An Encryption Enabled Metaheuristic Optimization based Feed Forward Neural Network for Cloud based Big Data Environment

Avula Satya Sai Kumar[1], Dr. S. Mohan[2]

[1]*Research Scholar, Department of Computer Science and Engineering, Annamalai University Tamilnadu, India.*

[2]*Assistant Professor, Department of Computer Science and Engineering, Annamalai University Tamilnadu, India.*

[1]*satya.avula@gmail.com*

[2]*mohancseau@gmail.com*

Abstract: ***In recent times, an exponential utilization of information resources and advancements in data analytic tools results in the extended use of big data. Security and privacy are considered as the major issues that existed in the cloud based big data platform, particularly in the healthcare sector. On the other hand, there is a requirement of efficient model for handling big data and has received attention among different researchers. This paper presents Encryption Enabled Metaheuristic Optimization-based Feed Forward Neural Network (EEMO-FFNN) for cloud based big data environment. The presented EEMO-FFNN model intends to perform secure communication and effective big data analytics in the healthcare environment. The EEMO-FFNN model initially enables the augmentation of patient data using SMOTE for the generation of big data. Next, secure data transmission from the source to cloud server takes place using Elliptic Curve Cryptography (ECC) based encryption technique. Besides, the MO-FFNN model based data classification process is performed on the Hadoop ecosystem to identify the existence of the disease. In order to adjust the weight and bias parameters involved in the FFNN model, the Salp Swarm Optimization (SSA) algorithm is applied. An extensive set of simulations were performed to ensure the effectual outcome of the EEMO-FFNN model and the results are examined under distinct aspects.***

Keywords: Cloud computing, Big data, Healthcare, Security, Encryption, Data analytics

1. INTRODUCTION:

Big data defines a massive quantity of data where the organizations need to process, examine and archive [1]. The intensified usage of information resources and the requirement of progressive data processing techniques results in the generation of big data. A brief discussion of big data characteristics such as variety, volume, security, and privacy are provided in [2]. Big data analytics include service tools, like Hadoop Distributed File System (HDFS) that offers support to manage, store massive quantity of data, fasten decision making, and reduce human errors. The HDFS is considered as the commonly available tool which offers redundancy, parallel processing, trustworthiness, scalability, and distributed architecture

system [3] are realized for handling distinct kinds of big data namely structured, semi-structured and unstructured. Furthermore, Hadoop MapReduce Job-Scheduling technique performs clustering of big data in a distributed platform. Additionally, big data analytics offer chances to resolve distinct data security issues through Hadoop technologies and HDFS. The data value which is produced from the big data by the analytics stage is highly essential. But the exhaustive utilization of big data poses several new security issues, mainly in the computation of secret data like trading details of an organization, medical reports of the patients, etc.

For utilizing the effective benefits of big data, a major intention lies is to save the secret data from possible risk factors. The significance of cybersecurity in the existence of big data is deliberated in [4], where big data is assumed as a targeting point for attackers to gain large amount of data. But the classical security techniques are not sufficient to protect big data mobility. Therefore, safeguarding big data is a major issue which necessitates recent techniques to achieve data security. The Fog computing [5] ropes a controller to handle the secret data. The communication of big data over the cloud and fog offers the computation of storage devices data type which necessitate few clustering models which arranges the data based on the heat and victimize big data depending upon the delay sensitivity and data temperature. But it becomes essential to apply security methods for the prevention of any data threat or potential risk factors. In recent times, several research works have been concentrated on security process of big data. But numerous security schemes are developed for protecting the predefined data over risks which are not adequate for big data and it goes beyond the computation abilities of the available databases.

Private data is a valuable mark for threats which harmfully disturb the trustworthiness and creditability of the organization. For instance, big data might generate security risk to user emails through the creation of websites for phishing depending upon the email behavior and interest. Big data security remains challenging which affecting the cloud platform and damages the reputation of the organization. Security and privacy are the crucial needs available to store, manage, analyze, and transmit big data [6]. Any inclusive big data security solutions need to fulfill data confidentiality, integrity, and availability. Big data security issues are treated in the design of big data environment and protect big data through storing or computation, whereas many encryption schemes are designed for prohibiting illegal operators to access big data [25-30].

This paper presents Encryption Enabled Metaheuristic Optimization-based Feed Forward Neural Network (EEMO-FFNN) for cloud based big data environment. The presented EEMO-FFNN model intends to perform secure data transmission and effective big data analytics in the healthcare environment. The EEMO-FFNN model initially enables the augmentation of patient data using SMOTE for the generation of big data. Subsequently, secure data transmission from the source to cloud server takes place using Elliptic Curve Cryptography (ECC) based encryption technique. Besides, the MO-FFNN model based data classification process is performed on the Hadoop ecosystem to identify the existence of the disease. For adjusting the weight and bias parameters involved in the FFNN model, the Salp Swarm Optimization (SSA) algorithm is applied. A detailed experimental results analysis is performed to verify the goodness of the proposed EEMO-FFNN model.

2. LITERATURE SURVEY

Big data privacy is considered as the two complementary features mainly, data security and access control [7], where the major big data issues exist are data management and classification

[8]. Big data security handling can be managed by the Kerberos management strategy which secures the data at varying stages of transmission, authentication, and storage [9]. The Kerberos is intended to authenticate data, transport secure layer for data transmission, and encryption. Though Kerberos protects the data, different sources of big data pose distinct security and governance rules making the presented solution difficult to implement. Big data security policies are recommended throughout functions like Know, Prevent, Detect, Respond, and Recover [10] for successful attack detection in such a way that the compromising of data can be found and resolved. Nonetheless, any possible risk factors like breach and attack on sensitive data remains a major issue.

A big data security model including the access control features of logging model is presented [11]. The access control characteristics are utilized to create the big data file. But the major limitation is the tediousness of computing the user behavior and overlapping of client duties over the big data. Additionally, a model for selecting the fields of big data needs to be protected in [12], where big data has considered every object has the individual characteristics which are sorted based on the significance level. However, none of the ranking policies are adapted to the characteristics of the unstructured big data.

Securing Big data while transmitting it over the cloud platform can be attained through the secure socket layer SSL link which is carried out among the transmitting and receiving cloud name nodes [13]. Hash values are utilized for the creation of a sequence of encrypted tickets which are utilized in the data transmission process. But, the issue with the individual certificate authority affects the user verification at both clouds ends enabling the hacker for ticket collection. The existing data mining models also protect the secrecy of the data by the substitution of the values of sensitive characteristics and forbidding the revelation of secret data. But they are not adequate to handle big data and do not resolve security issues. Besides, data mining techniques also face the issue of handling unstructured data types.

3. THE PROPOSED EEMO-FFNN MODEL

Fig. 1 demonstrates the working procedure involved in the EEMO-FFNN model. The figure portrayed that the medical data from the patients are initially augmented by the SMOTE technique for the generation of big data. Followed by, the generated big data is securely transmitted to the cloud server using ECC based encryption technique. Then, the cloud server decrypts the data and performs MO-FFNN based classification process at the Hadoop ecosystem. Finally, the optimal parameter adjustment of the FFNN takes place via SSA for improved performance.

A. Synthetic Data Generation using SMOTE Technique

At the beginning stage, the medical records of the patients are collected and the data size is augmented by the use of Synthetic Minority Oversampling Technique (SMOTE). It is a commonly used oversampling technique introduced by Chawla et al. [14] and works in feature space instead of data space. By the use of this method, the instance count from the minority class of the actual dataset is raised by the generation of synthetic samples resulting in a broad decision region for the minority class whereas the naive oversampling with replacement results in the decision region of the minority class to be particular.

The new synthetic instances are generated by the representation of two variables namely oversampling rate (%) and nearest neighbor count (k).

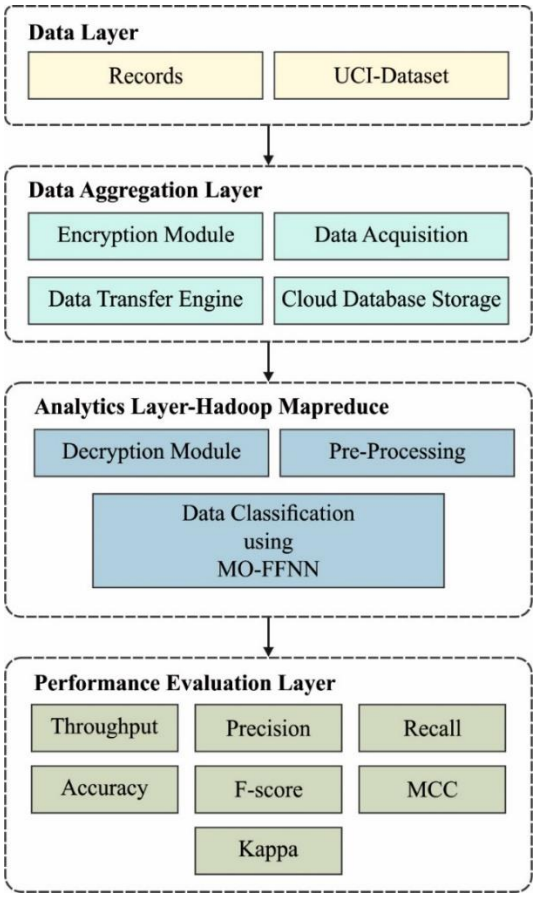


Fig. 1 Block diagram of EEMO-FFNN model

The processes involved in the generation of new synthetic samples for continuous features are created are given in the following:

Step 1: Compute the distance among the feature vectors of the minority class with a k nearest neighbors.

Step 2: Multiplication of the distance achieved in the previous step with an arbitrary number from the range of [0, 1].

Step 3: Add the values got from Step 2 to the feature values of the actual feature vectors. And, the new feature vectors can be defined as

$$x_n = x_o + \delta \cdot (x_{oi} - x_o) \quad (1)$$

where x_n signifies a new synthetic sample, x_o is represented as a feature vector of all instances in the minority class, x_{oi} is the ith designated nearest neighbor of x_o, and δ is an arbitrary number lies in the range of [0, 1]. For instance, for a provided $\beta\%= 900\%$ and $k = 5$, it is needed to produce 9 new synthetic samples for the actual sample. The previous steps are iterated for a set of 9 times. Since a new sample is generated at every time, one of the 5 nearest neighbors of x_o is arbitrarily selected [15]. Furthermore, the synthetic generation of instances for nominal features take place as follows:

Step 1: Find the majority votes among the features under attention and its k nearest neighbors for the minor feature value.

Step 2: Allocate the gained values to the innovative synthetic minority class samples.

For instance, for a collection of characteristics of an instance $\{A, B, C, D, E\}$ and the two adjacent neighbors have the set of features as $\{A, F, C, G, N\}$ and $\{H, B, C, D, N\}$, the new synthetic sample has generated a set of the features, that is defined by $\{A, B, C, D, N\}$.

B. ECC based Encryption Technique

For the secure transmission of synthetically generated medical data, the ECC technique is applied for the encryption of data prior to data communication process. Generally, elliptic curve (EC) is employed for the public key system and it is noted that the elliptic curve is not equivalent to the ellipse. The ellipse includes two symmetrical lines such as vertical and horizontal. At the same time, the elliptical curve includes a single horizontal line. Based on [16], the ECC is defined as follows:

$$y^2 = x^3 + ax + b \tag{2}$$

The above EC equation is called the Weierstrass equation, where a and b are constant values which can compute the shape of the curve, as given below:

$$4a^3 + 27b^2 \neq 0 \tag{3}$$

The ECC formula for a finite field Z_p with p primes is defined below:

$$y^2 = (x^3 + ax + b)\ mod\ p \tag{4}$$

Where two major functions are included namely point addition and point doubling. The former function takes place when two distinct points are added. Besides, the coordinate outcome is the third point over by the line that starts from the first two coordinate points. For performing this function, the gradient (λ) can be represented as follows:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}\ mod\ p \tag{5}$$

Subsequently, the third search point can be defined by

$$x_3 = \{\lambda^2 - x_1 - x_2\} mod\ p \tag{6}$$
$$y_3 = \{\lambda(x_1 - x_3) - y_1\}\ mod\ p$$

where (x_1, y_1) is the first point and (x_2, y_2) is the second point.

Point doubling is determined in case every point has identical coordinate points. The resultant coordinate points are the points where the line goes through the *tan* position over the primary coordinate points. The process of point doubling does not vary from the point addition process which is started with the gradient (λ) search as given below.

$$\lambda = \frac{3{x_1}^2 + a}{2y_1} \tag{7}$$

It is generated from the implied reduction of the EC formula in Equation 1, trailed by the searching process of x_3 and y_3.

$$x_3 = \{\lambda^2 - 2x_1\}\ mod\ p \tag{8}$$
$$y_3 = \{\lambda(x_1 - x_3) - y_1\}\ mod\ p$$

where (x_1, y_1) is the source coordinate. When P is a point of the elliptic curve, then the multiplication of a positive integer k to P is represented as the point addition which is iterated for k times. The multiplication is defined by $kP = P + P + P + P + \cdots + P$. If $x_1 = x_2$ and $y_1 = y_2 = 0$ or $x_1 = x_2$ and $y_1 = -y_2$, at that time, the points are intersected at ∞, representing 0.

C. Hadoop Ecosystem

For big data management, Hadoop Ecosystem with the components is mainly applied [17]. For a distributed setting, Hadoop is a type of openly accessible structure that allows the stakeholder to store and analyze big data over the clustering using easy programming tools.

The major components of Hadoop are MapReduce, HDFS, and Hadoop YARN. Based on Google File System (GFS), the HDFS is defined as a model of master/slave network where every master comprises a set of data nodes. It is named as actual data and dissimilar name nodes are known as metadata. To offer maximum scalability over 1000's of Hadoop clusters, Hadoop Map Reduce is utilized that is commonly named as programming framework at the heart of the Apache Hadoop. For processing large quantity of data over numerous clusters, MapReduce is applied. Two important stages present in the MapReduce job processing such as Reduce and Map phases. These phases hold a pair of key - value as input and output; explicitly, in the file system, the output and input of the jobs are protected.

Finally, Hadoop YARN is a model widely utilized to manage clusters. By looking at the attained knowledge from the initial Hadoop creation, it is defined as the next Hadoop generation that performs as a major characteristic. For offering privacy, trustworthiness, and data governing tools, YARN acts as a fundamental model and resource manager. For big data process, other tools and components are placed on the Hadoop structure.

D. FFNN based Classification

FFNN is a simpler kind of ANN which comprises a set of computing components known as "neurons". Here, the data moves over an individual direction, forwards from the input to output layers via the hidden layer. Here, every individual neuron determines the total of the input weights at the existence of the bias and the total is fed into the activation function (like sigmoid function) so that the outcome can be attained. It is defined in Eqs. (9)-(10):

$$h_{\mathrm{j}} = \sum_{i=1}^{R} iw_{j,i} x_i + hb_j, \tag{9}$$

where $iw_{j,i}$ is the weights linked among the neurons $i = (1, 2, \dots, R)$ and $\mathrm{j} = (1,2, \dots, N)$, $\mathrm{hb_j}$ is a bias in hidden layer, R is the total neuron count in input layer, and x_i is the equivalent input data.

At this point, the S-shaped curved sigmoid function is employed as the activation function as defined below

$$f_{(x)} = \frac{1}{1 + e^{-x}}. \tag{10}$$

So, the outcome of the neurons in the hidden layer is represented by

$$\mathrm{ho}_j = f_j(h_j) = \frac{1}{\left(1 + e^{-h_j}\right)}. \tag{11}$$

At the final layer, the outcome of the neuron can be represented as follows

$$y_k = f_k\left(\sum_{j=1}^{N} \mathrm{hw}_{k,j}\, \mathrm{ho}_j + \mathrm{ob}_k\right), \tag{12}$$

where $\mathrm{hw}_{k,j}$ are the weights linked among the neurons $\mathrm{j} = (1, 2, \dots, N)$ and $k = (1,2, \dots, S)$, ob_k is a bias in the final layer, N and S denotes the neuron count in the hidden and output layers respectively. The training procedure is carried out for weight and bias adjustment until few error criteria are satisfied [18]. The design of FFNN seems to be difficult due to the impact of several components on the training performance like neuron count in hidden layer, link among neurons, error function, and activation function.

E. SSA based Parameter Optimization of FFNN Model

The salp is a transparent kind of jellyfish. At present, it mostly exists in the sea nearby Oceania. The researchers have exposed that the predation method of the sea squirt is a chain-similar performance that depends on the chain system of the group to optimal food. Mirjalili [19] presents the SSA stimulated by the predation system of the salps. The SSA depends on chain performance for determining the better solution. In the SSA model, the population is separated into a leader that is at the front of the chain, and followers; the leader guides the salp chain and followers follow in sequence. During this procedure of the salp, the leader maintains the population to better forage, and the follower follows the present salp and broadcasting feed signals for continuing the suitability of the population, and avoid population falling as to the local better solution. Afterward, an important step of the SSA is analyzed. Assume the position vector of all the salps be an $N \times D$ dimensional, where N is number of search agents and D signifies the dimension of continuous solution space. Therefore, the vector position of the population is demonstrated by a multi-dimensional matrix, as depicted in Eq. (13):

$$X_j^i = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_{\mathrm{D}}^1 \\ X_1^2 & X_2^2 & \cdots & X_{\mathrm{D}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ X_1^N & X_2^N & \cdots & X_{\mathrm{D}}^N \end{bmatrix} \tag{13}$$

The subsequent formula referred to upgrading the leader position is illustrated as [20]:

$$X_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 \geq 0.5 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 < 0.5 \end{cases} \tag{14}$$

where X_j^1 is the location of primary generation leader of the population in the j_{th} dimension, ub_j refers the upper bound of j_{th} dimension search space, lb_j denotes the lower bound of jth dimension search space, c_2 as well as c_3 are the arbitrary numbers in the interval [0, 1]. A coefficient c_1 is the vital parameter in the iterative model of the techniques due to balances among exploration as well as exploitation propensities, and it is provided as:

$$c_1 = 2e^{-\left(\frac{4l}{l_{\max}}\right)^2} \tag{15}$$

where l is the present iteration of the technique and $l_{\max}$ is the maximal count of iterations.

The slap position of follower is updated based on Newton's law of motion, and its equation is shortly explained as follows:

$$X_j^i = \frac{1}{2}at^2 + v_0 t \tag{16}$$

where $i \geq 2$, X_j^i implies the location of i_{th} follower salp in j_{th} dimension, t is time, v_0 refers the primary speed, and $a = \frac{v_1}{v_0}$ where $v = \frac{x - x_0}{t}$.

In joint with the standard Newton, the time interval in the technique for upgrading the position time is equivalent to 1, and assuming $v_0 = 0$, this formula is written as follows:

$$X_j^i = \frac{1}{2}\left(X_j^i + X_j^{i-1}\right) \tag{17}$$

where $i \geq 2$, and X_j^i implies the position of i_{th} follower salp in j_{th} dimension. Based on Eq. (14) and Eq. (17) are created the salp chains.

At the time of designing FFNN, layer count and neuron count in the layers are needed to be determined. When the hidden layer and node count are found to be high, then the network becomes complex. Here, the input and output neuron count in the MLP model is problem-

specific and the hidden node count is determined based on the Kolmogorov theorem [21]: Hidden $= 2 \times$ Input $+1$.

By the use of SSA for optimal selection of weight and bias in the FFNN, the dimension of the salps are treated as D, and is given by

$$D = (Input \times Hidden) + (Hidden \times Output) + \text{Hidden}_{\text{bias}} + \text{Output}_{\text{bias}}, \quad (18)$$

where Input, Hidden, and Output refers to the input, hidden, and output neuron count of the FFNN. In addition, $\text{Hidden}_{\text{bias}}$ and $\text{Output}_{\text{bias}}$ indicates the bias in the hidden and output layers.

In SSA, the salps are determined based on the fitness value. The validation is performed by feeding the vector of weight and bias to the FFNN and the Mean Squared Error (MSE) criteria are determined using the predictive outcome of the NN by the use of training data. Using the incessant iterations, the optimum solutions are attained that is considered as the weight and bias of the NN. The MSE is defined as follows.

$$MSE = \frac{1}{R}\sum_{i=1}^{R}(y - \hat{y})^2. \quad (19)$$

where y and $\hat{y}$ are the real and the assessed values depending upon the presented method and R indicates the sample count in the training data. Based on [22], the weight and bias of FFNNs for all agents in the metaheuristic algorithm can be encoded and defined in the way of a vector, matrix, or array. Here, vector encoding technique is used. At the time of initialization, $X = (X_1, X_2, \ldots, X_N)$ is fixed based on the N salps. Every salp $X_{\text{i}} = \{iw, hw, hb, ob\}$ (i $= 1,2, \ldots, N$) signifies a whole set of FFNN weight and bias that is transformed into a single vector of real numbers. To design a classifier model, accuracy is used along with the MSE metric. It determines the capability of the classification model by the generation of precise performance that can be determined as given below:

$$Accuracy = \frac{\widetilde{N}}{N}, \quad (20)$$

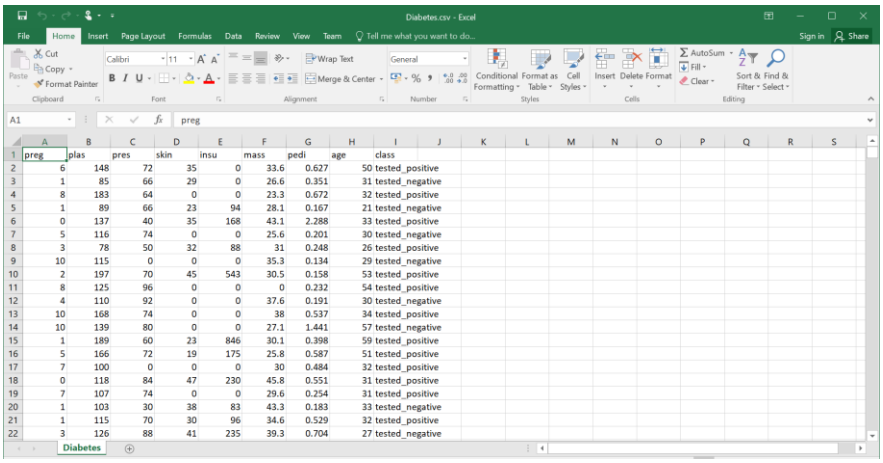
where $\widetilde{N}$ indicates the total of properly categorized samples by the classification model and N is the total number of samples in the dataset.

4. PERFORMANCE VALIDATION

To validate the effectual result analysis of the EEMO-FFNN model, a set of simulations was performed on PIMA Indians Diabetes dataset. The dataset comprises a number of 6580 synthetic instances and the actual number of instances remains to be 768. Besides, it contains the samples with the existence of 8 features with 2 class labels. The details of the dataset are given in Table 1 and some sample illustration of the dataset is given in Fig. 2.

Table 1: Dataset Description

Description	**Pima Indian Diabetes**
Number of Instances	6580 (768)
Number of Attributes	8
Number of Class	2
Percentage of Positive Samples	3340 (268)
Percentage of Negative Samples	3240 (500)
Data sources	[23]



preg	plas	pres	skin	insu	mass	pedi	age	class
6	148	72	35	0	33.6	0.627	50	tested_positive
1	85	66	29	0	26.6	0.351	31	tested_negative
8	183	64	0	0	23.3	0.672	32	tested_positive
1	89	66	23	94	28.1	0.167	21	tested_negative
0	137	40	35	168	43.1	2.288	33	tested_positive
5	116	74	0	0	25.6	0.201	30	tested_negative
3	78	50	32	88	31	0.248	26	tested_positive
10	115	0	0	0	35.3	0.134	29	tested_negative
2	197	70	45	543	30.5	0.158	53	tested_positive
8	125	96	0	0	0	0.232	54	tested_positive
4	110	92	0	0	37.6	0.191	30	tested_negative
10	168	74	0	0	38	0.537	34	tested_positive
10	139	80	0	0	27.1	1.441	57	tested_negative
1	189	60	23	846	30.1	0.398	59	tested_positive
5	166	72	19	175	25.8	0.587	51	tested_positive
7	100	0	0	0	30	0.484	32	tested_positive
0	118	84	47	230	45.8	0.551	31	tested_positive
7	107	74	0	0	29.6	0.254	31	tested_positive
1	103	30	38	83	43.3	0.183	33	tested_negative
1	115	70	30	96	34.6	0.529	32	tested_positive
3	126	88	41	235	39.3	0.704	27	tested_negative

Fig. 2 Sample Dataset

Table 2 and Fig. 3 investigates the throughput analysis of the EEMO-FFNN model under the existence of with and without Hadoop. The figure states that the throughput of the EEMO-FFNN model gets increased by the inclusion of Hadoop structure. For instance, on the applied dataset of 1000 instances, the EEMO-FFNN model with Hadoop achieves a higher throughput of 5600Kbps whereas the EEMO-FFNN model without Hadoop results in a reduced throughput of 1500Kbps. At the same time, on the applied dataset of 2000 instances, the EEMO-FFNN method with Hadoop attains a maximum throughput of 6300Kbps whereas the EEMO-FFNN technique without Hadoop results in a reduced throughput of 2500Kbps. Likewise, on the applied dataset of 3000 instances, the EEMO-FFNN approach with Hadoop obtains a superior throughput of 7200Kbps whereas the EEMO-FFNN technique without Hadoop results in a lesser throughput of 3200Kbps. Similarly, on the applied dataset of 4000 instances, the EEMO-FFNN manner with Hadoop reaches a maximum throughput of 7800Kbps whereas the EEMO-FFNN methodology without Hadoop results in a minimum throughput of 4100Kbps. Along with that, on the applied dataset of 5000 instances, the EEMO-FFNN model with Hadoop obtains a higher throughput of 8100Kbps whereas the EEMO-FFNN method without Hadoop outcomes to a reduced throughput of 5300Kbps. Simultaneously, on the applied dataset of 6000 instances, the EEMO-FFNN model with Hadoop obtains a superior throughput of 8400Kbps whereas the EEMO-FFNN method without Hadoop results in a minimum throughput of 6200Kbps.

Table 2: Performance Analysis of Throughput (Kbps) on With and Without Hadoop

Dataset Size (Instances)	**Without Hadoop**	**With Hadoop**
1000	1500	5600
2000	2500	6300
3000	3200	7200
4000	4100	7800
5000	5300	8100
6000	6200	8400

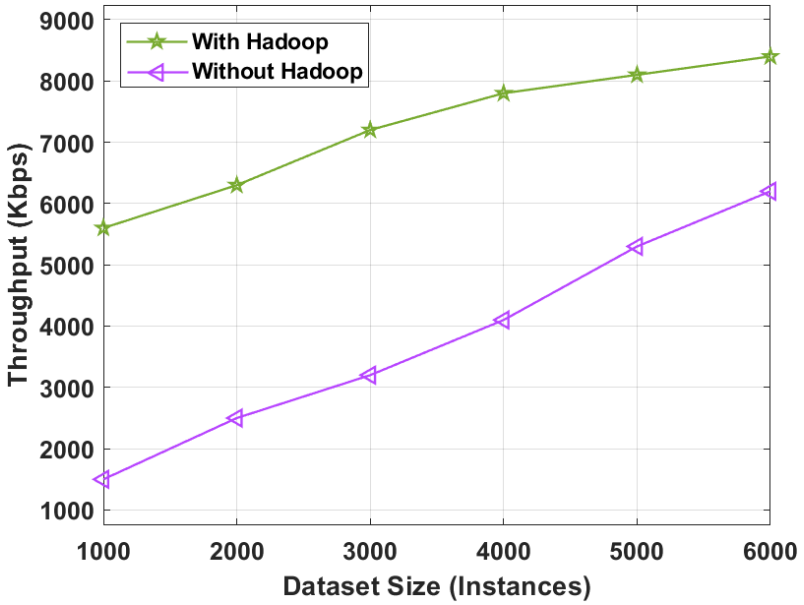


Fig. 3 Throughput analysis of EEMO-FFNN model under with and without Hadoop

Table 3 provides an elaborative comparative study of the EEMO-FFNN model with existing methods interms of distinct measures.

Table 3: Performance Evaluation of Different Classifiers with Proposed EEMO-FFNN Method on Applied Dataset

Methods	**Precision**	**Recall**	**Accuracy**	**F-score**	**MCC**	**Kappa**
EEMO-FFNN	0.996	0.978	0.984	0.989	0.980	0.961
AKM+GBT	0.992	0.975	0.978	0.983	0.950	0.950
BKM+GBT	0.918	0.909	0.887	0.913	0.740	0.750
Logistic Regression	0.880	0.793	0.772	0.834	0.480	0.473
Voted Perceptron	0.924	0.680	0.668	0.784	0.170	0.135
LogitBoost	0.846	0.776	0.741	0.810	0.410	0.407
Decision Tree	0.814	0.790	0.738	0.802	0.420	0.416

Fig. 4 offers precision and recall analysis of the EEMO-FFNN model with existing methods. The figure demonstrated that the DT model has demonstrated ineffectual outcome with the least precision and recall of 0.814 and 0.79. Also, the LogitBoost model has exhibited slightly improved precision and recall of 0.846 and 0.776. Besides, the LR model has tried to exhibit moderate outcome with the precision and recall of 0.88 and 0.793. Followed by, the BKM-GBT model has resulted in a certainly improved outcome with the precision and recall of 0.918 and 0.909. Furthermore, the Voted Perceptron model has accomplished a reasonable performance with the precision and recall of 0.924 and 0.68. Moreover, the AKM-GBT model has showcased a competitive outcome with the precision and recall of 0.992 and 0.975. At last, the EEMO-FFNN model has led to a maximum outcome with the precision and recall of 0.996 and 0.978.

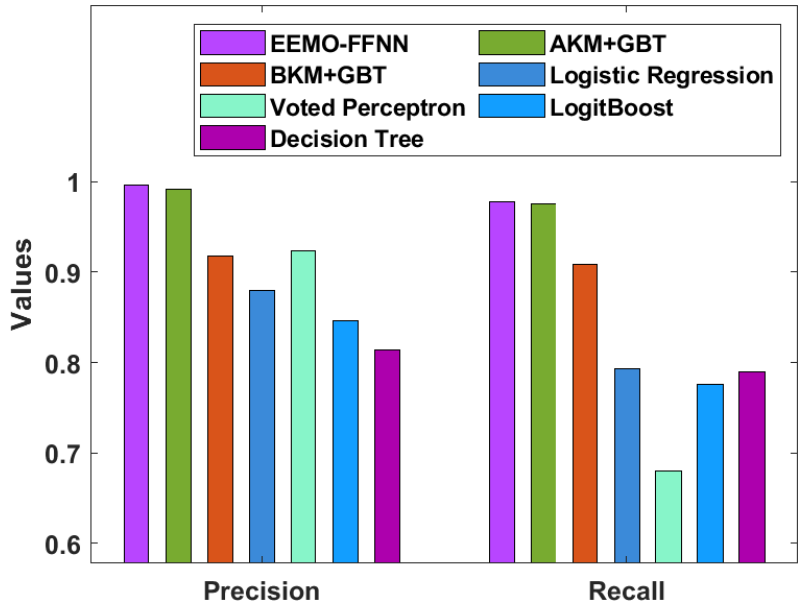


Fig. 4 Precision and recall analysis of EEMO-FFNN model

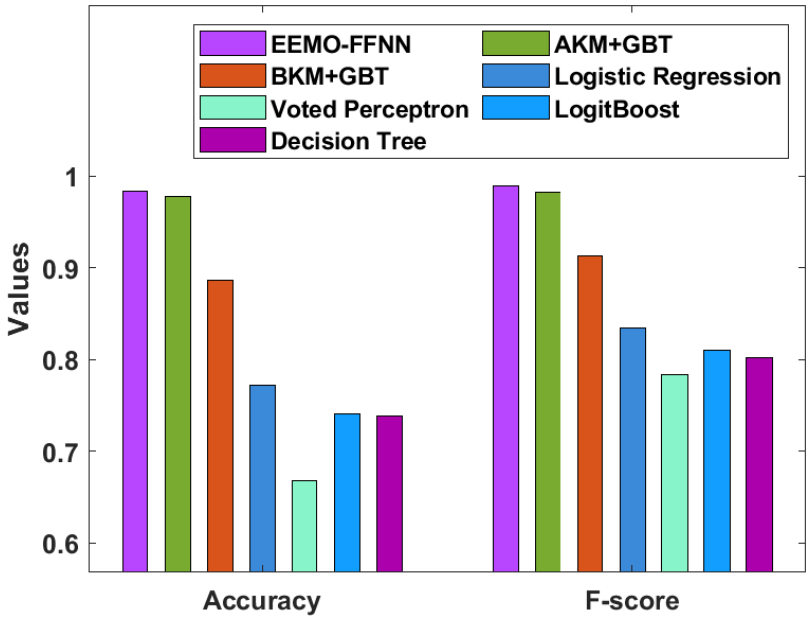


Fig. 5 Accuracy and F-score analysis of EEMO-FFNN model

Fig. 5 provides accuracy and f-score analysis of the EEMO-FFNN method with existing techniques. The figure has shown that the Voted Perceptron model has exhibited ineffectual results with the least accuracy and f-score of 0.668 and 0.784. Also, the DT model has exhibited slightly higher accuracy and f-score of 0.738 and 0.802. In line with, the LogitBoost model has tried to show moderate outcome with the accuracy and f-score of 0.741 and 0.810. Followed by, the LR technique has resulted in a certainly increased outcome with the accuracy and f-score of 0.772 and 0.834. Furthermore, the BKM+GBT model has accomplished a reasonable performance with the accuracy and f-score of 0.887 and 0.913. In addition, the AKM-GBT model has outperformed a competitive outcome with the accuracy and f-score of 0.978 and 0.983. At last, the EEMO-FFNN methodology has led to a superior result with the accuracy and f-score of 0.984 and 0.989.

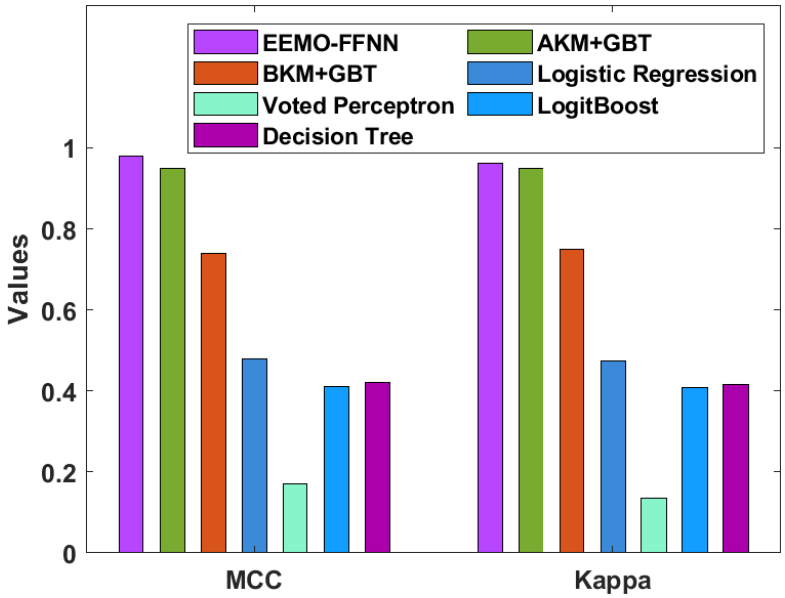


Fig. 6 MCC and kappa analysis of EEMO-FFNN model

Fig. 6 gives an MCC and kappa analysis of the EEMO-FFNN technique with existing models. The figure exhibited that the Voted Perceptron model has outperformed ineffectual outcome with the minimum MCC and kappa of 0.170 and 0.135. Along with that, the LogitBoost model has showcased somewhat superior MCC and kappa of 0.410 and 0.407. Besides, the DT model has tried to exhibit moderate results with the MCC and kappa of 0.420 and 0.416. Likewise, the LR model has resulted in a certainly improved outcome with the MCC and kappa of 0.480 and 0.473. Furthermore, the BKM+GBT approach has accomplished a reasonable performance with the MCC and kappa of 0.740 and 0.750. But, the AKM-GBT model has showcased a competitive result with the MCC and kappa of 0.950 and 0.950. At last, the presented EEMO-FFNN technique has led to a higher outcome with the MCC and kappa of 0.980 and 0.961.

Table 4 and Fig. 7 provides a comparative analysis of the EEMO-FFNN model with recent methods interms of accuracy [24]. The figure outperformed that the KNN model has demonstrated ineffectual outcome with the minimal accuracy of 0.676. Similarly, the CART and ELM models have exhibited slightly higher accuracy of 0.728 and 0.757. Likewise, the SGD and MLP models have tried to show moderate outcome with the accuracy of 0.766 and 0.819. Followed by, the Hybrid model and J48 (pruned) models have resulted in a certainly increased outcome with the accuracy of 0.845 and 0.893. Furthermore, the AMMLP and HPM models have accomplished a reasonable performance with the accuracy of 0.899 and 0.924. Besides, the FNCA and KM+LR models have outperformed even superior results with the accuracy of 0.945 and 0.954. Moreover, the BKM+GBT and AKM+GBT models have exhibited a competitive outcome with the accuracy of 0.887 and 0.978. At last, the proposed EEMO-FFNN model has led to a superior outcome with an accuracy of 0.984.

Table 4: Result Analysis with Recent Methods on Proposed EEMO-FFNN for Applied Dataset in terms of Accuracy

Methods	**Accuracy**
Proposed EEMO-FFNN	0.984
AKM+GBT	0.978

BKM+GBT	0.887
KM+LR	0.954
FNCA	0.945
HPM	0.924
AMMLP	0.899
J48 (pruned)	0.893
Hybrid Model	0.845
MLP	0.819
SGD	0.766
ELM	0.757
CART	0.728
KNN	0.676

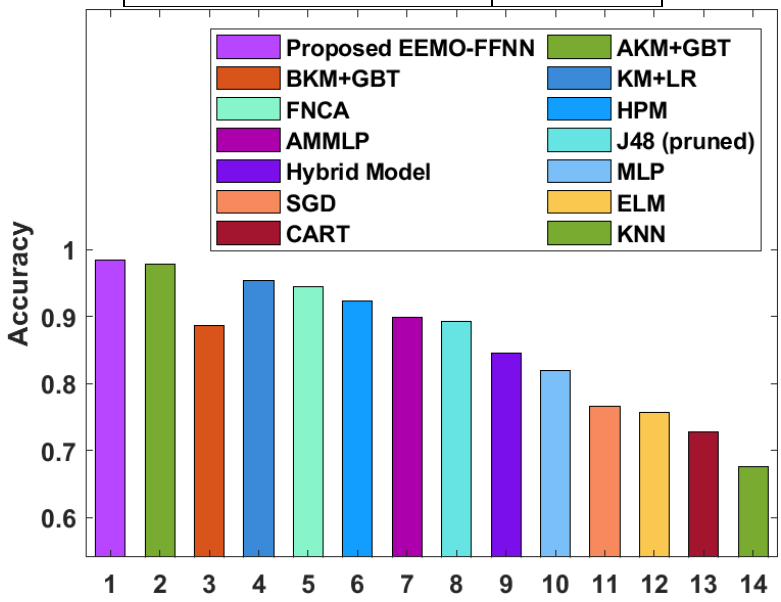


Fig. 7 Accuracy analysis of EEMO-FFNN model with recent methods

5. CONCLUSIONS

This paper has developed a novel EEMO-FFNN model for security and classification in cloud based big data environment. The presented EEMO-FFNN model intends to perform secure data transmission and effective big data analytics in the healthcare environment. Primarily, the medical data from the patients are initially augmented by the SMOTE technique for the generation of big data. Followed by, the generated big data is securely transmitted to the cloud server using ECC based encryption technique. Then, the cloud server decrypts the data and performs MO-FFNN based classification process at the Hadoop ecosystem. Finally, for adjusting the weight and bias parameters involved in the FFNN model, the SSA algorithm is applied. A detailed experimental results analysis is performed to verify the goodness of the

proposed EEMO-FFNN model. In future work, the performance of the EEMO-FFNN model can be improved using advanced encryption and deep learning techniques.

6. REFERENCES

[1] M. Paryasto, A. Alamsyah, B. Rahardjo, and M. Kuspriyanto, ‘‘Bigdata security management issues,’’ in Proc. 2nd Int. Conf. Inf. Commun. Technol. (ICoICT), May 2014, pp. 59–63.
[2] A. K. Tiwari, H. Chaudhary, and S. Yadav, ‘‘A review on big data and its security,’’ in Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS), 2015, pp. 1–5.
[3] J. V. Gautam, H. B. Prajapati, V. K. Dabhi, and S. Chaudhary, ‘‘A survey on job scheduling algorithms in big data processing,’’ in Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT), Mar. 2015, pp. 1–11.
[4] T. Mahmood and U. Afzal, ‘‘Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools,’’ in Proc. 2nd Nat. Conf. Inf. Assurance (NCIA), 2013, pp. 129–134.
[5] A. Khalid and M. Shahbaz, ‘‘Adaptive deadline-aware scheme (ADAS) for data migration between cloud and fog layers,’’ KSII Trans. Internet Inf. Syst., vol. 12, no. 3, pp. 1002–1015, 2018.
[6] E. Bertino and E. Ferrari, ‘‘Big data security and privacy,’’ in A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years. Cham, Switzerland: Springer, 2018, pp. 425–439.
[7] V. Gadepally et al., ‘‘Computing on masked data to improve the security of big data,’’ in Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST), Apr. 2015, pp. 1–6.
[8] K. S. Arvind and R. Manimegalai, ‘‘Secure data classification using superior naive classifier in agent based mobile cloud computing,’’ Cluster Comput., vol. 20, no. 2, pp. 1535–1542, 2017.
[9] N. Chaudhari and S. Srivastava, ‘‘Big data security issues and challenges,’’ in Proc. Int. Conf. Comput., Commun. Autom. (ICCCA), 2016, pp. 60–64.
[10] E. Damiani, ‘‘Toward big data risk analysis,’’ in Proc. IEEE Int. Conf. Big Data (Big Data), Oct./Nov. 2015, pp. 1905–1909.
[11] A. Gupta, A. Verma, P. Kalra, and L. Kumar, ‘‘Big data: A security compliance model,’’ in Proc. Conf. IT Bus. Ind. Government (CSIBIG), 2014, pp. 1–5.
[12] S.-H. Kim, N.-U. Kim, and T.-M. Chung, ‘‘Attribute relationship evaluation methodology for big data security,’’ in Proc. Int. Conf. IT Converg. Secur. (ICITCS), 2013, pp. 1–4.
[13] Q. Shen, L. Zhang, X. Yang, Y. Yang, Z. Wu, and Y. Zhang, ‘‘SecDM: Securing data migration between cloud storage systems,’’ in Proc. IEEE 9th Int. Conf. Dependable, Autonomic Secure Comput. (DASC), Dec. 2011, pp. 636–641.
[14] N.V.Chawla,K.W.Bowyer, L.O.Hall,W.P.Kegelmeyer, SMOTE: Syntheticminority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
[15] Wang, K.J., Makond, B., Chen, K.H. and Wang, K.M., 2014. A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. Applied Soft Computing, 20, pp.15-24.
[16] Natanael, D. and Suryani, D., 2018. Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC). Procedia Computer Science, 135, pp.283-291.
[17] Selvi, R.T. and Muthulakshmi, I., 2020. Modelling the map reduce based optimal gradient boosted tree classification algorithm for diabetes mellitus diagnosis system. Journal of Ambient Intelligence and Humanized Computing, pp.1-14.

[18] Wu, H., Zhou, Y., Luo, Q. and Basset, M.A., 2016. Training feedforward neural networks using symbiotic organisms search algorithm. Computational intelligence and neuroscience, 2016.
[19] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, ‘‘Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,’’ Adv. Eng. Softw., vol. 114, pp. 163–191, Dec. 2017.
[20] Chen, R., Dong, C., Ye, Y., Chen, Z. and Liu, Y., 2019. QSSA: Quantum Evolutionary Salp Swarm Algorithm for Mechanical Design. IEEE Access, 7, pp.145582-145595.
[21] R. Hecht-Nielsen, “Kolmogorov’s mapping neural network existence theorem,” in Proceedings of the IEEE 1st International Conference on Neural Networks, vol. 3, pp. 11–13, IEEE Press, San Diego, Calif, USA, 1987
[22] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training,” Applied Mathematics and Computation, vol. 185, no. 2, pp. 1026–1037, 2007
[23] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[24] Selvi, R.T. and Muthulakshmi, I., 2020. Modelling the map reduce based optimal gradient boosted tree classification algorithm for diabetes mellitus diagnosis system. Journal of Ambient Intelligence and Humanized Computing, pp.1-14
[25] K. Shankar and P. Eswaran. “RGB Based Multiple Share Creation in Visual Cryptography with Aid of Elliptic Curve Cryptography”, China Communications, Volume. 14, Issue. 2, page(s): 118-130, February 2017.
[26] Muzafer H. Saračević, Saša Z. Adamović, Vladislav A. Miškovic, Mohamed Elhoseny, Nemanja D. Maček, Mahmoud Mohamed Selim, K. Shankar, “Data Encryption for Internet of Things Applications Based on Catalan Objects and Two Combinatorial Structures”, IEEE Transactions on Reliability, Page(s): 1 – 12, August 2020.
[27] Mohamed Elhoseny and K. Shankar, “Reliable Data Transmission Model for Mobile Ad Hoc Network Using Signcryption Technique”, IEEE Transactions on Reliability, Volume. 69, Issue. 3, Page(s): 1077-1086, September 2020. https://doi.org/10.1109/TR.2019.2915800
[28] K. Shankar, Mohamed Elhoseny, R. Satheesh Kumar, S. K. Lakshmanaprabu, Xiaohui Yuan, “Secret image sharing scheme with encrypted shadow images using optimal homomorphic encryption technique”, Journal of Ambient Intelligence and Humanized Computing, December 2018. In press: https://doi.org/10.1007/s12652-018-1161-0
[29] Mohamed Elhoseny, K. Shankar, S. K. Lakshmanaprabu, Andino Maseleno, N. Arunkumar, “Hybrid optimization with cryptography encryption for medical image security in Internet of Things”, Neural Computing and Applications - Springer, October 2018. https://doi.org/10.1007/s00521-018-3801-x
[30] K. Shankar and P. Eswaran. “RGB Based Secure Share Creation in Visual Cryptography Using Optimal Elliptic Curve Cryptography Technique”, Journal of Circuits, Systems, and Computers, Volume. 25, No. 11, page(s): 1650138-1 to 23, November 2016.