Spark Architecture and Fractional Artificial Bee Colony-Chaotic Fruitfly RideNN for Big data Classification in Internet of Things

Naeem Th. Yousir

Abstract: ***The typical Internet of Things (IoT) device gathers a huge amount of data specifically termed as big data framework, which transfers the collected data from the sensing layer to the information processing layer. Various big data classification methods are adopted in the industrial applications, and smart cities, but accurately classifying the data in the IoT network poses a challenging task in the research community. Therefore, an effective big data classification model using spark-based architecture is proposed in this research. The big data classification is performed at the master node using the proposed Fractional Artificial Bee Colony- Chaotic Fruitfly Rider Optimization Algorithm (FABC-CFFRideNN). The concept of fictional computing is adopted by the rider optimization algorithm (ROA) to update the position of rider groups based on success rate and the foraging behavior of fruit flies along with the rider parameters is used to enhance to performance of data classification using the proposed CFFRideNN classifier. Moreover, the proposed Fractional Artificial Bee Colony- Chaotic Fruitfly Rider Optimization Algorithm attained better performance using the metrics, namely accuracy, specificity, and sensitivity with the values of 95.382%, 95.81%, and 98.824% for training percentage without node velocity.***

Keywords : Cluster Head (CH), Rider optimization algorithm (ROA), Chaotic Fruitfly algorithm (CFFO), Bhattacharya distance, big data classification

I. INTRODUCTION

Internet of Things (IoT) contains various static objects and mobile devices that are equipped or interconnected using the actuator, sensors and communication modules over internet [1] [2]. IoT [7] is a widespread usage of heterogeneous technologies, and static devices, that are evolved using the interconnectedness mobile devices through the TCP/IP protocol in the physical surroundings [3] [4]. Moreover, the IoT is applicable in various economic areas, which ranges from automation construction and management, smart grid industrial application, smart cities, agriculture, and water grids. Moreover, the sensors adopted in these systems are energy constrained, as they progresses the computational and the storage functions through the lossy channel communication process. The significant and the fundamental IoT driving forces are routing and networking, which assists and drives the interconnected IoT devices. The
factors which are considered while routing the device in the IoT network are energy efficiency, autonomy, secure communication, and scalability [5] [4]. Accordingly, the unique characteristic features of the IoT network make the devices vulnerable to security attacks. Moreover, data communication over the secure layer, and routing becomes a recent research topics in IoT environment [4]. The evolution of IoT increased the authentic network implementation in the smart homes. Due to the easier operation of the smart home system in the IoT device, the human life becomes more comfortable, secure, and convenient. The significant role in the IoT device is the cost-saving, energy consumption, and management flexibility [6]. Moreover, the rapid IoT evolution enhances the work efficiency and life quality of humans. IoT is a pattern of connecting different things, which includes two principal meanings. At first, the groundwork and core of the IoT is the network which is the expansion of internet. Next, the IoT end users are extensive to communication and perform the information exchange process. In general, the typical IoT device gathers a huge amount of data specifically termed as big data framework, which transfers the big data from the sensing layer to the information processing layer [10] [11]. Big data is explained using three different data factors, such as variety, velocity, and volume [12]. It refers that, if the variety, velocity, and volume of data maximized, the current available technologies and techniques failed to handle the data processing and storage systems. Hence, in such situation, the data is termed as big data, and the big data is also considered as data, which simultaneously grows for each year. Technological and scientific revolution of big data affects the data size, as it increases in a daily basis to enhance the profitable actions [8]. Big data is very complex to filter, store, collect, visualize, analyse, and share with the recent technologies [9]. The big data analytics is the process of understanding and analyzing characteristic features of large sized dataset through extracting the geometric and the functional statistical patterns [13]. In general, the originally extracted data is termed as the streaming data, as it represents the interactions, measurements, and actions which arrived from internet. However, the data is generated with respect to the time interval. In the streaming model, the algorithms and the big data progresses strict constraints over the space and time. The streaming approach uses the data structure to provide optimal and speedy answers [14]. Recently, the data gathering mechanism results an incredible performance in the data management system. The diversity, complexity, and volume of big data bring the information to hinder the knowledge extraction and analysis process [10] [11]. Big data classification is effectively carried out in the applications of social media, biomedicine, and marketing. The significant method adopted to solve the big data complexity is the usage of single traditional classification

Revised Manuscript Received October 05, 2019

Naeem Th. Yousir *, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq.

approach [15]. Most of the classifiers utilized in performing the big data classification frameworks are SVM, KNN, Naive Bayes, and so on. Naive Bayes classifier is widely used to perform the classification in data. Moreover, the data fusion model is also introduced to classify the big data [16], and Naive Bayes is used to perform the target tracking, and classification in robotics control, and cloud computing [22] [17]. The scalable and the elastic learning models are required for the cyber analysis, such that text and images are used for the big data analysis model [18] [19]. In general, KNN is a slow classifier, which may not have any testing and training model [20]. SVM is a binary classifier, such that multi SVM classifies the feature vectors into various subsets through trained oracles [19]. The training activity of the SVM classifier is performed by adopting the optimal hyperplane [21]. The primary intention of this research is to design and develop an effective big data classification model using spark architecture. Initially, the data is collected from the IoT nodes and forwarded to CH. The CH receives the data and feed the data to the BS, where the big data classification is performed. The BS is built with the spark framework, which contains two different nodes, namely two master nodes, and a single slave node-set. The master node receives the data from CH and forwards the data to the slave node for selecting the significant features using the Bhattacharya distance. Finally, the selected features are subjected to the data classification using the proposed FABC+CFFRideNN approach. Here, CFFRide is the integration of Fruitfly Optimization Algorithm (CFFO), and Rider Optimization Algorithm (ROA), which engages in the optimal learning of the NN classifier. The major contribution of this research is detailed as follows:

CFFRideNN classifier-based spark framework for big data classification: The big data classification is enabled in the network using the spark framework, which splits the big data using the master node and enables the classification in parallel. On the other hand, the classification in the master nodes is done using the NN classifier, which is trained optimally using the CFFRide algorithm.Proposed CFFRide algorithm: The proposed CFFRide algorithm is the integration of the CFFO algorithm in the standard ROA such that the effectiveness of the ROA towards acquiring the global optimal solution is boosted, which further enables the classification accuracy to be better.The rest of this paper is organized as follows: Section 2 describes the motivation of various existing data classification methods. Section 3 elaborates the system model of spark architecture, and section 4 discusses the proposed FABC+CFFRideNN approach for big data classification in IoT using spark. Finally, section 5 concludes the paper.

II. MOTIVATION

The motivation behind the data classification model using various existing classification methods along with their merits and demerits are discussed in this section.

A. Literature survey

Various existing literature works are surveyed in this section. I. Triguero et al. [8] introduced a divide and conquer model to handled the multiple subsets problem in the Mapreduce paradigm. It independently managed the minority and the majority of sample classes to maintain the subset. It attained extremely imbalanced performance in the big data classification. However, the computing nodes of the positive class were failed to fit in the memory. Mikel Elkano et al. [23] developed a fuzzy rule-based approach for handling the big data by aggregating the rules. However, this approach provides better configuration among the cluster nodes. It effectively handled the big data problems and attained better performance in terms of classification and runtime. However, this method failed to reduce the execution time. Mikel Elkano et al. [25] developed a compact fuzzy model for constructing the compact and accurate fuzzy based rule classification system in big data. It specifically handled the big data problems using apriori algorithm. It offered significant discrimination capability and attained better accuracy, but the classification performance was poor. Junhai Zhai et al. [26] introduced a promising algorithm to perform the binary imbalanced classification in big data. The nearest neighbor positive instances were identified using Mapreduce through uniform distribution. Moreover, the data subsets were generated and trained using the learning model. However, this model was simple to implement, but failed to perform multiple classification in big data. Jesus Maillo et al. [27] introduced a KNN classifier to handle the large sized big data. It effectively extracts the data based on spark. It computes the nearest neighbors by splitting the trained data. However, it attained better performance in terms of accuracy, and runtime, but failed to consider the semi-supervised model. Lakshmanaprabu et al. [28] developed an SVM-based classifier model to extract the features from big data through Mapreduce framework. The unwanted data and noise present in the data were eliminated using gabor filter. In enhanced the efficiency of the system by mapping and reducing the dataset, but failed to recover the path. Peng Li et al. [29] modelled a deep convolutional-based computation approach for learning the hierarchical features from big data. It utilized the topologies and the local features in big data. It prevents the over fitting and enhanced the efficiency of training. However, it attained better accuracy, but the computational complexity was high. S. K. Lakshmanaprabu et al. [30] developed a MapReduce and random forest classifier model to handle the big data in healthcare system. It effectively selects the optimal attributes to perform the data classification process. Based on the optimal features, the e-health data was classified. The computational performance was less for large sized database.

B. Challenges

The challenges associated with the data classification models are discussed in this section.

- Data collection, event sifting, data combination, and deliberation associated with the data mining results a significant issue in the multi layer model [30].
- To perform the data cleaning, storage, and data collection in the real time analytics model faces several challenges in the IoT technology. However, providing utility to the IoT classification model is a challenging issue in the optimization mechanism [28].
- To deal with the diversity, complexity, and volume in the big data poses a challenging issue in the research community. However, to use multiple Mapreduce data jobs over big data implies extra cost,

which is a challenging issue in the data classification model [27].

- The traditional privacy and security models are very complex to analyze the big data associated with the IoT technology. Moreover, data provenance, and access control in the distributed system affects the real time environment [9].
- To predict the intrusion attack in the traffic data poses a challenging issue in big data classification. However, it was very complex to handle the data classification using the geometric learning model [13].
- Based on the requirements of data processing, adaptively changing the storage infrastructure of big data system poses a challenging issue in spark environment.

III. SYSTEM MODEL OF IOT NETWORK

The system model of IoT network is enumerated in Fig. 1. IoT network consists of IoT nodes, which is distributed in the environment engaged in collecting the data. The nodes communicate the data with the CH and the data from the CHs is forwarded to the BS. The IoT nodes present in the IoT network is denoted as such that the network contains number of nodes, which is expressed as, . The CHs present in the network is denoted as, , where denotes the total number of CHs in the network. The data from the CH is communicated with the BS, which performs the classification on the collected data in the spark framework. The data communication from the CH to BS is based on the routing protocol named FABC [33], which is the energy-aware protocol. The system model of the proposed data classification approach is depicted in Fig. 1.

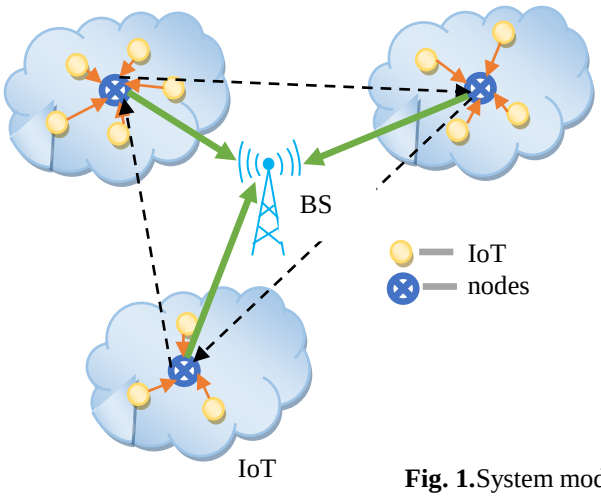


Fig. 1. System model of IoT network

IV. SPARK ARCHITECTURE-BASED BIG DATA CLASSIFICATION IN BASE STATION USING PROPOSED FRACTIONAL ARTIFICIAL BEE COLONY-CHAOTIC FRUITFLY RIDER OPTIMIZATION ALGORITHM

The primary goal of this research is to perform the big data classification in spark architecture using the proposed FABC + CFFRideNN model. Initially, let us assume the architecture in Fig. 2, which comprises of three CHs. The IoT nodes collect the data from the environment and forward the data to the CH represented as, , which further send the data to BS. The BS receives the data and performs the data classification through optimization enabled NN dependent spark architecture. Thus, the transmission between the CH and BS is based on a routing protocol, FABC, which is an energy-efficient protocol that smoothen the communication process. Hence, the data received in the BS is processed using the proposed spark architecture-based classification strategy. The spark architecture involves two functioning modules, such as master node, and slave node. The master node is responsible for splitting the data to the slave node, where the feature selection process using the Bhattacharya distance is done in order to select the optimal features effectively. Finally, the selected unique features are further processed in the master node, where the big data classification is done effectively using the proposed FABC+CFFRideNN approach. The NN classifier is an effective classifier and is employed for classification and the learning in the classifier is enabled using the proposed CFFRide algorithm, which is the integration of CFFO [31], and ROA [32]. Fig. 2 shows the architecture of the proposed FABC+CFFRideNN approach for data classification in IoT using spark. The spark architecture consists of two master nodes and a single slave node-set. The data collection and the classification are carried out in the master node, while the feature selection process is performed in the slave node.

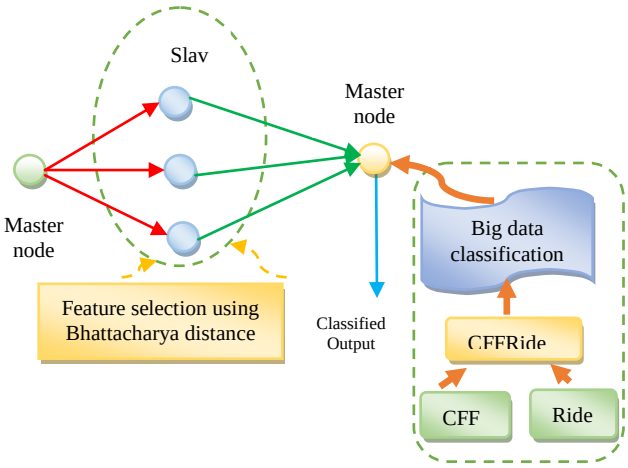


Fig. 2.Spark architecture-based big data classification in BS

A. Read the data at the master node

The data collection process is carried out by the nodes in the network, which is forwarded to the BS through the CHs. The IoT nodes present in the IoT network is mathematically

expressed as, [illegible] where, [illegible] denotes the IoT network, [illegible] denotes the [illegible] IoT node, and [illegible] denotes the total IoT nodes. The data collected by the IoT node [illegible] is expressed as,

[illegible] Eq. (1)

where, [illegible] denotes the data collected by the [illegible] node [illegible] and the total data attributes available in the collected data by [illegible] be [illegible]. The collected data by the nodes is send to the CH and then, to the base station based on the FABC routing protocol [33]. Thus, in the BS, the classification process using spark architecture is enabled, and in the spark architecture, the proposed CFFRideNN approach is used for classification. Accordingly, the collected data from all the nodes is read in the master node, which is denoted as, [illegible] and possesses the dimension of [illegible], which is fed as input to the slave nodes in the spark.

B. Feature selection using Bhattacharya distance in the slave nodes

Once the master node read the data then, the feature selection process is carried out in the slave node-set using the Bhattacharya distance measure. In the slave node, the information received from the master node [illegible] is partitioned using Bhattacharya distance. Instead of processing the data with the entire features, the selective features can be used for classification in order to achieve higher classification accuracy. Hence, the resultant feature is indicated as [illegible], which contains the features of data based on its attributes and the dimension of [illegible] is [illegible]. The Bhattacharya distance measure is mathematically specified using the below equation as,

[illegible] (2)

where, $P(b,c)$ indicates the Bhattacharyya distance between b and c, [illegible] denotes the variance of bth distribution, [illegible] specifies the variance of cth distribution, N_b represents the mean of bth distribution, N_c denotes the mean of cth distribution, and b and c are the distributions. The selected features using the bhattacharya distance is denoted as,

[illegible] (3)

It is clearly shown that [illegible], which specifies that the dimension is minimized through the election of the selective features in order to increase the classification accuracy.

C. Proposed Fractional Artificial Bee Colony Chaotic Fruitfly Rider Optimization algorithm for data classification in the master node

The features selected F in the slave node are forwarded to the master node to perform the classification using the CFFRideNN approach in the spark platform. The master node receives the unique features selected by the slave node, and performs the classification process with that unique feature in the master node. The data classification is performed in the master node based on the selected features using the proposed FABC+CFFRideNN. Moreover, CFFRide algorithm is the integration of CFFO [31] and ROA [32] algorithm, which engages in optimal tuning of the NN. Even though the fictional concept of RideNN enables an effective performance, the foraging behavior of CFF optimization is incorporated with the parametric features of RideNN to produce outstanding performance of data classification in spark. RideNN is nothing, but the NN classifier trained using ROA [], which operates using the fictional computing concept and used to solve the optimization problems. ROA considers four rider groups, whose aim is to move towards the target to win the riding race. However, the number of riders defined in each group is equally selected from the total number of riders. The four rider groups defined in this model is overtaker, attacker, follower, and by pass rider. The main parameters considered to ride the vehicle are accelerator, brake, steering, and gear. However, the rider update their position by modifying these parameters based on the success rate.

C1. Solution encoding: It is the representation of solution, which is intended to be determined using the proposed CFFRide optimization that trains NN classifier in order to enable effective learning in NN that further boosts the classification performance. The solution is nothing but the best optimal solution required for training the NN classifier.

C2. Fitness function: The fitness function is evaluated to compute the best solution, which is expressed as,

[illegible] (4)

where, H is the accuracy, I is the sensitivity, and W is the specificity.

C3. CFFRideNN for big data classification:

The CFFRideNN classifier is used to achieve the big data classification using the spark architecture. Here, the RideNN uses the NN classifier, which is trained by the ROA to update the position of rider groups. The CFF optimization algorithm includes the parametric features to the RideNN classifier to perform the data classification more effective. The NN is a multimodal and nonlinear function aims to use the ROA for training the classifier based on the features of rider groups. The key role of ROA is to accelerate the rider position towards the leading rider. It eliminates the local minima with the small local neighborhood that is handled by attacker. The rider groups initially explore their position is a random order for reaching the target location. In the rider groups, the follower uses the multidirectional search space, while the overtaker selects the optimal directional space.

a. Architecture of NN classifier

The NN architecture consists of three different layers, as input layer, hidden layer, and output layer. Each layer contains number of neurons in the NN architecture. The input given to the NN classifier is the selected

features [illegible] NN performs the classification using the weights that are assigned to the neurons at each layer in the network. The weight given to the neurons in the hidden layer is specified as,

[illegible] (5)

The bias applied to the hidden layer is specified as, [illegible], which is utilized to compute the potential. The weight of the output layer neurons is indicated as, [illegible] and the output layer bias is denoted as, [illegible]. The function used to calculate the output of classifier is represented as,

[illegible] (6)

where, [illegible] indicates the transfer function of log-sigmoid, which computes the output based on input, [illegible] represents the input of [illegible] neuron, [illegible] is the weight of [illegible] neuron, [illegible] denotes the weight of the output layer, [illegible] and [illegible] represents the biases.

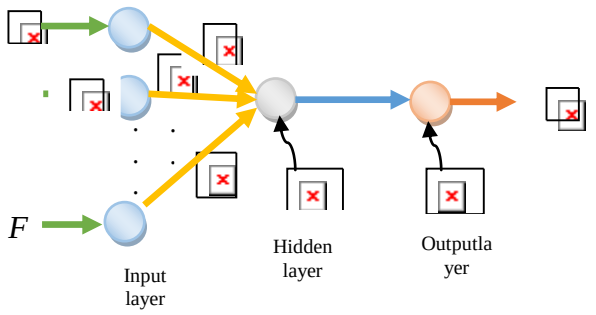


Fig.3 .Architecture of NN classiifer

b. CFFRide algorithm for optimal tuning of NN classifier: The algorithmic procedure of CFFRide algorithm is mathematically explained as follows:

Rider population and parameter initialization: The population is initialized with four rider groups, which is represented as, [illegible]. The position of the rider groups are assigned in a random manner such that the rider group initialization is expressed as,

[illegible] (7)

where, [illegible] represents the number of riders, and [illegible] indicates the number of coordinates or dimension, and $Y_l(u,v)$ represents the location of uth rider at time l. Moreover, the steering angle of the rider is indicated as B, the accelerator of rider vehicle is represented as e, the brake of uth rider is indicated as C, and the gear of vehicle is indicated as D, respectively.

c. Compute the success rate: The success of the rider group is computed as per the fitness function mentioned in equation (4).

d. Find the leading rider: The leading rider is identified based on the success rate and the rider having minimum distance and higher success rate is determined as leading rider. However, the leading rider may change when the location of rider changes. Hence, the best rider or best solution depends on the maximal success rate.

e. Rider position update: The location of rider is updated for finding the leading one, such that the rider changes their location based on the characteristic features of riders. Hence, the location update process for each rider group is discussed as follows:

Position update process of bypass rider: The bypass rider is the first rider group, who bypasses the leading path and wins the race such that it does not follow the path of leading rider. Therefore, the update equation of bypass rider is mathematically expressed as,

[illegible] (8)

where, [illegible] denotes the random number, which lies in the range between [0,1], [illegible] denotes the random number lies between [1,J], and [illegible] represents the random number lies in the range of 0 to 1, respectively. Let us assume [illegible], such that modified above equation is expressed as,

[illegible] (9)

The equation (9) highlights the standard equation of bypass rider, which is modified using CFFO in order to inherit the properties of CFFO. The purpose of modifying the standard equation of bypass rider is detailed below: By incorporating the foraging behavior of CFFO with fictional computing based approach, the parameters from both the algorithms are utilized to attain effective data classification. The CFFO algorithm determines the location of fruit fly swarm by selecting the best solution among the random solutions. The CFFO considers the novel parameter named chaos to enhance the performance of the data classification. The concentration value with the maximum smell is assigned to the foraging phase, such that the swarm will be directed to move towards the food source. The CFFO meta-heuristic technique considers the randomization of $Y_{u,v}$ variables based on the uniform distribution. In order to increase the overall speed and convergence rate of fruit flies, the parameter named 'alpha' is used to generate the food sources in CFFRideNN algorithm. Hence, the chaotic variable of the CFFO algorithm is mathematically modeled as,

[illegible] (10)

[illegible] (11)

[illegible] (12)

By substituting the Eq. (15) in Eq. (8), the resultant updated equation is expressed as,

(13)

(14)

(15)

(16)

(17)

(18)

where, denotes the random number, which lies in the range between [0,1], denotes the random value lies between [1,*J*], and represents the random number lies in the range of 0 to 1, respectively. Hence, equation (18) highlights the update rule of proposed CFFO-Ride

Position update process of follower: The follower changes its position based on the location of leading rider. The follower follows the leading rider path, such that it effectively and quickly reaches the target. The update equation of the follower is mathematically expressed as,

(19)

where, g denotes the coordinate selector, Y^O represents the leading rider location, O specifies the index of leading rider, $B_{u,g}$ indicates the steering angle of uth rider, and h^l_u denotes the distance travelled by u^{th} rider.

Position update process of overtaker: The overtaker update their location based on the indicators, such as direction indicator, coordinate selector, and success rate. However, overtaker considers its own location to win the race. Therefore, the position of overtaker is updated based on current rider, leading rider, and directional indicator, which is mathematically expressed as,

(20)

where, $Y_l(u,g)$ denotes the location of uth rider at g^{th} coordinate, and $P_l(u)$ denotes the direction indicator, respectively. However, direction indicator is computed using the success rate, which is expressed as,

(21)

Where, $E_l(u)$ specifies the success rate of uth rider at time l.

Update process of attacker: The attacker tries to take the leader position using higher speed to reach the target. However, the position update process of attacker is expressed as,

(22)

where, $Y^o(O,v)$ denotes the leading rider position, specifies the steering angle, and indicates the distance travelled by u^{th} rider.

f. Rider parameter update: The rider parameter is updated to find the optimal solution based on the activity counter, steering angle, gear, accelerator, and brake with respect to success rate.

g. Termination: The above process is repeated until the best solution is attained or reaches the maximum iteration.

Algorithm-I: shows the pseudo code of proposed FABC+CFFRideNN algorithm for data classification.

Sl. No	Pseudo code of FABC + CFFRideNN approach
1	position of rider is randomly assigned as
2	The leading rider is specified as
3	begin
4	Population initialization
5	Rider parameter initialization: steering angle , gear , brake , and accelerator
6	Success rate computation
7	while ; -off time
8	for 1 to
9	Modify the Bypass rider position using Eq. (18)
10	Update the follower position using Eq. (19)
11	Modify the overtaker position using Eq. (20)
12	Modify the attacker position using Eq. (22)
13	Rider ranking based on
14	Select the rider having maximum as leading rider
15	Update , and
16	return
17	
18	end for
19	end while
20	terminate

The optimization task of FABC+CFFRideNN offers better convergence rate, and attained global optimality with respect to the success rate. The major influence of the CFFO algorithm is the computation of food sources. The foraging phase of the fruit fly uses the foraging behavior for searching

and locating the food sources among the swarm. Hence, by integrating the fictional computing of ROA with the foraging behavior of fruit flies increased the performance of data classification in IoT using spark architecture.

V. RESULTS AND DISCUSSION

The results and discussion made using the proposed FABC+CFFRideNN algorithm is explained in this section. Moreover, the performance of the proposed algorithm is evaluated using the performance metrics, like accuracy, sensitivity, specificity, and throughput.

A. Experimentation setup

The implementation of the proposed algorithm is carried out in the MATLAB tool using Heart disease dataset [24]. The Heart disease dataset is obtained from UCI machine learning repository, which consists of 76 attributes. Here, the presence of disease is indicated as 1, 2, 3, or 4, while the absence of disease is specified as 0, respectively.

B. Evaluation metrics

The performance of the proposed CFFRideNN algorithm is evaluated and analyzed using the evaluations metrics, such as accuracy, sensitivity, specificity, and throughput.

Accuracy: It is termed as the ratio of observed true results with that of total results. Therefore, accuracy is computed using the below equation as,

(23)

where, denotes the true positive, specifies the true negative, indicates the false positive, denotes the false negative, and denotes the accuracy, respectively.

Sensitivity: It is defined as the measure of true positive rate to the probability of positive observation, which is mathematically expressed as,

(24)

Specificity: It is defined as the ratio of true negative measure to the probability of negative observed results, which is mathematically specified as,

(25)

Throughput: It is the rate at which the number of nodes transferred per unit time.

C. Comparative methods

The performance of the proposed FABC+CFFRideNN algorithm is analyzed and is compared with the existing methods, such as Leach+Mapreduce +Elephant Heard Optimization +Linear Kernel-Support Vector machine (Leach+Mapreduce+EHO+LK-SVM) [28], Wireless Body Area Network +Deep Convolutional Neural Network (WBAN+Deep CNN) [29], Random Forest (Artificial Bee Colony +RF) [30], and Particle Swarm Optimization+Neural network (PSO+NN).

D. Comparative analysis

The comparative analysis of the proposed FABC+FFRideNN by varying the selected features and training percentage in terms of accuracy, sensitivity, and specificity is explained in this section.

D.1 Comparative analysis of IoT nodes without moving velocity

The comparative analysis of accuracy, specificity, and sensitivity by varying the selected features and training percentage without moving the IoT nodes is elaborated in this section.

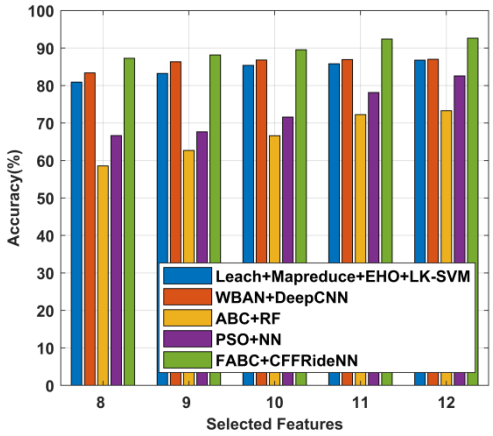
A. Comparative analysis by varying selected features

Fig. 4 (a) shows the comparative analysis of accuracy by varying the selected features. When selected features=8, the accuracy obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.911%, 83.360%, 58.552%, and 66.661%, while the proposed FABC+FFRideNN attained better accuracy of 87.273%, respectively. When selected features=9, the accuracy obtained by the proposed FABC+FFRideNN is 88.161%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 5%, 2%, 40%, and 30%, respectively. When selected features=10, the accuracy obtained by the proposed FABC+FFRideNN is 89.527%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 4%, 3%, 34%, and 25%, respectively. When selected features=11, the accuracy obtained by the proposed FABC+FFRideNN is 92.382%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 6%, 27%, and 18%, respectively. When selected features=12, the accuracy obtained by the proposed FABC+FFRideNN is 92.632%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 6%, 6%, 26%, and 12%, respectively.

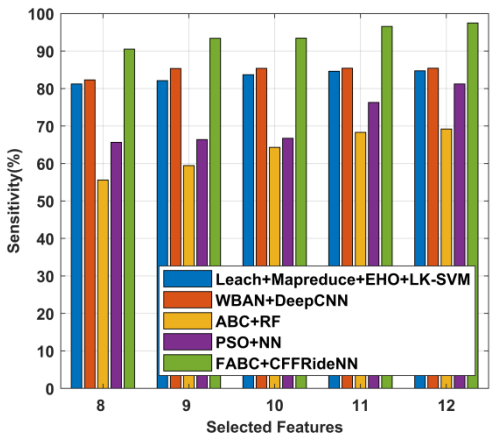
Fig. 4 (b) shows the analysis of sensitivity with respect to the selected features. When selected features=8, the sensitivity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 81.210%, 82.302%, 55.579%, and 65.656%, while the proposed FABC+FFRideNN attained better sensitivity of 90.501%, respectively. When selected features=9, the sensitivity obtained by the proposed FABC+FFRideNN is 93.403%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 13%, 9%, 57%, and 40%, respectively. When selected features=10, the sensitivity obtained by the proposed FABC+FFRideNN is 93.431%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 11%, 9%, 45%, and 40%, respectively. When selected features=11, the sensitivity obtained by the

existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 84.610%, 85.403%, 68.309%, and 76.276%, while the proposed FABC+FFRideNN attained better sensitivity of 96.562%, respectively. When selected features=12, the sensitivity obtained by the proposed FABC+FFRideNN is 97.473%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 15%, 14%, 40%, and 20%, respectively.

Fig. 4 (c) shows the analysis of specificity by varying the selected features. When selected features=8, the specificity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.762%, 84.532%, 62.929%, and 66.821%, while the proposed FABC+FFRideNN attained better specificity of 87.731%, respectively. When selected features=9, the specificity obtained by the proposed FABC+FFRideNN is 91.206%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 4%, 29%, and 28%, respectively. When selected features=10, the specificity obtained by the proposed FABC+FFRideNN is 93.145%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 6%, 5%,



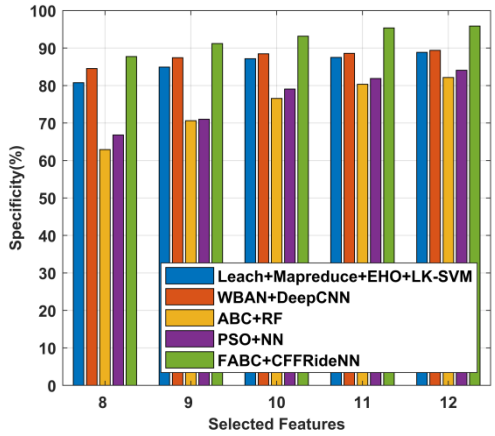
21%, and 17%, respectively. When selected features=11, the specificity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 87.500%, 88.587%, 80.314%, and 81.854%, while the proposed FABC+FFRideNN attained better specificity of 95.359%, respectively. When selected



features=12, the specificity obtained by the proposed FABC+FFRideNN is 95.841%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 7%, 16%, and 14%, respectively.

(a)

(b)



(c)

Fig. 5. Comparative analysis based on selected features without considering velocity, (a) accuracy, (b) sensitivity, (c) specificity

B. Comparative analysis by moving the IoT nodes with 3 meter/ minute

The comparative analysis of accuracy, specificity, and sensitivity by varying the selected features and training percentage by moving the nodes with the speed of 3 meter/min is elaborated in this section.

a) Comparative analysis by varying selected features

Fig. 6 (a) shows the comparative analysis of accuracy by varying the selected features. When selected features=8, the accuracy obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.562%, 83.549%, 64.941%, and 73.627%, while the proposed FABC+FFRideNN attained better accuracy of 84.762%, respectively. When selected features=9, the accuracy obtained by the proposed FABC+FFRideNN is 89.350%.

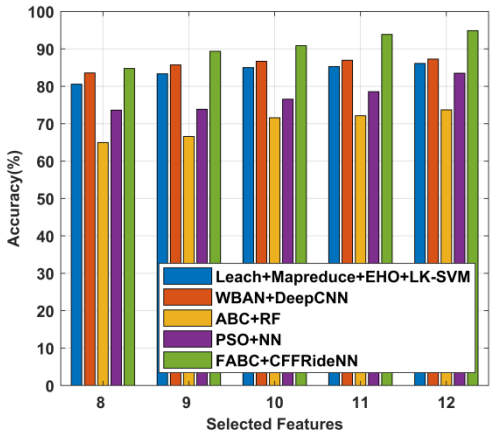
Hence, the percentage of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

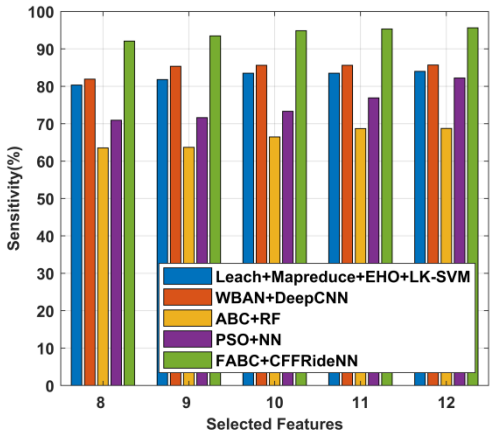
improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 4%, 34%, and 20%, respectively. When selected features=10, the accuracy obtained by the proposed FABC+FFRideNN is 90.837%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 6%, 4%, 26%, and 18%, respectively. When selected features=11, the accuracy obtained by the proposed FABC+FFRideNN is 93.870%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 10%, 7%, 30%, and 19%, respectively. When selected features=12, the accuracy obtained by the proposed FABC+FFRideNN is 94.849%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 10%, 8%, 28%, and 13%, respectively.

Fig. 6 (b) shows the analysis of sensitivity with respect to the selected features. When selected features=8, the sensitivity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.33%, 81.90%, 63.54%, and 70.93%, while the proposed FABC+FFRideNN attained better sensitivity of 92.08%, respectively. When selected features=9, the sensitivity obtained by the proposed FABC+FFRideNN is 93.46%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 14%, 9%, 46%, and 30%, respectively. When selected features=10, the sensitivity obtained by the proposed FABC+FFRideNN is 94.87%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 13%, 10%, 42%, and 29%, respectively.

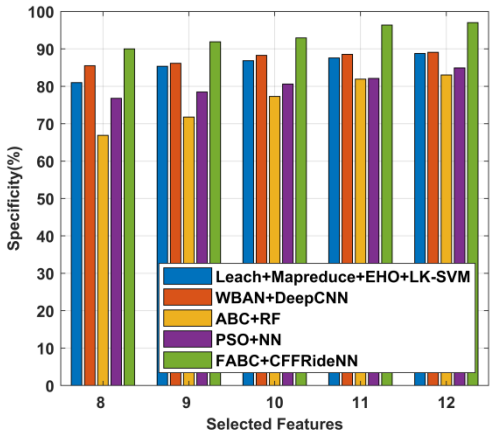
Fig. 6 (c) shows the analysis of specificity by varying the selected features. When selected features=8, the specificity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.98%, 85.51%. 66.89%, and 76.77%, while the proposed FABC+FFRideNN attained better specificity of 90.00%, respectively. When selected features=9, the specificity obtained by the proposed FABC+FFRideNN is 91.89%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 6%, 28%, and 17%, respectively. When selected features=10, the specificity obtained by the proposed FABC+FFRideNN is 92.92%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 7%, 5%, 20%, and 15%, respectively.



(a)



(b)



(c)

Fig. 6. Comparative analysis based on selected features with the movement of nodes at 3m/min, a) accuracy, b) sensitivity, c) specificity

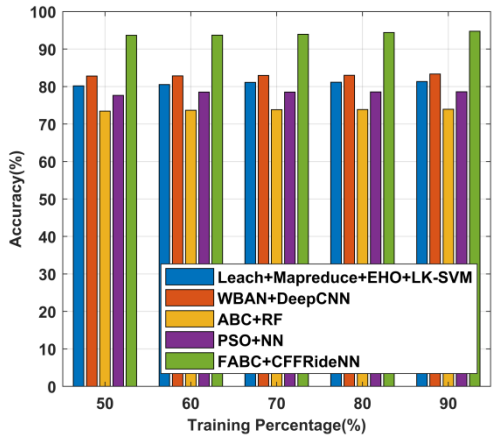
b) Comparative analysis by varying training percentage

Fig. 7 (a) shows the analysis of accuracy by varying the training percentage. When training percentage=50%, the accuracy obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.168%, 82.813%, 73.431%, and 77.631%, while the proposed FABC+FFRideNN attained better accuracy of 93.672%, respectively. When training percentage=60%, the accuracy

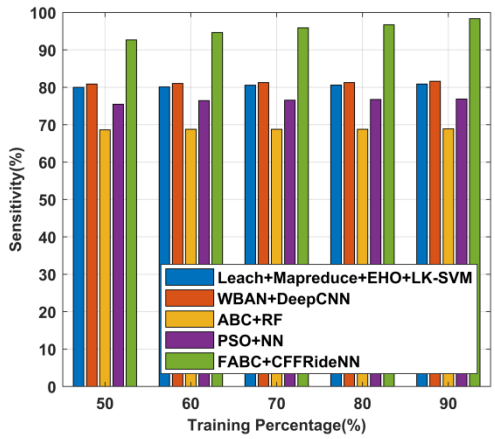
obtained by the proposed FABC+FFRideNN is 93.721%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 16%, 13%, 27%, and 19%, respectively. When training percentage=70%, the accuracy obtained by the proposed FABC+FFRideNN is 93.921%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 15%, 13%, 27%, and 19%, respectively. When training percentage=80%, the accuracy obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 81.130%, 82.990%, 73.859%, and 78.561%, while the proposed FABC+FFRideNN attained better accuracy of 94.406%, respectively.

Fig. 7 (b) shows the analysis of sensitivity with respect to the training percentage. When training percentage=50%, the sensitivity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 79.97%, 80.88%, 68.63%, and 75.47%, while the proposed FABC+FFRideNN attained better sensitivity of 92.66%, respectively. When training percentage=60%, the sensitivity obtained by the proposed FABC+FFRideNN is 94.62%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 18%, 16%, 37%, and 23%, respectively. When training percentage=70%, the sensitivity obtained by the proposed FABC+FFRideNN is 95.89%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 19%, 18%, 39%, and 25%, respectively. When training percentage=80%, the sensitivity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 80.59%, 81.27%, 68.79%, and 76.75%, while the proposed FABC+FFRideNN attained better sensitivity of 96.69%, respectively.

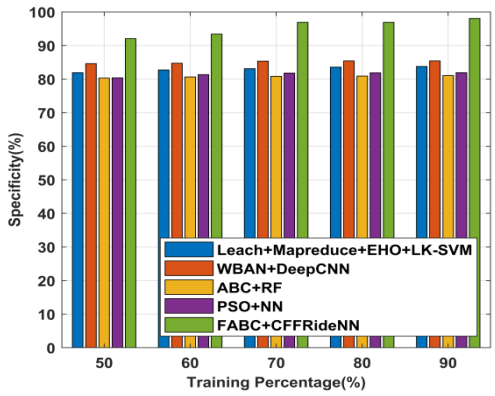
Fig. 7 (c) shows the analysis of specificity by varying training percentage. When training percentage=50%, the specificity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 81.906%, 84.606%, 80.295%, and 80.375%, while the proposed FABC+FFRideNN attained better specificity of 92.063%, respectively. When training percentage=60%, the specificity obtained by the proposed FABC+FFRideNN is 93.381%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 12%, 10%, 15%, and 14%, respectively. When training percentage=70%, the specificity obtained by the proposed FABC+FFRideNN is 96.907%. Hence, the percentage of improvement reported when comparing the proposed FABC+FFRideNN with the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 16%, 13%, 19%, and 18%, respectively. When training percentage=80%, the specificity obtained by the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 83.558%, 85.397%, 80.896%, and 81.890%, while the proposed FABC+FFRideNN attained better specificity of 96.911%, respectively.



(a)



(b)



(c)

Fig. 7. Comparative analysis based on training percentage with the movement of nodes at 3m/min, (a) accuracy, (b) sensitivity, (c) specificity

C. Comparative analysis of throughput

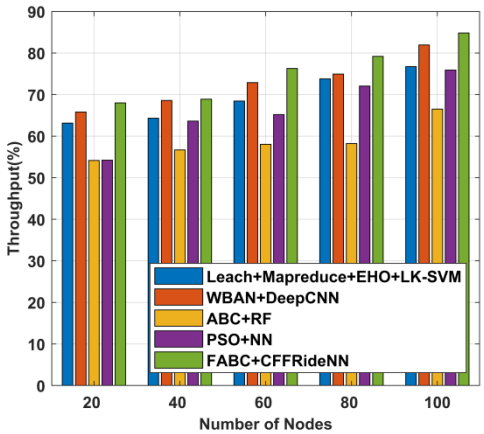
Here, the number of rounds considered to perform the routing is 500, such that the nodes allocated to entire the routing process are 20, 40, 60, 80, and 100. Initially 20 nodes are assigned to perform the routing process for 500 rounds, then 40 nodes are allocated to perform the same routing process at 500 rounds. Later 60 nodes are assigned to perform the routing process for 500 rounds, and then 80 nodes are assigned for routing at 500 rounds, and

finally100 nodes are assigned to perform the routing process for 500 rounds.

Fig. 8 (a) shows the analysis of throughput by varying the number of nodes without velocity. When number of nodes=40, the throughput obtained by the existing methods, such as Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 71.806%, 81.939%, 68.059%, and 71.720%, while the proposed FABC+FFRideNN attained better throughput of 90.289%, respectively.

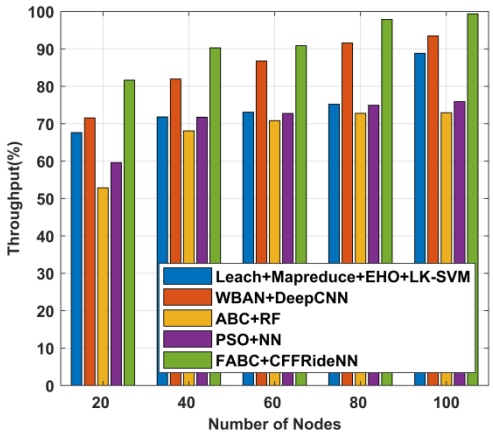
Fig. 8 (b) shows the analysis of throughput by varying the number of nodes with the movement of nodes at 3 meter/min. When number of nodes=60, the throughput obtained by the existing methods, such as Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN is 68.443%, 72.874%, 58.044%, and 65.190%, while the proposed FABC+FFRideNN attained better throughput of 76.287%, respectively.

(a)



(b)

Fig. 8. Comparative analysis of number of nodes, (a) throughput without node velocity, (b) throughput with the movement of 3 meter/min



From the observed result it is noticed that the throughput increases when the number of node increases, which further lead to increase the performance. When a greater number of nodes are used, the delay might be reduced, which results effective training and performance.

E. Comparative discussion

The comparative discussion made using the proposed FABC+FFRideNN is explained in this section.

Table- I: Comparative analysis of nodes without velocity

Metrics	Accuracy		Sensitivity		Specificity	
Methods	*Selected features (12)*	*Training percentage (90%)*	*Selected features (12)*	*Training percentage (90%)*	*Selected features (12)*	*Training percentage (90%)*
Leach+Mapreduce+EHO+LK-SVM	86.746	81.559	84.711	82.687	88.851	82.24
WBAN+Deep CNN	86.998	83.454	85.405	82.716	89.386	84.44
ABC+RF	73.266	73.433	69.201	68.487	82.139	80.48
PSO+NN	82.548	78.396	81.201	76.652	84.078	81.92
Proposed FABC+FFRideNN	92.632	95.382	97.473	98.824	95.841	95.81

Table- I shows the comparative discussion of nodes without velocity. The highest accuracy obtained by the proposed FABC+FFRideNN is 92.632 with the selected features of 12, while the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN obtained the accuracy for 12 selected features is 86.746,86.998, 73.266, and 82.548, respectively. The maximum sensitivity obtained by the proposed FABC+FFRideNN is 98.824 for the training percentage of 90%. Moreover, the maximum specificity obtained by the FABC+FFRideNN is 95.841 for selected features=12.

Table- II: Comparative analysis of nodes with the movement of 3 meter/min

Metrics	Accuracy		Sensitivity		**Specificity**	
Methods	*Selected features (12)*	*Training percentage (90%)*	*Selected features (12)*	***Training percentage (90%)***	***Selected features (12)***	***Training percentage (90%)***
Leach+Mapreduce+EHO+LK-SVM	86.106	81.323	83.97	80.88	88.78	83.76
WBAN+Deep CNN	87.251	83.332	85.69	81.62	89.06	85.429
ABC+RF	73.684	73.943	68.74	68.9	83.04	81.06
PSO+NN	83.472	78.592	82.21	76.87	84.92	81.892
Proposed FABC+FFRideNN	94.849	94.747	**95.62**	**98.31**	**97.01**	**98.051**

Table-II shows the comparative discussion of nodes with the movement of 3 meter/min. The highest accuracy obtained by the proposed FABC+FFRideNN is 94.849 with the selected features of 12, while the existing methods, like Leach+Mapreduce+EHO+LK-SVM, WBAN+Deep CNN, ABC+RF, and PSO+NN obtained the accuracy for 12 selected features is 86.106, 87.251, 73.684, and 83.472, respectively. The maximum sensitivity obtained by the proposed FABC+FFRideNN is 98.31 for the training percentage of 90%. Moreover, the maximum specificity obtained by the FABC+FFRideNN is 97.01 for selected features=12.

VI. CONCLUSION

In this research, an effective Fractional Artificial Bee Colony Chaotic Fruitfly Rider Optimization algorithm is developed to perform the big data classification using spark-based architecture. At first, the data is collected from the IoT nodes and forward the collected data to Cluster Head. The Cluster Head receives the data and passed the data to the Base Station, where the big data classification is performed using the proposed algorithm with spark architecture. Here, the big data classification is at the master node using the proposed Fractional Artificial Bee Colony- Chaotic Fruitfly Rider Optimization Algorithm. Rider Optimization Algorithm uses the concept of fictional computing to update the position of rider groups, which further used to increase the performance of big data classification. The proposed Chaotic Fruitfly Rider Optimization algorithm attained better performance using the metrics, such as accuracy, specificity, and sensitivity with the values of 95.382%, 95.81%, and 98.824% for training percentage without node velocity. In future, the performance of the data classification process will be enhanced using some other optimization algorithm.

REFERENCES

1. Maha Bouaziz and Abderrezak Rachedi," A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology", Computer Communications, vol.74, pp. 3-15, January 2016.
2. Amol V. Dhumane and Rajesh S. Prasad, "Multi-objective fractional gravitational search algorithm for energy efficient routing in IoT," Wireless Networks, pp.1-15, 2017.
3. Thomas Watteyne,Antonella Molinaro,Maria Grazia Richichi and Mischa Dohler," From MANET To IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols", IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 688 – 707, September 2010.
4. David Airehrour,Jairo A. Gutierrez and Sayan Kumar Ray," SecTrust-RPL: A Secure Trust-Aware RPL Routing Protocol for Internet of Things", Future Generation Computer Systems,March 2018.
5. Terence K.L. Huia, R. Simon Sherratt and Daniel Díaz Sánchez," Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies",Future Generation Computer Systems, vol. 76, pp. 358-369, November 2017.
6. Daemin Shin, Vishal Sharma, Jiyoon Kim, Soonhyun Kwon and Ilsun You," Secure and Efficient Protocol for Route Optimization in PMIPv6-based Smart Home IoT Networks", IEEE Access, vol.5, pp.11100 – 11117,June 2017.
7. Zheng-Yang Ai, Yu-Tong Zhou and Fei Song," A Smart Collaborative Routing Protocol for Reliable Data Diffusion in IoT Scenarios",vol.18,no.6, June 2018.
8. I. Triguero, M. Galar, D. Merino, J. Maillo, H. Bustince, F. Herrera, "Evolutionary Undersampling for Extremely Imbalanced Big Data Classification under Apache Spark", Knowledge-Based Systems, 117, pp.3-15, 2017.
9. Terzi, D.S., Terzi, R. and Sagiroglu, S., "A survey on security and privacy issues in big data", In proceedings of 10th International Conference for Internet Technology and Secured Transactions (ICITST) pp. 202-207, December 2015.
10. M. Minelli, M. Chambers, and A. Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", 2013.
11. Maillo, J., Triguero, I. and Herrera, F., "A mapreduce-based k-nearest neighbor approach for big data classification", IEEE Trustcom/BigDataSE/ISPA, vol. 2, pp. 167-172, August 2015.
12. Srinivasan, A., Faruquie, T.A. and Joshi, S., "Data and task parallelism in ILP using MapReduce", Machine learning, vol.86, no.1, pp.141-168, 2012.
13. Suthaharan, S., "Big data classification: Problems and challenges in network intrusion prediction with machine learning", ACM Sigmetrics Performance Evaluation Review, vol.41, no.4, pp.70-73. 2014.
14. Zhang, C., Li, F. and Jestes, J., "Efficient parallel kNN joins for large data in MapReduce", In Proceedings of the 15th International Conference on Extending Database Technology, pp. 38-49, March 2012.
15. Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., Benítez, J.M. and Herrera, F., "Nearestneighbor classification for high-Speed big data streams using spark", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol47, no.10, pp.2727-2739, 2017.
16. Bhagat, R.C. and Patil, S.S., "Enhanced SMOTE algorithm for classification of imbalanced big-data using random forest", In proceedings of International Advance Computing Conference (IACC), IEEE , pp. 403-408. June 2015.
17. Bishwas, A.K., Mani, A. and Palade, V., "Big data classification with quantum multiclass SVM and

quantum one-against-all approach", In proceedings of 2nd International Conference on Contemporary Computing and Informatics, pp. 875-880, December 2016.
18. Ravi, D., Wong, C., Lo, B. and Yang, G.Z., "A deep learning approach to on-node sensor data analytics for mobile or wearable devices", IEEE journal of biomedical and health informatics, vol.21, no.1, pp.56-64, 2017.
19. Liu, B., Blasch, E., Chen, Y., Shen, D. and Chen, G., "Scalable sentiment classification for big data analysis using naive bayes classifier", In proceedings of International Conference on Big Data, IEEE,pp. 99-104, October. 2013.
20. Hassanat, A.,"Norm-Based Binary Search Trees for Speeding Up KNN Big Data Classification Computers", vol.7, no.4, pp.54,2018.
21. Yokoyama, T., Ishikawa, Y. and Suzuki, Y., "Processing all k-nearest neighbor queries in hadoop", In proceedings of International Conference on Web-Age Information Management, pp. 346-351, August 2012.
22. Deng, Y., Ren, Z., Kong, Y., Bao, F. and Dai Q, "A hierarchical fused fuzzy deep neural network for data classification", IEEE Transactions on Fuzzy Systems, vol.25, no.4, pp.1006-1012, 2017.
23. Mikel Elkano, Mikel Galara, Jose Sanza, Humberto Bustinc,"CHI-BD: A Fuzzy Rule-Based Classification System for Big Data classification problems", Fuzzy Sets and Systems, 348, pp.75-101, 2018.
24. Heart disease dataset taken from"http://archive.ics.uci.edu/ml/datasets/heart+disease", accessed on August 2019.
25. Mikel Elkano, Jose Sanz, Edurne Barrenechea, Humberto Bustince, and Mikel Galar, "CFM-BD: a distributed rule induction algorithm for building Compact Fuzzy Models in Big Data classification problems", IEEE Transactions on Fuzzy Systems, 2019.
26. Junhai Zhai, Sufang Zhang, Mingyang Zhang, Xiaomeng Liu, "Fuzzy integral-based ELM ensemble for imbalanced big data classification", Soft computing, February 2018.
27. Jesus Maillo, Sergio Ramirez, Isaac Triguero, Francisco Herrera,"kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data",Knowledge-Based Systems, vol.117, pp.3-15, 2017.
28. Lakshmanaprabu S. K, Shankar K., Ashish Khanna, Deepak Gupta, Joel J. P. C. Rodrigues, Placido R. Pinheiro, and Victor Hugo C. De Albuquerque, "Effective Features to Classify Big Data Using Social Internet of Things", IEEE Access, vol.6, pp.24196-24204, April 2018.
29. Peng Li, Zhikui Chen, Laurence T. Yang, Qingchen Zhang and M. Jamal Deen," Deep Convolutional Computation Model for Feature Learning on Big Data in Internet of Things",IEEE transactions on industrial informatics, vol.14 , no.2 , pp.790-798, Feb. 2018.
30. S. K. Lakshmanaprabu, K. Shankar, M. Ilayaraja, Abdul Wahid Nasir, V. Vijayakumar, and Naveen Chilamkurti, "Random forest for big data classification in the internet of things using optimal features", International Journal of Machine Learning and Cybernetics",pp.1-10, January 2019.
31. Marko Mitic, Najdan Vukovic, Milica Petrovic, Zoran Miljkovic, "Chaotic fruit fly optimization algorithm", Knowledge-Based Systems, vol. 89, pp.446-458, November 2015.
32. Binu, D. and Kariyappa, B.S., "RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits", IEEE Transactions on Instrumentation and Measurement, 2018.
33. Kumar, R. and Kumar, D., "Multi-objective fractional artificial bee colony algorithm to energy aware routing protocol in wireless sensor network", Wireless Networks, vol. 22, no. 5, pp. 1461-1474, 2016.

AUTHORS PROFILE

First Author profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.

Dr. Naeem Thjeel yousir

Minister of the ministry of communication Iraq/now

Phone: Cell phone No# 07901304647

Email: nyousir@gmail.com

Languages: Arabic, English and Turkish

Education:

- **PhD** in Computer Engineering/ Virtual private multimedia network published as SaaS in cloud computing environment. Hacettepe University, Ankara – Turkey.
- **Msc** in computer science 2003**.**
- **Higher Diploma** in informatics, National Center of Computers.
- **BSc** in science of statistics and computers.
- **Diploma** of Turkish Language.

Committees Participations:

- Member of the Committee of the Diwani Order No. 45 of 2016, Committee for the Coordination and Management of Government Activity towards the Establishment of Electronic Governance – General secretariat for the Council of ministers .
- Member of the Committee of Diwani Order No. 216 , Print new maps for all border lines (Land and Sea) – Prime minister office.
- Member of the Committee of Diwani Order No. 345 , Electronic payment and collection workshop – Prime minister office .
- Member of the Committee of Diwani Order No. 526 , Coordinate the process of blocking Malicious and Terrorist pages and websites – Prime Minister office.
- Member of the committee of E-Visa Ministry of Foreign Affairs 2016.
- Member of the committee Criminal justice
- data 2015.
- Head of the Committee of Electronic Payment System 2015.
- Head of the Committee of National Policy for Security and Information Sharing and Communications Protection 2016.

Conferences Participations :

- Attended a TSAG conference for the International Telecommunication Union Switzerland - Geneva 1-5/2/2016.
- Attended a CWC Conference for communications in

Turkey – Istanbul 6-7/10/2015.

- Attended the presentation of IDGA (USA- Washington) 26-28 2017 for Biometric fingerprinting and building a solid information system that matches global standards.
- Leadership course for managers 17-21/1/2016.

Publications:

- Virtual Private Multimedia Network Published As Saas (Software As A Service) in Cloud Computing Environment

N Thjeel Yousir

- Single-Camera Computer Vision Algorithm for Robot Shortest Path Estimator using morphological structuring element with variable sizes

SSJ Sami Hasan, Naeem Th.Yousir
International Journal of Engineering and Technology (UAE) 7 (29),

- Artificial intelligence in the management of the firm

Naeem Th.Yousir, Yaser Saad Zenad, Yaser Issam Hamodi
European Journal of Computer Science and Information Technology 6 (3), 13-23

- Optimization of Mobile User Data Sharing on Secure Cloud

NT Yousir
Journal of Fundamental and Applied Sciences ISSN 1112-9867 10 (5s 2018), 183-186

- Demographic Analysis and Identification of E-Commerce Spending Tendencies

NT Yousir
International Journal of Scientific and Engineering Research (IJSER)

- E-Government for Modern Municipal Corporation

NT Yousir
International Journal of Database Theory and Application 9 (4), 231-238

- Employing" Mobile Agents (MAs)" In the Internet services

Alla Talal Yassin, Naeem Th Yousir AL-Rubaie
Baghdad College of Economic sciences University, 339-361

- Security in E-government

SWEB SITES

- Intelligent Enhancement of Organization Work Flow and Work Scheduling Using Machine Learning Approach Tree Algorithm

NTY Tareq Abed Mohammed, Yaser Issam Hamodi
IJCSNS International Journal of Computer Science and Network Security 18