CAT-ANT SWARM OPTIMIZATION BASED ON REPETITIVE DEEP LEARNING NEURAL NETWORK FOR BIG DATA PROCESSING

Noor Basha[1]
Assistant Professor, Dept. Of CSE, Vemana I T, Bengaluru – 560034, Karnataka, India, md.noor202@gmail.com
Dr. Ashok Kumar P S[2]
Professor, Dept. Of CSE, DBIT, Bengaluru– 560074, Karnataka, India,ashokdbit2017@gmail.com
Shilpa Reddy K[3]
Assistant Professor, Dept. Of CSE, Vemana I T, Bengaluru – 560034, Karnataka, India,shilpa.reddy@vemanait.edu.in
RoopaLakshmi S[4]
Assistant Professor, Dept. Of CSE, Vemana I T, Bengaluru – 560034, Karnataka, India,rdruvaraj94@gmail.com

Abstract:

Due to technology upgradation huge data is accumulating in the server from various devices very rapidly, even though acquired data to be organized in an effective manner is a challenging task. Withrespect to the data classification machine learning approach is very efficient.This paper presents a technique for extensive data processing, which utilizes a repetitive deep learning neural network, based on cat-ant swarm optimization approach. This algorithm integrates the ant-lion optimization and cat-swarm optimization techniques.

The proposed methodology in the paper encompasses the map-reduce programming technique in Big-data framework, is to classify the input data. The Pearson correlation method is utilized in the feature selection process, where, outcome of the suggested approach is derived from the comparison of the established classification techniques, i.e. the outcome is derived from Binary Particle Swarm Optimization algorithm, Whale Optimization algorithm, and Wolf-based Correlative Naive Bayes classifier.

Cat-Swarm optimization approach derived from neural network, which employs iterative deep learning techniques, it leads to enhance the outcomes of the system. While evaluating the attack detection systems, it assess several parameters, but they are very decisive in the comparison of such systems, like, precision rate, recall rate,attack detection rate, accuracy in attack detection and operational time. Finally result set have been tabulated and depicted using graphical representations.

Index Terms: Big data, cat-ant swarm, ant lion, optimization, map-reduce, person correlation, deep learning

1. Introduction

An adversary could potentially introduce malicious profiles into the collaborative recommendation system, but it artificially inflate their products' ratings in the market, thisevent can be traced by the recommendation system, it innate frailties and openness, even though this activity are to be blamed[1]. Various methods are available for detecting this kind of attack, and each of them uses a unique piece of information gleaned from the users' profiles. The objective of improving the precision regarding on attack detection is notyet accomplishedcompletely, right now various new techniques have been emerged in these years, so that it is very easy to differentiate between malicious attacks and fake users in the decisive systems.

Despite with these advancements, a considerable amount of time is still required in order to detect attacks within recommendation systems[2]. The feature extraction method is a very important approach, it is very cable to identify any type attacks, i.e., based on the relevant user characteristics,Feature extraction approach is identify the order to make the process of detecting attacks easily. The AdaBoost Incremental Partitioning around Medoid Clustering (GAIP-AMC) method was developed, in order to identify the profile injection attackinstances within the collaborative recommendation system[3].

MEMF-GEBSVC is a method developed to detect profile injection attacks in collaborative recommendation system. This method uses Multivariate Empirical Mode Decomposition and Gradient Support Vector Entropy Boosting Classifier. MEMF-GEBSVC is designed to detect profile injection attacks with fast and accurate, where MEMF-GEBSVC approach analyzes the MovieLens 1M dataset[4]. MovieLens 1M dataset finds the user ratings of many films. MEMF-GEBSVC involves two steps.

- Multivariate Empirical Mode Decomposition (MEMF) extracts data features, where IMFs can be extracted during the process. After selecting the features, categorize them based on user’s profile to check for legitimacy or attack.
- GEBSVC detects profile injection attacks, where weak support vector entropy classifier naming on user profiles, where each weak learner output has a loss function; it alters the weak classifier weights. The classifier with the lowest loss among less capable learners is the most reliable.

A robust classifier can be used to find the genuine user profiles from malicious one, it allows collaborative recommendation systems to detect profile injection attacks[5].

Big data has a massive volume of data, its main characteristics are identified with “4 V representation", i.e., variety, volume, veracity, and velocity, so big data is highly distinguished from other frame works. Big data's biggest challenge is diversity; this approach includes text, images, and video [6]. Big data is defined by its size, type, and data generation rate. Data parallelism and distributed processing are two key strategies to managing large amounts of data; it divides a large database into n number of subsets and processes each subset using one of the n methods. The subsequent procedure addresses the challenges of extensive data sets by utilizing the MapReduce programming framework. MapReduce is an algorithm and framework developed by Google[7].

While combining Map Reduce algorithm and distributed file system it make, much more simpler to manage the processing of large amounts of data across a network of systems. In data mining, the current methodology is regarded as an additional parallelization method, specifically for Message Passing Interface (MPI), because of its fault-tolerant mechanism in the network, i.e., MPItypically takes a significant amount of accessible time for the completion of wide variety of tasks.

In most of the cases, Map Reduce framework is implemented with the assistance of Hadoop, it is a well-known framework for parallel programming[8]. Parallel computing is one of the most effective method for managing large datasets, whereit aims to partition large problems into smallermanageable pieces, so that multiple processors can work simultaneously on piece of data. So parallel and distributed data processing allows, other operations to be carried out simultaneously with in the frame work. Deep learning strategies shine when applied to the problem of dealing with large amounts of unsupervised data because of their ability to acquire data representation incrementally and greedily [9]. A deep learning approach provides a highly effective solution for addressing issues associated with large amounts of data, it efficiently extract valuable insights from massive databases while minimizing the computational resources required. Before the development of deep learning methods, few choices were available for the feature selection and learning process. In recent years, there has been an explosion of new computational capabilities across a wide range of fields due to the rapid development of computer hardware and the introduction of the general processing unit and deep learning computing.

OSN provides the ideal framework for applying deep learning techniques to large-scale databases and their use in the AI and entertainment industries[10]. Deep learning approach is a powerful tool for big data analytics because of its ability to shift through large amounts of data, particularly unsupervised data, and extract complex abstractions using advanced and sophisticated techniques in the representation of big data analytics. Deep learning is being utilized to find solutions to a variety of big data issues, including data tagging, discriminative modeling, semantic indexing, and quick information retrieval, these issues are pertaining to big data.

It has been demonstrated that conventional approaches to machine learning and feature engineering are insufficient to extract nonlinear and complex patterns from large amounts of data. Deep learning makes it possible to use simpler linear methods for data analysis tasks involving large amounts of data, such as prediction and classification techniques by extracting a variety of features [11].

Due to the development of big data frame work, it manages massive amounts of data in significant advancement, even though, big data methodology has variety of shortcomings, such as low levels of accuracy, temporal complexities, and high-dimensional issues, but a novel strategy hasbeen proposed a potential solution to increasedthe precision, sensitivity, and threat score. This research paper presents a novel deep learning classifier that is optimization-driven and is implemented using a Map-Reduce framework[12]. The hope is that this classifier will make it easier to perform classification tasks on massive datasets. During the execution process, the modelcomprise of two stages, i.e., reducerphase and mapper phase. The Ant-Cat Swarm Optimization technique is enabled with Deep Recurrent Neural Network (ACSO-based Deep RNN) classifier; it classifies large amounts of data in Map-Reduce approach. Pearson correlationbased BHEFC approach is used to select features from data from multiple distributed sources in the mapper stage and reducer phase will classify data using the given filtered characteristics.

ACSO approach is a hybrid of Ant-Lion Optimisation (ALO) and Cat Swarm Optimisation (CSO) technique to train the reducer's Deep Recurrent Neural Network (RNN), i.e., it uses ACSO-based Deep RNNs and MapReduce. A Deep Recurrent Neural Network classifies features selected in the mapper phase in the reducer phase. Advanced Cuckoo Search Optimisation trains this network[13]. The newly derived ACSO algorithm combines the ALO and CSO methods for fine-tuning the network's weights and biases when it is applied to Deep RNN. This allows the algorithm to produce more accurate results. While execution of this work, a Deep RNN model founded on ACSO was utilized to enhance the effectiveness with which the system classified vast amounts of data.

2. Background Study

Classifying massive datasets now requires a wide variety of methods, however, it is very tedious, while acquiring from different things. The effectiveness of the proposed method is shown by comparing it to several other methods in the same domain currently considered to be state-of-the-art[14]. In this paper, the authors over the eight common methods for organizing big data based on the various ways it can be broken down and the difficulties inherent to each other. The authors, Dabbu et al., created both the MLNN and the LMNN, or the Linear-Morphological Neural Network. These networks use a combination of morphological and perceptron neurons to classify massive datasets. The MLNN comprises a hidden layer of morphological neurons and a typical perceptron output layer[15]. These parts of the system are in charge of finding interesting patterns in the data. In the second design, the LMNN was utilized as the feature extractor. The morphological neurons' output layer was subsequently added to aid in nonlinear classification. To achieve the desired result, two distinct architectures were trained using stochastic gradient descent. While this may seem like a good way to determine morphological weights, it's important to remember that other clever methods based on the geometrical features of the training data were not considered is a paramount significance. Tair et al. [16] put together groups from large datasets, where "Whale Optimisation Algorithm" or "WOA" was first made to find the best features for improving classification accuracy. During the pre-processing phase, LSH-SMOTE and SMOTE approaches are used to ensure each group's sample with same number of members. At the end of the article, result expresses how classification could be done with WOA and a bidirectional recurrent neural network. The method fixed the problem of unequal classes, but it was hard for Spark or Hadoop to process large databases quickly.

Abdalla et al. suggested using a Map-Reduce model (MRM) with a Cuckoo-Grey Wolf-based Correlative Naive Bayes classifier (CG-CNB)[17]. The CG-CNB algorithm is the combination of CGWO and CNB algorithms. The best parts of the Grey Wolf Optimizer (CGWO)and the Cuckoo Search have been put into this new CG-CNB algorithm. The CNB model will be best model, if constraints are chosen. CG-CNB-MRM put samples into groups using probabilities and an index table for probabilities.

Narayana et al. used Apache Spark and a deep learning model to sort huge datasets in 2022[13]. This research adds to the Rider Chaotic Biography by using a mix of the Rider Optimisation Algorithm and Chaotic Biogeology-based Optimisation to sort data into categories, but this method will not provide the best accuracy for classifying of data. Abolarinwa et al. used the CSO scheme to categorize massive datasets. CSO (ICSO) is aimproved classification algorithm and it is developed by enhancing the CSO algorithm for use with large datasets[19]. It will create less difficult to zero in on the set of characteristics that would allow for the most precise categorization. The study found that, while selecting the same number of features, or more, it will increasethe ICSO classification accuracy, however, a few extra steps need to be considered to improve the standard CSO tracing method.

Aghaeipoor et al. [20] created the CHI-PG method, which is a part of the Map Reduce Prototype Generation (PG) technique. This technique has linear time complexity and guarantees accuracy regardless of the number of concurrent operations. In addition, the prototypes were presented in a simplified system that was developed using a rule-generation procedure. The study concludes that the k-nearest neighbor (k-NN) method is the most effective for data reduction and classification[21]. However, in the search for answers to issues brought on by high-dimensional problems, other techniques that could help reduce the total number of features were not considered. The method of categorization known as Topology-controlled Scale-Free Binary Particle Swarm Optimisation (PSO) was developed by Prabakaran et al. [22]. A multi-objective fitness function was used to reduce the number of features and increase classification precision. However, nobody has investigated how different particle learning schemes have influenced the system's performance. Shen et al. developed a platform for collaborative learning called Compact Fuzzy Models in Big Data Classification (CFM-BD)[23]. The system intends to generate understandable and precise classification schemes for massive data sets. The current strategy was built from the ground up to address issues when processing massive amounts of data[24]. Pre-processing, rule induction, and rule selection were identified as the three main phases of the learning process. During the initialization process, we used the probability integral transform theorem[25]. Rule induction was then successfully completed using the Apriori and CHI-BD algorithms. Global evolutionary optimization techniques were ultimately applied to rule selection. However, the multi-objective framework developed to enhance classification did not contain the method for selecting evolutionary rules. In their study, Ashraf and colleagues showed how a deep convolutional neural network can be used to process and categorize large amounts of medical image data[26]. While the just-described technique is promising, it requires further investigation in the context of large-scale image datasets before it can contribute to advances in image classification. The authors developed a Structural-Labelled Locally Deep Nonlinear Embedding to classify massive datasets. The strategy worked. Efficiency gains are being seen throughout the computing process. However, some structural details must be considered if accurate vector representations are to be obtained with sparse input[27]. Ashok Kumar P S.et al. demonstrated on heart disease prediction classification using an ML approach in a big data frame work. We've indeed come a long way in terms of efficiency, but we can do better overall performance,to do calculations we have to cut down on the time [18].

3. Methodology

Big data analysis helps to analyze huge amounts of data, which have been obtained from different sources. Feature selection techniques have been used in this proposed framework, to manage the computation time and distributed data, meanwhile big data classification has been done using the Map Reduce framework, where mapper and the reducerfunction will perform different functions, it classify the features of the data, in order to produce the eventual output. The architecture of the proposed methodologyshown in Figure 1.

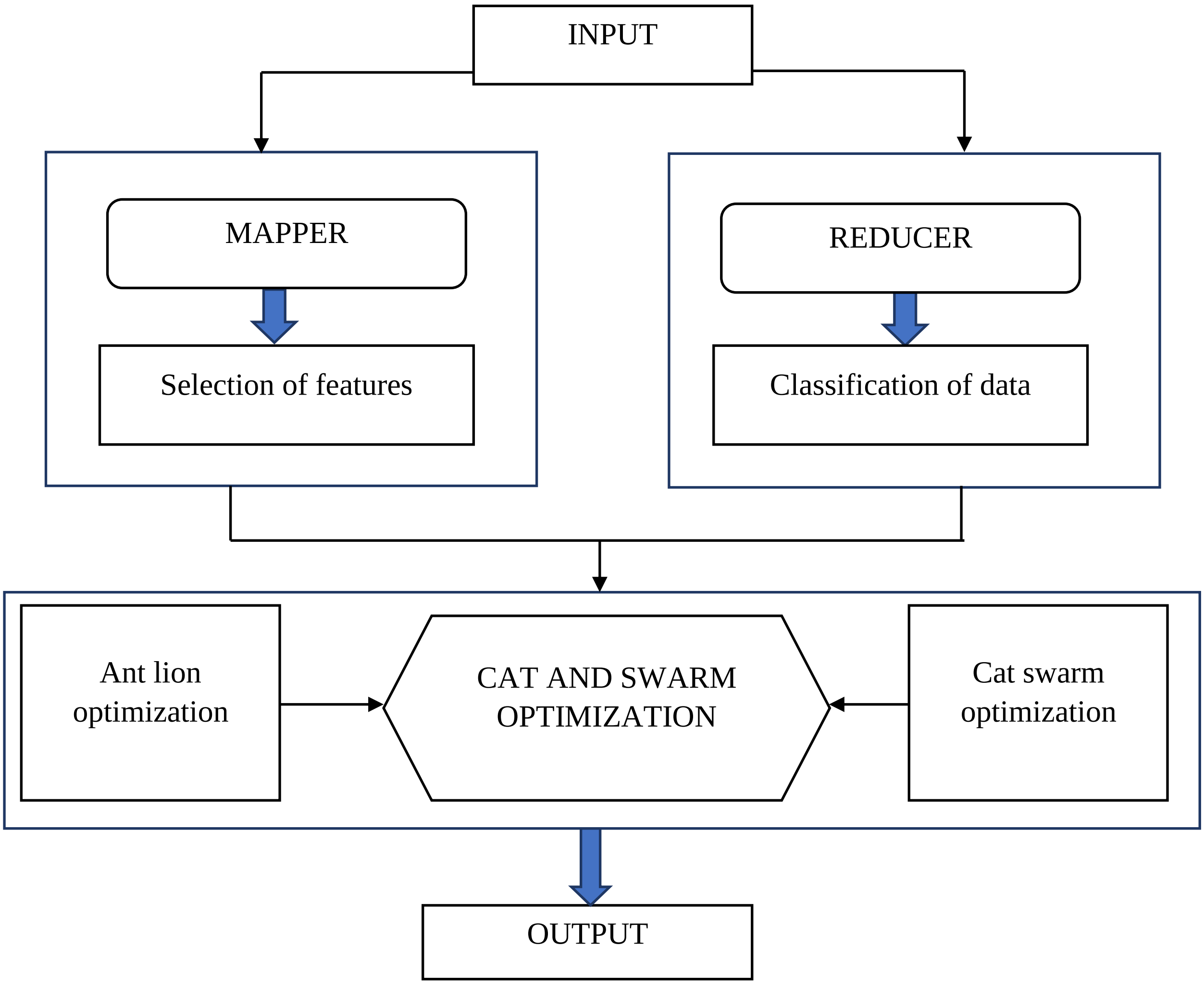

Figure 1: Architecture of the proposed work.

The mapper is responsible for selecting features from the given input data, and thenreducerfunction will help to classify it. The cat-ant swarm optimization combines the ant-lion optimization and the cat swarm optimization techniques.

1.1. Acquisition of data:

The map-reduce is a software platform that can be used to compute vast amounts of data. The computation is done using the mapper function and the reducer function. The mapper segments the input data into multiple files. Many mapper functions will work on these files.

$$\boldsymbol{g = d_{rs}; (1 \leq R \leq p); (1 \leq S \leq q)} \quad - \quad \mathbf{1}$$

$$d_{rs} = \{K_E\}; (1 \leq E \leq r) \quad \mathbf{-2}$$

Let g denote the data in the big data, which contains several attributes. R indicates the R^{th} data, and S represents the S^{th} attribute. The total count of all the attributes is denoted by q, and the total count of the data present is denoted by p.

1.2. Selection of the features based on the mapper:

The acquired data is given to the mapper function, in which the feature selection process is handeled using the pearson based correlation approach, it increases the accuracy of the data classification. This correlation has three different components to accomplishment of data integration. They are the entropy of the black hole, Bayeson interference, and the clustering through fuzzy.

$$\vec{p}(a_i, r) = p(a_i, r).p(a_i)\beta \prod_{C=1}^{S} exp \frac{1}{2}\left\{a_i \left[\frac{x(CD)}{\sigma_C \sigma_D}\right]\right\} {a_i}^{-L} \mathbf{- 3}$$

$$\vec{p}(z, y_i) = p(z, y_i).p(y_i)\beta\, exp\left\{-\frac{1}{2}{a_i}^{L}\left[\frac{x(CD)}{\sigma_C \sigma_D}\right]\right\} {a_i}^{-L} \times exp\left\{\gamma + \phi \sum^{\Sigma\left[\frac{x(CD)}{\sigma_C \sigma_D}\right]} ln\, \{\}_i^{-L}\right\} \mathbf{- 4}$$

The above equation gives the fuzzy c means clustering of the Pearson correlation based on the Legrangian principle.

1.3. Reducer stage:

After selecting the features through the Pearson correlation model, the concatenated features can be represented by the following equation.

$$b = \{h_1||h_2, h_i||h_r\} \quad - \quad \mathbf{5}$$

The classification can be done using the proposed deep learning recurring network method. The optimal weights of the classification can be obtained using the proposed method.

1.4. Algorithm:

Input: population of the cat swarm

Output: optimum solution

Steps:

Start

Initialization of the population
Read the parametric values
While G< maximum generation
Calculate the fitness value of each of the cats
Order the position of the cats based on the calculated value
For U==1:
If SCP==1
Trace the manner of the seeking
Else
Update the trace equation
Update the ant-lion equation
Update the ant cat optimization equation
End if
End for
Check the attainability of the solutions
Return the optimum solution
G=G+1
End while
End if
End for
Attainment of the best solution
Stop

Figure 2 shows the block diagram of a repetitive deep-learning neural network, where redundant neural networks are a particular type of artificial intelligence-based network, it is also called as recurrent neural networks, it is capable of processing sequential data. Normal neural networks process the data from a single input and produce a single output.

For repetitive neural networks, model have several connections, it form a loop, this loop allows the redundant neural networks to process the data recurrently to turn, makes it suitable for processing sequential data.

One of the outstanding features of the repetitive neural network is that they have many hidden layers capable of capturing the information from past inputs sequentially. As this process continues, the hidden layers identify a pattern over time.

The hidden layers consist of a set of neurons that are interconnected. These neurons are also connected to the input and output layers. The input will be combined with the hidden past state to form a novel hidden state. The production of the output follows this. The task complexity and the number of data available to be processed determine the number of hidden layers in a repetitive neural network. A repetitive neural network with many hidden layers can process the data effectively. However, the drawback is that there would be more complexity and resource requirements. The hidden layers' activation function also plays a significant part in data processing.

The cat-ant swarm algorithm dataset training is based on a repetitive deep-learning neural network. The determination of the optimized weight value is the critical factor for training the repetitive deep-learning neural network. The proposed cat-ant optimization is based on the ant-lion optimization and the cat-swarm optimization techniques. The ant-lion optimization is based on the interactions between the ant lions and the ants at the corner positions. The ants are assumed to shift in the free space.

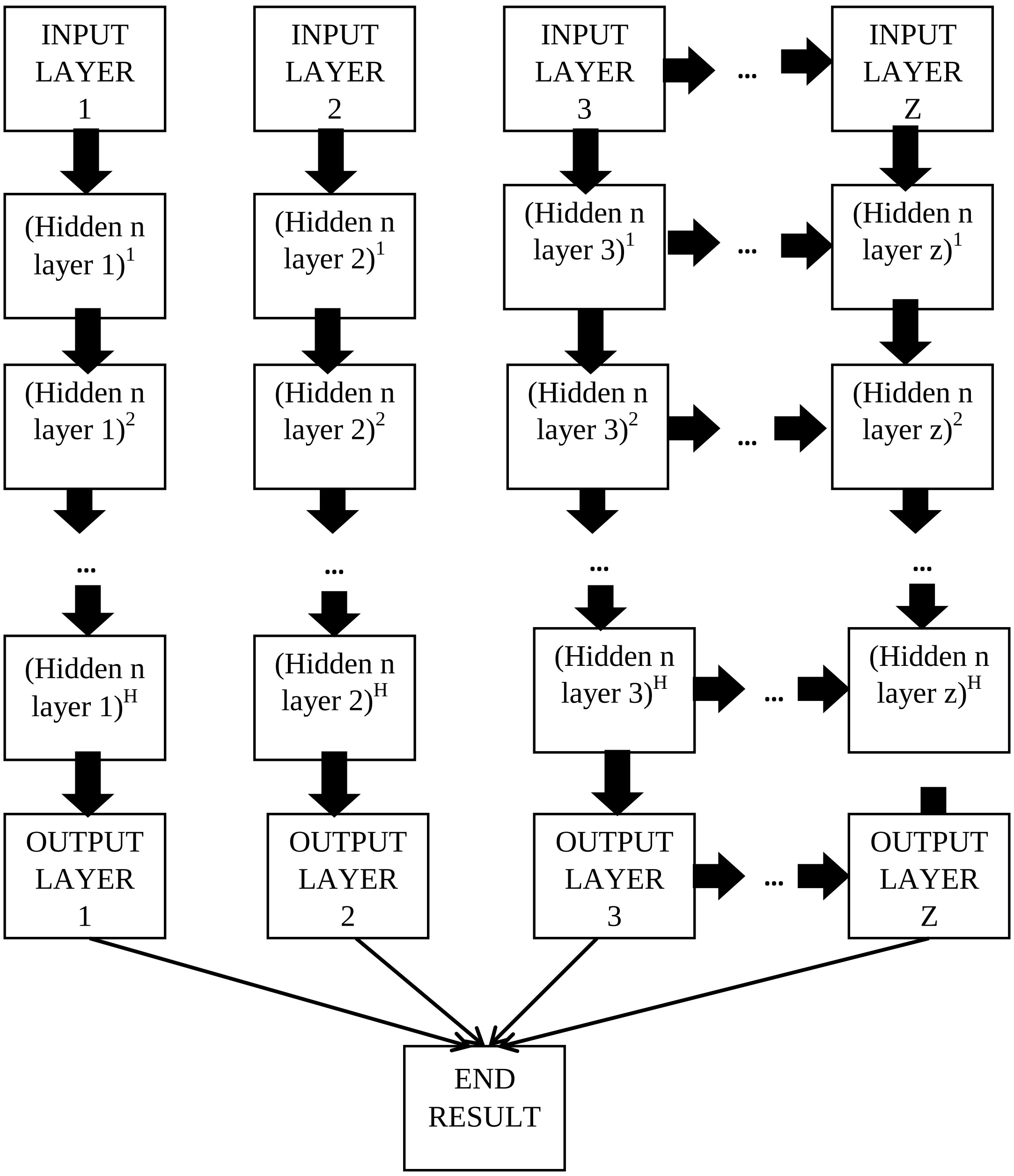

Figure 2: Block diagram of repetitive deep learning neural network

The ant lions become fit by hunting the ants. In the cat swarm optimization, the cats' distribution is random. The population of cats is categorized into two modes. They are the seeking phase and the tracing phase. The seeking mode denotes the observing power of the cats, and the tracing phase represents the movement of the cats in search of prey.

$$X_{U,G,NEW} = X_{U,G,OLD} + P_{U,G} \quad - \quad \mathbf{6}$$

$$P_{U,G,NEW} = p_{U,G} + L_1 M_1 (X_{best} - X_{U,G} \quad - \quad \mathbf{7}$$

$$X_{U,G,NEW} = X_{U,G,OLD} + P_{U,G,NEW} - L_1 M_1 (X_{best} - X_{U,G}) \quad - \quad \mathbf{8}$$

$$X_{U,G,NEW} = X_{U,G,OLD} + P_{U,G,NEW} - L_1 M_1 X_{best} - L_1 M_1 X_{U,G} \quad - \quad \mathbf{9}$$

$$X_{U,G,NEW} = X_{U,G,OLD}(1 + L_1 M_1) + P_{U,G,NEW} - L_1 M_1 X_{best\ .G} \quad - \quad \mathbf{10}$$

When the cat observes some risk during the resting process, it shifts its position slowly. The first equation gives the standardized equation for the tracing phase of the cat. The final equation denotes the updated position of the cat after its movement.

4. Data Analysis and Findings

Table 1 shows the rate of attack detection in percentage for the various types of algorithms. Figure 3 represents the graphical representation of the proposed algorithm's rate of attack detection compared to the existing algorithm.

Table 1: Rate of detection of attack in percentage

User Count	Binary particle swarm optimization	Whale optimization algorithm	Wolf-based Correlative Naive Bayes classifier	Cat-ant swarm optimization
500	78	82	87	91
1000	77	81	86	92
1500	76	86	85	94
2000	78	84	88	93
2500	79	85	89	96
3000	77	83	86	95
3500	75	82	84	92
4000	78	81	87	93
4500	76	82	83	95
5000	79	84	90	94

The user counts are considered from 500 to 5000. The binary particle swarm optimization has given an attack detection rate of 78, 77, and 76 percent when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the percentage of attack detection rate of the binary particle swarm optimization is 78, 79, and 77, respectively. About 75, 78, 76, and 79 percent of attack detection rates have been attained through binary particle swarm optimization when the number of users is 3500, 4000, 4500, and 5000, respectively.

The user counts are considered from 500 to 5000. The Whale optimization algorithm has given an attack detection rate of 82, 81, and 86 percent when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the percentage of attack detection rate of the Whale optimization algorithm is 84, 85, and 83, respectively. About 82, 81, 82, and 84 percent of attack detection rates have been attained through the Whale optimization algorithm when the number of users is 3500, 4000, 4500, and 5000, respectively.

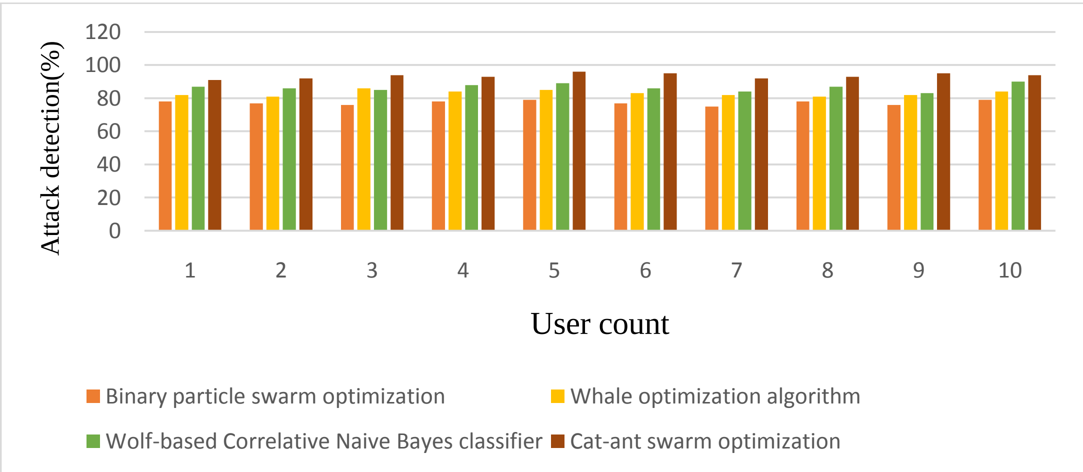

Figure 3: Graphical representation of the rate of detection of attack in percentage

The percentages of attack detection rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 500, 1000, and 1500 are 87, 86, and 85 respectively. Similarly, the percentages of attack detection rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 2000, 2500, and 3000 are 88, 89, and 86, respectively. Using the Wolf-based Correlative Naive Bayes classifier, the attack detection rate percentage for the user counts of 3500, 4000, 4500, and 5000, 84, 87, 83, and 90, respectively.

The percentages of attack detection rate obtained through proposed cat-ant swarm optimization for the user counts of 500, 1000, and 1500 are 91, 92, and 94, respectively. Similarly, the percentages of attack detection rate obtained through the proposed cat-ant swarm optimization for the user counts of 2000, 2500, and 3000 are 93, 96, and 95, respectively. Using the proposed cat-ant swarm optimization, for the user counts of 3500, 4000, 4500, and 500, the attack detection rate percentage is 92, 93, 95, and 94, respectively.

Table 2 gives the accuracy of attack detection of different algorithms in percentage, and Figure 4 depicts the graphical representation of the accuracy of attack detection of various algorithms compared to the proposed algorithm.

Table 2: Accuracy of attack detection different algorithms in percentage

User Count	Binary particle swarm optimization	Whale optimization algorithm	Wolf-based Correlative Naive Bayes classifier	Cat-ant swarm optimization
500	75	81	85	92
1000	74	86	87	95
1500	76	84	86	97
2000	79	82	89	96
2500	81	83	87	97
3000	76	84	88	95
3500	83	81	87	92
4000	81	82	86	94
4500	74	83	85	93
5000	82	85	86	94

The binary particle swarm optimization has given attack detection accuracy rates of 75, 74, and 76 percent when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the percentage of accuracy of the binary particle swarm optimization attack detection is 79, 81, and 76 respectively. About 83, 81, 74, and 82 percent of attack detection accuracy have been attained through binary particle swarm optimization when the number of users is 3500, 4000, 4500, and 5000, respectively.

The Whale optimization algorithm has given the accuracy of attack detection of 81, 86, and 84 percentages when the user count is 500, 1000, and 1500, respectively. When the user count is 2000, 2500, and 3000, the percentage of accuracy of attack detection of the Whale optimization algorithm is 82, 83, and 84, respectively. About 81, 82, 83, and 85 percent of the accuracy of attack detection have been attained through the Whale optimization algorithm when the number of users is 3500, 4000, 4500, and 5000, respectively.

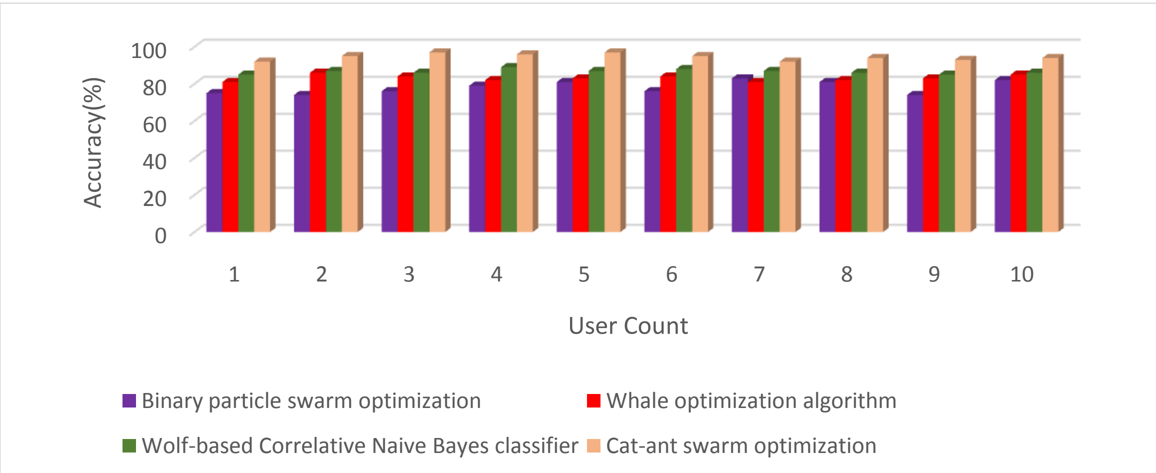

Figure 4: Graphical representation of the accuracy of attack detection different algorithms in percentage

The percentages of the accuracy of attack detection rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 500, 1000, and 1500 are 85, 87, and 86, respectively. Similarly, the percentages of the accuracy of attack detection rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 2000, 2500, and 3000 are 89, 87, and 88, respectively. Using the Wolf-based Correlative Naive Bayes classifier for the user counts of 3500, 4000, 4500, and 500, the attack detection rate accuracy is 87, 86, 85, and 86, respectively.

The percentages of the attack detection rate accuracy obtained through the proposed cat-ant swarm optimization for the user counts of 500, 1000, and 1500 are 92, 95, and 97, respectively. Similarly, the accuracy percentages of the attack detection rate obtained through the proposed cat-ant swarm optimization for the user counts of 2000, 2500, and 3000 are 96, 97, and 95, respectively. For the user counts of 3500, 4000, 4500, and 500, the accuracy of attack detection rate percentage is 92, 94, 93, and 94, respectively, using the proposed cat-ant swarm optimization.

Table 3 gives the rate of precision in percentage. Figure 5 gives the graphical representation of the precision rate in percentage value.

Table 3: Rate of precision in percentage

User Count	Binary particle swarm optimization	Whale optimization algorithm	Wolf-based Correlative Naive Bayes classifier	Cat-ant swarm optimization
500	75.21	78.52	85.92	93.51
1000	76.24	76.51	86.24	95.24
1500	74.32	78.91	86.54	94.62
2000	76.42	76.54	89.42	93.57
2500	74.14	77.64	87.62	94.61
3000	75.16	79.61	89.24	96.52
3500	76.24	78.51	88.67	92.35
4000	77.24	76.34	89.38	95.42
4500	76.21	77.95	88.64	96.24
5000	77.51	78.62	87.62	93.51

The binary particle swarm optimization has a precision percentage rate of 75.21, 76.24, and 74.32 when the user count is 500, 1000, and 1500, respectively. When the user count is 2000, 2500, and 3000, the precision percentage rate of the binary particle swarm optimization is 76.42, 74.14, and 75.16, respectively. About 76.24, 77.24, 76.21, and 77.51 precision percentage rates were attained through binary particle swarm optimization when the number of users is 3500, 4000, 4500, and 5000, respectively.

The Whale optimization algorithm has given a precision percentage rate of 78.52, 76.51, and 78.91 percentages when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the precision percentage rate of the Whale optimization algorithm is 76.54, 77.64, and 79.61, respectively. About 78.51, 76.34, 77.95, and 78.62 precision percentage rates were attained through the Whale optimization algorithm when the number of users is 3500, 4000, 4500, and 5000, respectively.

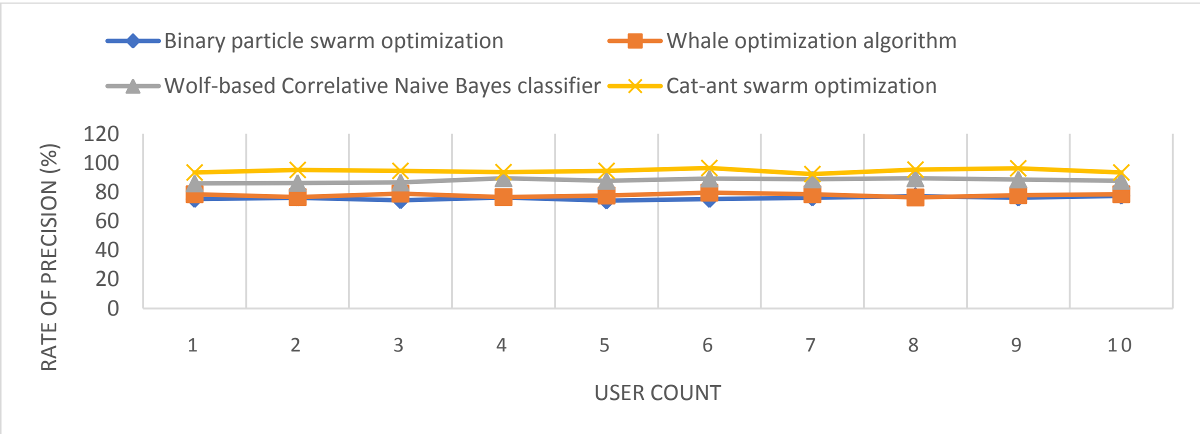

Figure 5: Graphical representation of the rate of precision in percentage

The precision percentage rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 500, 1000, and 1500 is 85.92, 86.24, and 86.54, respectively. Similarly, the precision percentage rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 2000, 2500, and 3000 is 89.42, 87.62, and 89.24, respectively. For the user counts of 3500, 4000, 4500, and 500, the precision percentage rate percentage is 88.67, 89.38, 88.64, and 87.62 respectively by using the Wolf-based Correlative Naive Bayes classifier.

The precision percentage rate obtained through the proposed cat-ant swarm optimization for the user counts of 500, 1000, and 1500 is 93.51, 95.24, and 94.62, respectively. Similarly, the precision percentage rate obtained through the proposed cat-ant swarm optimization for the user counts of 2000, 2500, and 3000 is 93.57, 94.61, and 96.52, respectively. For the user counts of 3500, 4000, 4500, and 500, the precision percentage rate is 92.35, 95.42, 96.24, and 93.51, respectively, using the proposed cat-ant swarm optimization.

Table 4 gives the percentage values of the rate of recall of various algorithms, and Figure 6 gives the graphical presentation of the percentage values of the recall rate of different algorithms and the proposed algorithm.

Table 4: Rate of recall in percentage

User Count	Binary particle swarm optimization	Whale optimization algorithm	Wolf-based Correlative Naive Bayes classifier	Cat-ant swarm optimization
500	77.52	78.12	86.12	92.47
1000	76.31	77.32	85.12	93.36
1500	75.52	76.52	84.54	95.68
2000	73.63	75.40	86.57	96.24
2500	72.54	74.24	87.87	93.43
3000	71.82	77.35	86.63	95.86
3500	76.84	78.62	85.32	94.57
4000	74.42	78.43	86.22	93.36
4500	75.32	76.65	87.75	92.54
5000	77.11	79.87	88.92	91.21

The binary particle swarm optimization has given a recall rate of 77.52, 76.31, and 75.52 percentages when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the recall rate of the binary particle swarm optimization is 73.63, 72.54, and 71.82, respectively. About 76.84, 74.42, 75.32, and 77.11 recall rates were attained through binary particle swarm optimization when the number of users is 3500, 4000, 4500, and 5000, respectively.

The Whale optimization algorithm has given recall rates of 78.12, 77.32, and 76.52 percentages when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the recall rate of the Whale optimization algorithm is 75.40, 74.24, and 77.35, respectively. About 78.62, 78.43, 76.65, and 79.87 recall rates were attained through the Whale optimization algorithm when the number of users is 3500, 4000, 4500, and 5000, respectively.

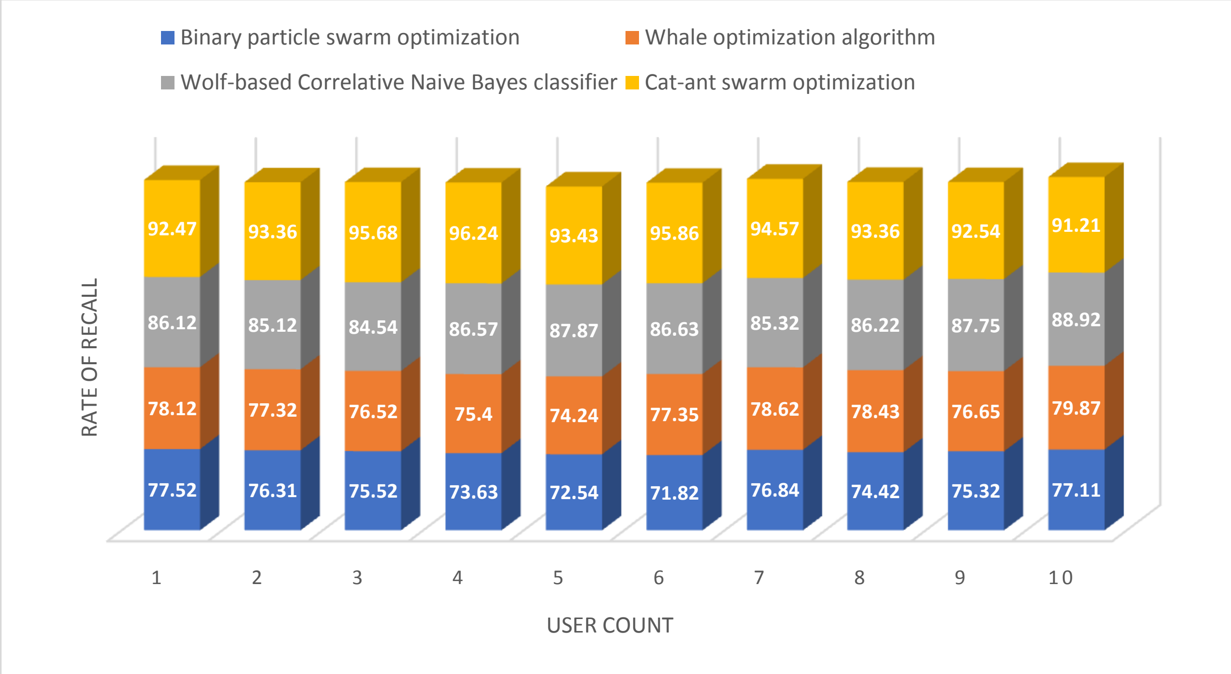

Figure 6: Graphical representation of the rate of recall in percentage

The recall rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 500, 1000, and 1500 is 86.12, 85.12, and 84.54, respectively. Similarly, the recall rate obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 2000, 2500, and 3000 is 86.57, 87.87, and 86.63, respectively. For the user counts of 3500, 4000, 4500, and 500, the recall rate percentage is 85.32, 86.22, 87.75, and 88.92 respectively by using the Wolf-based Correlative Naive Bayes classifier.

The recall rate obtained through the proposed cat-ant swarm optimization for the user counts of 500, 1000, and 1500 is 92.47, 93.36, and 95.68, respectively. Similarly, the recall rate obtained through the proposed cat-ant swarm optimization for the user counts of 2000, 2500, and 3000 is 96.24, 93.43, and 95.86, respectively. For the user counts of 3500, 4000, 4500, and 500, the recall rate is 94.57, 93.36, 92.54, and 91.21, respectively, using the proposed cat-ant swarm optimization.

Table 5 gives the time taken for execution in percentage, and Figure 7 gives the graphical representation of the time taken for execution in percentage.

Table 5: Time taken for execution in percentage

User Count	Binary particle swarm optimization	Whale optimization algorithm	Wolf-based Correlative Naive Bayes classifier	Cat-ant swarm optimization
500	51	41	35	30
1000	55	43	37	32
1500	57	46	41	34
2000	61	52	43	37
2500	63	54	47	38
3000	65	57	49	43
3500	68	59	51	45
4000	72	62	54	47
4500	74	63	57	50
5000	75	64	59	51

The binary particle swarm optimization has given the time taken for execution of 51, 55, and 57 percentages when the user count is 500, 1000, and 1500 respectively. When the user count is 2000, 2500, and 3000, the time taken for execution of the binary particle swarm optimization is 61, 63, and 65 percent, respectively—about 68, 72, 74, and 75 times taken for execution attained through binary particle swarm optimization when the number of users is 3500, 4000, 4500, and 5000 respectively.

The Whale optimization algorithm has given the time taken for execution of 41, 43, and 46 percentages when the user count is 500, 1000, and 1500, respectively. When the user count is 2000, 2500, and 3000, the time taken for execution of the Whale optimization algorithm is 52, 54, and 57, respectively. About 59, 62, 63, and 64 times are taken for execution attained through the Whale optimization algorithm when the number of users is 3500, 4000, 4500, and 5000, respectively.

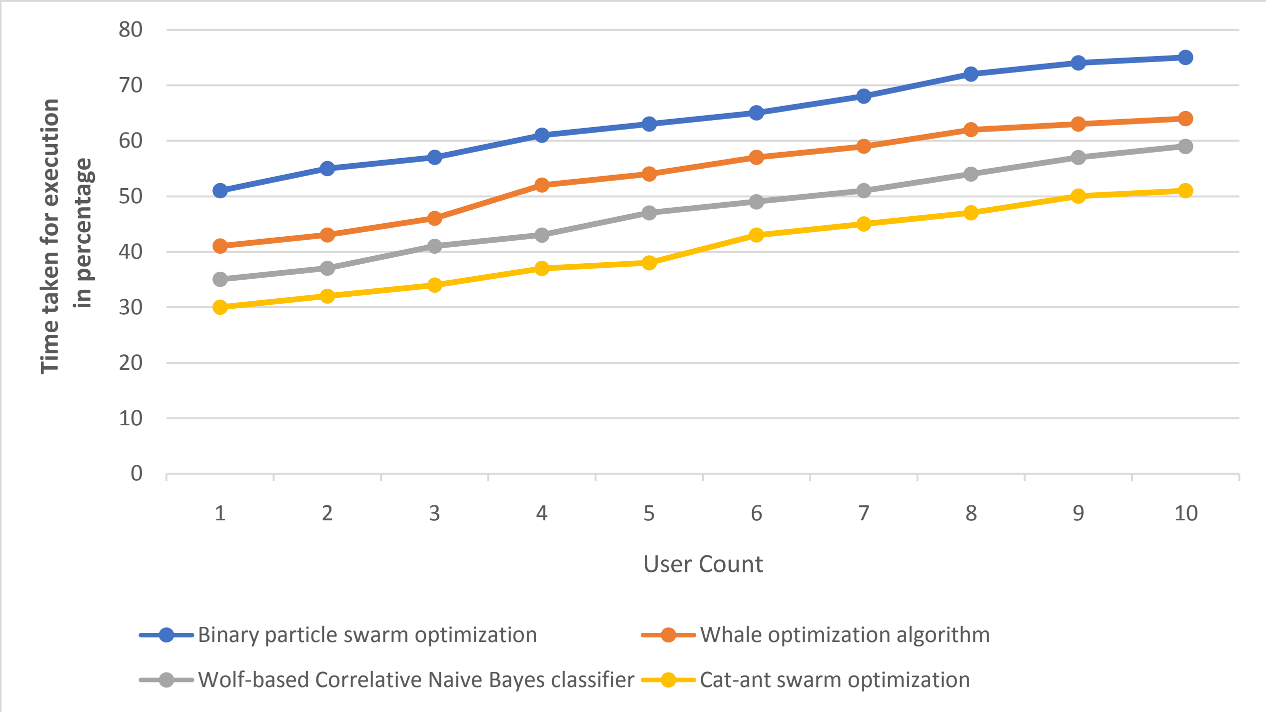

Figure 7: Graphical representation of time taken for execution in percentage

The time taken for execution obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 500, 1000, and 1500 is 35, 37, and 41, respectively. Similarly, the time taken for execution obtained through the Wolf-based Correlative Naive Bayes classifier for the user counts of 2000, 2500, and 3000 is 43, 47, and 49, respectively. For the user counts of 3500, 4000, 4500, and 500, the time taken for execution percentage is 51, 54, 57, and 59, respectively, using the Wolf-based Correlative Naive Bayes classifier.

The time taken for execution obtained through proposed cat-ant swarm optimization for the user counts of 500, 1000, and 1500 is 30, 32, and 34, respectively. Similarly, the time taken for execution obtained through proposed cat-ant swarm optimization for the user counts of 2000, 2500, and 3000 is 37, 38, and 43, respectively. For the user counts of 3500, 4000, 4500, and 500, the time taken for execution is 45, 47, 50, and 51, respectively, using the proposed cat-ant swarm optimization.

.5. Conclusion

A repetitive deep learning-based big data analysis technique has been proposed in this paper. The proposed method has been designed based on the cat-swarm optimization algorithm, which combines the ant-lion optimization and the cat swarm optimization techniques. Deep learning neural network has been brought into context for sequentially processing of big data. The hidden layers present in the neural network combine the input with the past state to produce a new hidden state. The hidden layer consists of neurons that are interlinked to each other, as well as the input and the output states. The results obtained through the proposed technique have been compared with the existing techniques, and it has been proved that the proposed method has provided efficient results among the other algorithms. The time taken for execution obtained through proposed cat-ant swarm optimization for the user counts of 500,1000,1500,2000,2500,3000,3500,4000,4500 and 5000is32,34,37,38,43,45,47,50 and 51 respectively. With respect to the result analysis, while comparing the execution time with all specific algorithms, where Cat-ant swarm optimization algorithm be the best in execution time compare to other algorithms.

References

[1] Himeur, Yassine, AyaSayed, Abdullah Alsalemi, FaycalBensaali, Abbes Amira, IraklisVarlamis, Magdalini Eirinaki, Christos Sardianos, and George Dimitrakopoulos. "Blockchain- based recommender systems: Applications, challenges, and future opportunities." *ComputerScience Review* 43 (2022): 100439.
[2] Carrillo-Mondéjar, J., Jose Luis Martinez, and Guillermo Suarez-Tangil. "On how VoIP attacks foster the malicious call ecosystem." *Computers & Security* 119 (2022): 102758.
[3] Styawati, Styawati, AndiNurkholis, Ahmad Ari Aldino, SelametSamsugi, Emi Suryati, and Ryan PujiCahyono. "Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm." In *2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, pp. 163-167. IEEE, 2022.
[4] Sharma, Kapil, SatishSaini, Shailja Sharma, Hardeep Singh Kang, Mohamed Bouye, and Daniel Krah. "Big Data Analytics Model for Distributed Document Using Hybrid Optimization with-Means Clustering." *Wireless Communications and Mobile Computing* 2022 (2022).
[5] Majumder, Abhishek, Keya Chowdhury, and Joy LalSarkar. "Taxonomy of Shilling Attack Detection Techniques in Recommender System." *Machine Learning Algorithms and Applications in Engineering* (2023): 141.
[6] Al-Zahrani, Adel, and Mohammed Al-Hebbi. "Big Data Major Security Issues: Challenges and Defense Strategies." *Tehničkiglasnik* 16, no. 2 (2022): 197-204.
[7] Sundaresan, S., K. Suresh Kumar, T. Ananth Kumar, V. Ashok, and E. Golden Julie. "Blockchain architecture for intelligent water management system in smart cities." In *Blockchain for Smart Cities*, pp. 57-80.Elsevier, 2021.
[8] Bashabsheh, Mohammad Qassem, LaithAbualigah, and Mohammad Alshinwan. "Big data analysis using hybrid meta-heuristic optimization algorithm and MapReduce framework." In *Integrating meta-heuristics and machine learning for real-world optimization problems*, pp. 181-223. Cham: Springer International Publishing, 2022.
[9] Gao, Zhenyu, and Dimitri N. Mavris. "Statistics and Machine Learning in Aviation Environmental Impact Analysis: A Survey of Recent Progress." *Aerospace* 9, no. 12 (2022): 750.
[10] Ravi, Vinayakumar, VasundharaAcharya, and Tuan D. Pham. "Attention deep learning- based large- scale learning
classifier for Cassava leaf disease classification." *Expert Systems* 39, no. 2 (2022): e12862.
[11] Yang, Xinting, Song Zhang, Jintao Liu, QinfengGao, Shuanglin Dong, and Chao Zhou. "Deep learning for smart fish farming: applications, opportunities and challenges." *Reviews in Aquaculture* 13, no. 1 (2021): 66-90.
[12] Kumar, K. Suresh, A. S. Radhamani, S. Sundaresan, and T. Ananth Kumar. "Medical image classification and manifold disease identification through convolutional neural networks: a research perspective." *Handbook of Deep Learning in Biomedical Engineering and Health Informatics* (2021): 203-225.
[13] Narayana, Satyala, Suresh BabuChandanapalli, MekalaSrinivasaRao, and KalyanapuSrinivas. "Ant cat swarm optimization-enabled deep recurrent neural network for big data classification based on the map-reduce framework." *The Computer Journal* 65, no. 12 (2022): 3167-3180.
[14] Berton, Gabriele, Carlo Masone, and Barbara Caputo. "Rethinking visual geo-localization for large-scale applications." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4878-4888. 2022.
[15] Dabbu, Murali, LoheswaranKaruppusamy, DileepPulugu, SubbaRamaiahVootla, and Venkateswar Reddy Reddyvari. "Water atom search algorithm-based deep recurrent neural network for the big data classification based on spark architecture." *International Journal of Machine Learning and Cybernetics* 13, no. 8 (2022): 2297-2312.
[16] Tair, Milan, NebojsaBacanin, MiodragZivkovic, and K. Venkatachalam. "A chaotic oppositional whale optimization algorithm with firefly search for medical diagnostics." *Comput.Mater.Contin* 72, no. 1 (2022): 959-982.
[17] Abdalla, HemnBarzan, and BelalAbuhaija. "Comprehensive Analysis of Various Big Data Classification Techniques: A Challenging Overview." *Journal of Information & Knowledge Management* (2022): 2250083.
[18] Noor Basha and Ashok Kumar P S. "Early detection of heart disease using machine learning techniques."2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT) 387-391.
[19] Abolarinwa, M. O., A. W. Asaju-Gbolagade, A. A. Adigun, and K. A. Gbolagade. "A Proposed Framework for Optimum Feature Selection using Improved Chicken Swarm Optimization Algorithm for Face Recognition System."*University of Ibadan Journal of Science and Logics in ICT Research* 8, no. 1 (2022): 12-18.
[20] Aghaeipoor, Fatemeh, Mohammad MasoudJavidi, and Alberto Fernández. "IFC-BD: an interpretable fuzzy

classifier for boosting explainable artificial intelligence in big data." *IEEE Transactions on Fuzzy Systems* 30, no. 3 (2021): 830-840.

[21] PB Patil, PM Shastry, Ashok Kumar P S, "Machine learning based algorithm for risk prediction of cardio vascular disease (Cvd)", Journal of critical reviews, 2020

[22] Prabakaran, D., S. Mohammed Nizar, and K. Suresh Kumar. "Software-defined network (SDN) architecture and security considerations for 5G communications." In *Design methodologies and tools for 5G network development and application*, pp. 28-43. IGI Global, 2021.

[23] Shen, Yinghua, WitoldPedrycz, Yuan Chen, Xianmin Wang, and Adam Gacek. "Hyperplane division in fuzzy C-means: Clustering big data." *IEEE Transactions on Fuzzy Systems* 28, no. 11 (2019): 3032-3046.

[24] Wang, Yichuan, LeeAnn Kung, and Terry Anthony Byrd. "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations." *Technological Forecasting and social change* 126 (2018): 3-13.

[25] ElkanoIlintxeta, Mikel, José Antonio Sanz Delgado, EdurneBarrenecheaTartas, HumbertoBustince Sola, and MikelGalarIdoate. "CFM-BD: a distributed rule induction algorithm for building compact fuzzy models in Big Data classification problems." *IEEE Transactions on Fuzzy Systems, 2020, 28 (1), 163-177* (2020).

[26] Ashraf, Murtaza, WillmerRafellQuiñones Robles, Mujin Kim, Young Sin Ko, and Mun Yong Yi. "A loss-based patch label denoising method for improving whole-slide image analysis using a convolutional neural network," *Scientific Reports* 12, no. 1 (2022): 1392.

[27] Wang, Minghua, Qiang Wang, Danfeng Hong, Swalpa Kumar Roy, and Jocelyn Chanussot. "Learning tensor low-rank representation for hyperspectral anomaly detection." *IEEE Transactions on Cybernetics* 53, no. 1 (2022) 679-691.