  $w_{ij}^{(1)}$  of BPNN

**Step 2:** Input Customer Data= (*custid, cust field1, ... fieldn*), target  $t_i$

**Step 3:** class MAPPER

Method MAP (*custida, custd*)

For all instances  $w \in$  cust  $d$  do

For all instances  $u \in$  NEIGHBORS ( $w$ ) do

EMIT (*pair (w, u), count l*) % Emit count for each

Co-occurrence

**Step 4:** class REDUCER

Method REDUCE (*pair p, counts [c1, c2, ...]*)

$s \leftarrow 0$

For all count  $c \in$  count[ $c_1, c_2, \dots$ ] do

$s \leftarrow s + c$

EMIT (*pair p, count s*) % Sum co-occurrence counts

$x_j =$ Map Reduce Data = EMIT (*pair p, count s*)

**Step 5:** Calculate the error from difference between the output ( $y_i$ ) and the target ( $t_i$ )

$$e_i = t_i - y_i \quad i = 1, 2, \dots, n$$

$$y_j = \varphi x_j + b$$

where  $\varphi = \frac{1}{1+e^{-x}}$  is the sigmoid transfer function,  $e_i$  is the error difference,  $x_j$  is the Map Reduce Data and  $b$  is the bias.

**Step 6:** Calculate the weights update using SGD: Stochastic Gradient Descent (SGD) is used as training function method. In this method, error is calculated for each training data and the weights are updated immediately. This is expressed in equation as follows

$$\Delta w_{ij}^m = \alpha \delta_i x_j$$

Where  $\alpha$  is the learning rate,  $m = (1), (2), (3)$  are hidden layer  $k1$  and  $k2$  and Output layer

$$\delta_i = \varphi' y_i e_i$$

where

$$\varphi' = \frac{d}{dx}(\varphi(x)) = \varphi(x) (1 - \varphi(x))$$

**Step 7:** Adjust the weight updates

$$w_{n+1,k1}^{(1)} = w_{ij}^{(1)} + \Delta w_{ij}^m$$

$$w_{k1+1,k2}^{(2)} = w_{n+1,k1}^{(1)} + \Delta w_{ij}^m$$

$$w_{k2+1,m}^{(3)} = w_{k1+1,k2}^{(2)} + \Delta w_{ij}^m$$

**Step 8:** Repeat step 2 to 4 for all the training data

**Step 9:** Repeat step 2 to 5 until the error reaches an acceptable level (epoch)

**Step 10:** Output trained network ( $trained_{net}$ )

**Step 11:** Input test data ( $test_{data}$ )

**Step 12:** Calculate Euclidean distance as follows

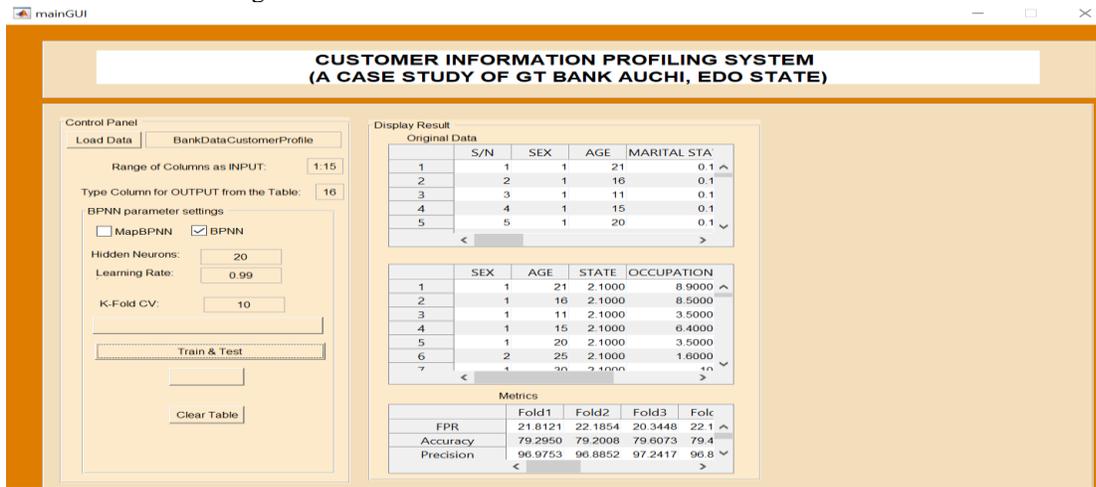
$$euc_{dist}(trained_{net}, test_{data}) = \min \sqrt{\sum_{i=1}^n (trained_{neti} - test_{datai})^2}$$

**Step 13:** If  $euc_{dist} < set\ value$  then *customer is loyal* else *customer is not loyal*

Where *set value = is the user set* value to determine loyal customer or not

### Implementation in MATLAB

The implementation of developed technique MRBPNN in the classification of the bank customer Information Profiling is done using MATLAB(R2023a). Figure 4 showed an interactive Graphic User Interface (GUI) developed with a real time database consisting of selected banks datasets in Edo state.



**Figure 4: Graphical User Interface of Training and Testing Phase**

### Performance Evaluation of Developed Technique

The performance of the hybridized Map-Reduce Back Propagation Network technique on the classification of the Bank Customer Information Profiling was evaluated based on false positive rate, recall, precision, accuracy and computation time . More so, the MRBPNN technique was compared with the standard Back Propagation algorithm. Confusion matrix was used to determine the value of the performance metrics. It contains True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN).

TP: accurately identified positive instances.

FP: mistakenly identified negative instances as positive.

TN: accurately identified negative instances.

FN: mistakenly identified positive instances as negative [21].

More so, Precision, recall, and accuracy were calculated using these terms.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Sensitivity/True Positive Rate/ Recall} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{False Positive Rate} = \frac{FP}{TN+FP} \tag{3}$$

$$F - \text{Measure} = 2 * \left( \frac{\text{Precision} * \text{Recall}}{TN+FP} \right) \tag{4}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \quad (5)$$

## RESULTS AND DISCUSSION

### Results

In this research, the performance of the developed Map-Reduce Back Propagation Neural Network (MRBPNN) and the conventional Back Propagation Neural Network (BPNN) for customer profiling classification was evaluated under varying learning rates. The simulations were performed using MATLAB R2020a on a computer system equipped with an Intel® Core™ i7 processor, running Windows 10 Professional (64-bit), with a processing speed of 3.0 GHz, 8 GB of RAM, and a 500 GB hard drive. A Graphical User Interface (GUI) was developed in MATLAB as shown in Figure 4.

The performance of both techniques was assessed across learning rate intervals of 0.00–0.49, 0.50–0.75, 0.76–0.89, and 0.90–0.99, respectively. Each range yielded similar accuracy levels for the algorithms. The learning rate, a key hyper parameter in neural network training, determines how rapidly a model adjusts to the problem. Typically, its value lies between 0.0 and 1.0. If the learning rate is set too high, the model may converge prematurely to a suboptimal solution; conversely, if too low, training may become excessively slow or stagnate. Since all ranges in this study produced comparable results, representative values of 0.49, 0.75, 0.89, and 0.99 were selected for performance evaluation as illustrated in Table 1.

The Learning Rate (LR) served as one of the primary parameters for generating the outputs of MRBPNN and BPNN during classification. It played a crucial role in training the MRBPNN to identify dataset patterns relevant to customer profiling. The dataset used for prediction was obtained from selected banks and consisted of 1,000 customer records, each containing 28 attributes and five status updates, including details such as occupation, card type, transaction amount, and deposit history. The dataset of 2,000 samples was prepared for training and validation. A 10-fold cross-validation method was employed to ensure robust model evaluation, allowing each subset of the dataset to participate in both training and testing phases. For both MRBPNN and BPNN models, 20 hidden nodes were assigned during experimentation. The average results from the 10-fold cross-validation were computed and presented in Table 1, corresponding to the four selected learning rates (0.49, 0.75, 0.89, and 0.99). The performance of MRBPNN and BPNN in customer profiling classification was assessed using standard evaluation metrics, including False Positive Rate (FPR), Accuracy, Precision, Recall, and F-measure.

### Result of MRBPNN

Table 1 defined the performance analysis of MRBPNN with learning rate of 0.49, 0.75, 0.89, and 0.99 for the Customer Profiling classification using accuracy, precision, recall, and F-measure metrics as earlier stated respectively. The average values gotten using learning rate of 0.49 for FPR, accuracy, precision, recall, F-measure and classification time were 10.83%, 89.8%, 98.67%, 89.88%, 94.07% and 12.74sec respectively. In the case of 0.75 learning rate, the average values had been 9.17%, 91.19%, 98.89%, 91.23%, 94.91% and 12.50sec of FPR, accuracy, precision, recall, F-measure and classification time, respectively. In addition, MRBPNN had 5.83% of FPR, 94.59% of accuracy, 99.31% of precision, 94.64% of recall, 96.92% of F-measure and 13.23 sec of classification time averagely, respectively. Also, at learning rate of 0.99, MRBPNN produced averagely 10.85% of FPR, 89.83% of accuracy, 98.66% of precision, 89.87% of recall, 94.09% of F-measure and 12.41sec of classification time

**Table 1: Result of MRBPNN at different learning rate**

Learning Rate	FPR (%)	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Time (s)
0.49	10.83	89.81	98.67	89.88	94.07	12.74
0.75	9.17	91.19	98.89	91.23	94.91	12.50
0.89	5.83	94.59	99.31	94.64	96.92	13.23

0.99	10.85	89.83	98.66	89.87	94.09	12.41
------	-------	-------	-------	-------	-------	-------

averagely, respectively. It was observed from the result that accuracy, precision, recall and F-measure increases from learning rate of 0.49 to 0.89 and decrease from 0.90 to 0.99. This occurred because a learning rate that is too high may cause the model to converge too quickly to a poor outcome, while a learning rate that is too low may cause the training process to stagnate or fail to progress.

### Result of BPNN

Table 2 defined the performance analysis of BPNN with learning rate of 0.49, 0.75, 0.89, and 0.99 for the Customer Profiling classification using accuracy, precision, recall, and F-measure metrics as earlier stated respectively. The average values gotten using learning rate of 0.49 for FPR, accuracy, precision, recall, F-measure and classification time were 22.10%, 79.25%, 96.90%, 79.40%, 87.28% and 27.09sec respectively. In the case of 0.75 learning rate, the average values had been 19.60%, 81.63%, 97.32%, 81.78%, 88.87% and 27.12sec of FPR, accuracy, precision, recall, F-measure and classification time, respectively. In addition, BPNN had 15.61% of FPR, 84.97% of accuracy, 97.96% of precision, 85.04% of recall, 91.05% of F-measure and 26.69 sec of classification time averagely, respectively. Also, at learning rate of 0.99, BPNN produced averagely 22.22% of FPR, 79.26% of accuracy, 96.88% of precision, 79.43% of recall, 87.29% of F-measure and 27.38sec of classification time averagely, respectively. It was noticed from the result that accuracy, precision, recall and F-measure increases from learning rate of 0.49 to 0.89 and decrease from 0.90 to 0.99. This occurred because a learning rate that is too high may cause the model to converge too quickly to a poor outcome, while a learning rate that is too low may cause the training process to stagnate or fail to progress.

**Table 2: Result of BPNN at different learning rate**

Learning Rate	FPR (%)	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Time (s)
0.49	22.10	79.25	96.90	79.40	87.28	27.09
0.75	19.60	81.63	97.32	81.78	88.87	27.12
0.89	15.61	84.97	97.96	85.04	91.05	26.69
0.99	22.22	79.26	96.88	79.43	87.29	27.38

### Comparing Result of MRBPNN and BPNN at Different Learning Rate

The result of MRBPNN was compared with the existing standard BPNN algorithm and the performance of the technique on the classification of Customer data was evaluated based on false positive rate, accuracy, precision, recall, F-measure and computational time to determine its overall performance and efficiency. Table 3 demonstrated the performance analysis of MAPBPNN at Learning Rate of 0.89 for the Customer Profiling classification using FPR, accuracy, precision, recall, F-measure and computational time metric. It could be seen from the table that MRBPNN outperformed BPNN in terms of FPR, accuracy, precision, recall, F-measure and Computational time.

### Discussion of Results

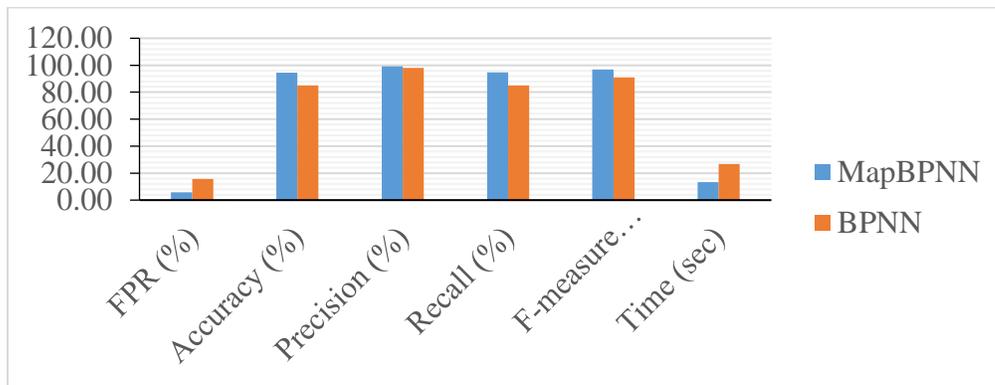
Figures 5 showed the bar chart expressing the average FPR, accuracy, precision, recall, F-measure and classification time obtained for MRBPNN and BPNN method tested on selected banks in Edo state datasets for each Learning Rate. This study considered Learning Rate of 0.81, 0.89, 0.91, and 0.95 for training and classifying Customer Profiling. The study observed that MRBPNN with Learning Rate of 0.89 has the highest Customer Profiling classification performance in all the metrics considered compared with BPNN. It was observed that Learning Rate has significant effect on the Customer Profiling classification performance of the developed MRBPNN method used.

The average accuracy, precision, recall and F-measure for the Customer Profiling classification has variable behavior for various Learning Rates with MRBPNN method. Thus, it was observed that Learning Rates affect the Customer Profiling classification performance using MRBPNN method. MRBPNN for Customer Profiling classification system

was comparatively high in terms of all the metrics considered, considering all the four-Learning Rates which justified the general high classification performance of MRBPNN.

**Table 3: Comparison results at learning rate of 0.89**

Technique	FPR (%)	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Time (s)
BPNN	15.61	84.97	97.96	85.04	91.05	26.69
MRBPNN	5.83	94.59	99.31	94.64	96.92	13.23



**Figure 5: Performance Result of MRBPNN and BPNN at Learning Rate of 0.89**

The main feature of the developed MRBPNN for the Customer Profiling classification with learning rate of 0.89 Learning Rate is that MRBPNN with learning rate of 0.89 arbitrarily assigned the kernels instead of tuning them as usually done by other Learning Rates. Compared with other Learning Rates such as Learning Rate of 0.49, 0.75, and 0.99, the developed MRBPNN with learning rate of 0.89 for the Customer Profiling classification easily completed the learning phase at very fast speed and provided more compact network when evaluated with other considered Learning Rates. Also, it achieved best generalization performance of the Customer Profiling classification when compared with other variants.

Moreover, this implies that the highest percentage of correctly classified (positive and negative) Customer Profiling classification was achieved using MRBPNN and BPNN with learning rate of 0.89. Highest accuracy (precision) was achieved because the number of correctly classified positive class was obtained when evaluated with other learning rate values. Recall metric further justified that MRBPNN and BPNN with learning rate of 0.89 network gave the highest correctly predicted positive class. Invariably, F-measure metric gave the best performance by combining unique characteristics of both precision and recall metric. In conclusion, MRBPNN with learning rate of 0.89 performed best in terms of false positive rate, accuracy, precision, recall, F-measure and processing time for Customer Profiling classification.

## Conclusion

In this research, MRBPNN and BPNN learning algorithm with different Learning Rate such as Learning Rate of 0.49, 0.75, 0.89 and 0.99 was used for Customer Profiling classification. This selected Learning Rate of MRBPNN and BPNN for Customer Profiling classification were measured in terms of accuracy, precision, recall and F-measure. The results of MRBPNN were also compared with BPNN. All the metrics used by MRBPNN effectively improve the quality and efficiency of predicting Customer Profiling. However, classification technique such as BPNN was also adopted for Customer Profiling classification in this study, but MRBPNN outperformed it and gave better results. Hence, in this study MRBPNN with learning rate of 0.89 performed excellently in predicting the customer profiling in banking system in terms of FPR, accuracy, precision, recall and F-measure compared with BPNN coupled with different Learning Rate considered in this study. MRBPNN with learning rate of 0.89 for Customer Profiling classification has better predictive performance as compared with BPNN. MRBPNN with learning rate of 0.89 can enable managers and development teams of banks to restructure customer profile application based on this highly effective and actionable classification model.

## Recommendations

In view of the results obtained from this research, the following are therefore recommended:

- i. Future work can be carried out by investigating adaptive Learning Rate in MRBPNN.
- ii. Some optimization technique can be introduced to select optimal parameters of MRBPNN in the future to further examine the performance of the system and possibly improve upon the results obtained.

## Contributions to Knowledge

This research has added to the existing body of knowledge by

- i) Introducing Map-Reduce into BPNN which gave rise to parallel processing and data locality, which reduced the computational time tremendously.
- ii) Exploring the intrinsic properties of Learning Rate in MRBPNN and BPNN for predicting accurate Customer Profiling.

## Abbreviations

AI:	Artificial Intelligence
MR:	MapReduce
MRNN:	MapReduce Neural Network
BPNN:	Back Propagation Neural Network
MRBPNN:	MapReduce Back Propagation Neural Network
FP:	False Positive
FN:	False Negative
FPR:	False Positive Rate
ML:	Machine Learning
LR:	Learning Rate
ReLU:	Rectified Linear Unit
SGD:	Stochastic Gradient Descent
TP:	True Positive
TN:	True Negative

## References

- Annamalai, M., & Muthiah, P. B. (2022). An early prediction of tumor in heart by cardiac masses classification in echocardiogram images using robust back propagation neural network classifier. *Brazilian Archives of Biology and Technology*.
- AlFayez, F., & Bouhamed, H. (2023). Machine learning and uLBP histograms for posture recognition of dependent people via Big Data Hadoop and Spark platform. *International Journal of Computers Communications & Control*, 18(1).
- Arockiam, J. M., & Pushpanathan, A. C. (2023). MapReduce-iterative support vector machine classifier: Novel fraud detection systems in healthcare insurance industry. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(1), 756–769.

- Awasthy, N., & Valivarthi, N. (2023). Evolution of Hadoop and big data trends in smart world. In *Sustainable computing: Transforming Industry 4.0 to Society 5.0* (pp. 99–127). Springer International Publishing.
- Bergui, M., Hourri, S., Najah, S., & Nikolov, N. S. (2024). Predictive modelling of MapReduce job performance in cloud environments using machine learning techniques. *Journal of Big Data*, *11*(1), Article 98.
- Biswas, P. K., & Liu, S. (2022). A hybrid recommender system for recommending smartphones to prospective customers. *Expert Systems with Applications*, *208*.
- Chandrasekar, J., & Ponnusamy, A. (2025). Hybrid jellyfish search sine cosine optimisation-based deep learning for big data classification using MapReduce framework on epileptic seizure data. *International Journal of Intelligent Information and Database Systems*, *17*(1), 1–31.
- Cui, Y., Liu, H., Wang, Q., Zheng, Z., Wang, H., Yue, Z., Ming, Z., Wen, M., Feng, L., & Yao, M. (2022). Investigation on the ignition delay prediction model of multi-component surrogates based on back propagation neural network. *Combustion and Flame*, *237*, 111852.
- Dampfhofer, M., Mesquida, T., Valentian, A., & Anghel, L. (2023). Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(9), 11906–11921.
- Dong, H., & Wang, M. (2025). Forecasting behavior and reverse engineering of self-vibrating string-mass systems utilizing a backpropagation neural network. *Journal of Vibration Engineering & Technologies*, *13*(7), 501.
- Gao, J., Wang, J., Xu, Z., Wang, C., & Yan, S. (2023). Multiaxial fatigue prediction and uncertainty quantification based on back propagation neural network and Gaussian process regression. *International Journal of Fatigue*, *168*.
- Gazis, A., & Katsiri, E. (2022). Crowd tracking and monitoring middleware via Map-Reduce. *International Journal of Parallel, Emergent and Distributed Systems*, *37*(3), 333–343.
- Gattoju, S., & Nagalakshmi, V. (2023). Design of ChaApache framework for securing Hadoop application in big data. *Multimedia Tools and Applications*, *82*(10), 15247–15269.
- Hamdi, M., Bouhamed, H., Badreddine, F., & Alkanhel, R. I. (2024). Deep recurrent neural networks distributed on a Hadoop/Spark cluster for fall detection. *International Journal of Computers Communications & Control*, *19*(3).
- Hosseini, S., Poormirzaee, R., & Hajihassani, M. (2022). Application of reliability-based back-propagation causality-weighted neural networks to estimate air-overpressure due to mine blasting. *Engineering Applications of Artificial Intelligence*.
- Jansen, B. J., Aldous, K. K., Salminen, J., Almerakhi, H., & Jung, S. G. (2024). Data preprocessing. In *Understanding audiences, customers, and users via analytics: An introduction to the employment of web, social, and other types of digital people data* (pp. 65–75). Springer Nature Switzerland.
- Jonathan, L. E., Winston, A. G., & Chukwuemeka, P. (2025). Machine learning and morphometric analysis for runoff dynamics: Enhancing flood management and catchment prioritization in Bayelsa, Nigeria. *Journal of Computational Systems and Applications*.
- Kumar, K. S., Manoj, H. K., & Geetha, D. E. (2023). Twitter data analysis using Hadoop and R and emotional analysis using optimized SVNN. *Computer Systems Science & Engineering*, *44*(1).
- Li, S., Wang, W., Lu, B., Du, X., Dong, M., Zhang, T., & Bai, Z. (2023). Long-term structural health monitoring for bridge based on back propagation neural network and long and short-term memory. *Structural Health Monitoring*, *22*(4), 2325–2345.
- Lin, J., & Dyer, C. (2022). *Data-intensive text processing with MapReduce*. Springer Nature.