https://doi.org/10.48047/AFJBS.6.14.2024.7961-7976

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

A SECURED MODEL FOR MEDICAL DATA USING HYBRID FUZZY NEURAL NETWORK WITH MODIFIED DEER HUNT OPTIMIZATION AND PALLIER HOMOMOMORPHIC ENCRYPTION IN CLOUD COMPUTING

[1]D Kalpana, [2]Dr. K Ram Mohan Rao

Research Scholar, Osmania University, Hydearbad kalpanadev@gmail.com

[2]Professor & HOD(IT),Vasavi College of Engineering, Hyderabad hodit@staff.vce.ac.in

Volume 6, Issue 14, Aug 2024

Received: 15 June 2024

Accepted: 25 July 2024

Published: 15 Aug 2024

doi: 10.48047/AFJBS.6.14.2024.7961-7976

Abstract: With the deep integration of "AI + medicine", AI-assisted technology has been of great help to human beings in the medical field, especially in the area of predicting and diagnosing diseases based on big data, because it is faster and more accurate. However, concerns about data security seriously hinder data sharing among medical institutions. This research work introduced a novel Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) based Pallier Homomorphic Encryption (PHE) scheme for enhancing the data security of the cloud from malware and attacks. Initially, collected datasets are stored in the cloud using the cloud sim tool, and collected datasets are transferred into the developed FNNHMDH-PHE framework. At first, generate the key for each dataset and separate the private key for all datasets. Moreover, convert the plain text into ciphertext using the FNN and deer fitness function in PHE. Finally, cloud-stored data are encrypted successfully and the attained performance outcomes of the developed framework are associated with other existing techniques in terms of decryption time, encryption time, efficiency, and throughput.

Keywords: Medical data, security, Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) based Pallier Homomorphic Encryption (PHE).

1. Introduction

With the rapid development of information technology, all kinds of data break the restrictions of time and space and accumulate in different fields to form data treasures [1-3]. It breeds huge commercial value and unlimited potential. Therefore, the intelligent use of data and the maximization of data value have become the focus of competition in various industries. But, in the face of the complex network environment, it must ensure data security while exploiting the value of data. Therefore, the research on data sharing based on privacy protection has become a major challenge [4-5]. At present, there are many data sharing schemes, the most widely used of which are the centralized processing mode and distributed processing mode. In the centralized processing mode, all participants need to share their data, that is to say, this kind of centralized mode requires the participant to upload their data to the server, and all of the data are applied for centralized learning or training on the server. If the server is malicious or the server is vulnerable to external attacks, then there is a risk of data privacy leakage [6]. So, this kind of sharing mode undoubtedly reduces the possibility of data sharing among different participants. To solve the above problems of the centralized processing mode and avoid privacy leakage, a new distributed processing mode has emerged.

To deliver results that are sufficiently reliable to be considered in clinical routines, machine learning-based solutions have to heavily rely on available medical data records [7-8]. However, as patient health information has some of the highest privacy requirements among all data types associated with an individual, its usability is greatly hindered. Moreover, given the fact that machine learning-based solutions require access to such sensitive information, concerns have recently been raised regarding data privacy and security. Usually, a proper anonymization must be done in order to export sensitive data without violating confidentiality. Through data masking, some of the information properties are thus changed, resulting in a trade-off between confidentiality and utility [9]. In certain cases, e.g., genomic data, the use of anonymized data limits the neural network's ability to gain valuable information and insight from the data. Herein, a method based on homomorphic encryption (HE) is employed as a way to address the limitations imposed by conventional methods, and to maintain confidentiality of biometric data.

HE is a specific form of encryption which allows data to be encrypted while it is being manipulated. By preserving the mathematical structures that underline the data, HE represents a promising solution for guaranteeing privacy while still maintaining full utility [10-11]. The chosen HE scheme (MORE) allows for a limited set of operations to be conducted directly on encrypted data without exposing the underlying information or the encryption key. In the context of optimized based learning, this property is especially useful as it ensures that both data and predictions are kept private while data are processed. Taking into account the practical difficulties arising from the use of networks on encrypted data and the inefficiency of current approaches, propose a method that improves the effectiveness of encryption models in real-world applications by facilitating calculations over rational numbers, faster operations, and performance close to that achieved with an unencrypted model. This research work introduced a novel Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) based Pallier Homomorphic Encryption (PHE) scheme for enhancing the data security of the cloud from malware and attacks.

The rest of the research work is structured as follows, section 2 reviews the some of the recent techniques for the encryption over medical data for the prediction of patient health data. section 3 presents the process of the proposed methodology. section 4 provides the results and discussion. section 5 deals with the conclusion and future work.

2. Literature Review

In this section, reviews the some of the recent techniques for the encryption over medical data for the prediction of patient health data.

Li et al [12] proposed two Secure and Efficient Dynamic Searchable Symmetric Encryption (SEDSSE) schemes over medical cloud data. First, we utilize the secure k-Nearest Neighbor (kNN) and Attribute-Based Encryption (ABE) techniques to construct a dynamic searchable symmetric encryption scheme, which can achieve forward privacy and backward privacy simultaneously. These tow security properties are vital and very challenging in the area of dynamic searchable symmetric encryption. Then, we propose an enhanced scheme to solve the key sharing problem which widely exists in the kNN based searchable encryption scheme. Compared with existing proposals, our schemes are better in terms of storage, search and updating complexity. Extensive experiments demonstrate the efficiency of our schemes on storage overhead, index building, trapdoor generating and query.

Denis et al [13] developed 2D Discrete Wavelet Transform 1 Level (2D-DWT-1 L) or 2D Discrete Wavelet Transform 2 Level (2D-DWT-2 L) steganography with the proposed hybrid encryption scheme. The hybrid encryption scheme is built by strategically applying Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) algorithms to secure diagnosis data to be embedded with the RGB channels of medical cover image. One of the key novelties is the use of an Adaptive Genetic Algorithm for Optimal Pixel Adjustment Process (AGA-OPAP) that enriches data hiding ability as well as imperceptibility features. To evaluate the efficiency of the proposed model, numerical tests are performed. The results show that the proposed algorithm is capable of safely transmitting medical data. Comparison of results is carried out concerning the datasets with the state-of-the-art algorithm.

Pushpa et al [14] presented a new hybridization of data encryption model to shelter the diagnosis data in medical images. The proposed model is presented by the integration of 2D discrete wavelet transform technique with a proposed hybrid encryption scheme. The presented hybrid encryption scheme is derived by the integration of Blowfish and Two fish encryption algorithms. The presented model begins with the encryption of secrecy data and then concealed the outcome by the use of outcome in a cover image and 2D-DWT-1L or 2D-DWT-2L. The color images are utilized as cover images for concealing various text sizes.

Abdelfattah et al [15] developed an efficient search scheme over encrypted data for a multi-data-owner setting. To secure our scheme, the cloud server obtains noisy similarity scores and doctors de-noise them to download the most relevant documents. Our scheme enables doctors to prescribe search conditions to customize the search without revealing the conditions to the server. Our formal proof and analysis indicate that our scheme can preserve privacy and is secure against known-plaintext/background and linkability attacks, and the results of extensive experiments demonstrate the efficiency of our scheme compared to the existing works.

Vizitiu et al [16] proposed a fully homomorphic encryption (FHE) for the processing of sensitive health information without disclosing the underlying data. The considered encryption scheme, MORE (Matrix Operation for Randomization or Encryption), enables the computations within a neural network model to be directly performed on floating point data with a relatively small computational overhead. We consider the well-known MNIST digit recognition problem to evaluate the feasibility of the proposed method and show that performance does not decrease when deep learning is applied on MORE homomorphic data. To further evaluate the suitability of the method for healthcare applications, we first train a model on encrypted data to estimate the outputs of a whole-body circulation (WBC) hemodynamic model and then provide a solution for classifying encrypted X-ray coronary angiography medical images. The findings highlight the potential of the proposed privacy-preserving deep learning methods to outperform existing approaches by providing, within a reasonable amount of time, results equivalent to those achieved by unencrypted models.

Alzubi et al [17] presented a new privacy-preserving encryption with DL based medical data transmission and classification (PPEDL-MDTC) model. The presented model derives multiple key-based homomorphic encryption (MHE) technique with sailfish optimization (SFO), called MHE-SFO algorithm-based encryption process. In addition, the cross-entropy based artificial butterfly optimization-based feature selection technique and optimal deep neural network (ODNN) based classification is carried out. In ODNN model, the hyperparameter optimization of the DNN model is carried out utilizing the use of chemical reaction optimization (CRO) algorithm. The proposed method has been simulated utilizing Python 3.6.5 tool, which is tested using activity recognition and sleep stage dataset. A detailed comparative outcomes analysis makes sure the higher efficiency of the PPEDL-MDTC on the state of art techniques with the detection accuracy of 0.9813 and 0.9650 on the applied activity recognition and University College Dublin Sleep Stage dataset.

Pandey et al [18] presented a bit mask oriented genetic algorithm based secure medical data transmission mechanism. A bit mask oriented genetic algorithm (BMOGA) is utilized to reduce the replication of medical tests data which are transferred across organizations. The encrypted data is embedded into the medical images through 1-level and 2-level Discrete Wavelet Transform (DWT). The reverse process of the BMOGA is implemented for the extraction of secret message from the encrypted one. Numerical experiments are conducted to determine the performance of the proposed algorithm. Results reveals that the proposed algorithm is capable of secure data transmission. Performance comparison is done with the state-of-the-art algorithm with respect to the datasets. Comparative results indicated the superiority of the proposed algorithm in terms of various statistical measures such as peak signal to noise ratio (PSNR), correlation, structural content (SC), structure similarity (SSIM) and mean square error (MSE) to report the results.

Hesamifard et al [19] developed a new techniques to provide solutions for running deep neural networks over encrypted data. In this paper, we develop new techniques to adopt deep neural networks within the practical limitation of current homomorphic encryption schemes. More specifically, we focus on classification of the well-known convolutional neural networks (CNN). First, we design methods for approximation of the activation functions commonly used in CNNs (i.e. ReLU, Sigmoid, and Tanh) with low degree polynomials which is essential for efficient homomorphic encryption schemes. Then, we train convolutional neural networks with the approximation polynomials instead of original activation functions and analyze the performance of the models. Finally, we implement convolutional neural networks over encrypted data and measure performance of the models. Our experimental results validate the soundness of our approach with several convolutional neural networks with varying number of layers and structures. When applied to the MNIST optical character recognition tasks, our approach achieves 99.52\% accuracy which significantly outperforms the state-of-the-art solutions and is very close to the accuracy of the best non-private version, 99.77\%.

Bost et al [20] constructed three major classification protocols that satisfy this privacy constraint: hyperplane decision, Naïve Bayes, and decision trees. These protocols may also be combined with AdaBoost. They rely on a library of building blocks for constructing classifiers securely, and we demonstrate the versatility of this library by constructing a face detection classifier. Our protocols are efficient, taking milliseconds to a few seconds to perform a classification when running on real medical datasets.

Bokhari et al [21] used K nearest neighbors algorithm to decide whether the data is normal sensitive or high sensitive, and then according to the level of sensitivity we proposed a framework to do data encryption. In order to ensure user authentication, we used one time password to authenticate the user, and for the data which belong to normal sensitivity level we have applied AES (advanced encryption standard)-192 algorithm. Finally, for the data which belong to high sensitivity level, AES-256 algorithm has been applied, and RSA (Rivest–Shamir–Adleman) algorithm is used to encrypt the key of AES 256, then we use hash-based massage authentication code to be attached in the end of message to ensure integrity and authenticity of message.

Jäschke et al [22] introduced an unsupervised learning, which is an important area in Machine Learning and has many real-world applications, by addressing the clustering problem. To this end, we show how to implement the K-Means-Algorithm. This algorithm poses several challenges in the FHE context, including a division, which we tackle by using a natural encoding that allows division and may be of independent interest. While this theoretically solves the problem, performance in practice is not optimal, so we then propose some changes to the clustering algorithm to make it executable under more conventional encodings. We show that our new algorithm achieves a clustering accuracy comparable to the original K-Means-Algorithm, but has less than 5% of its runtime.

3. Proposed Methodology

This research work introduced a novel Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) based Pallier Homomorphic Encryption (PHE) scheme for enhancing the data security of the cloud from malware and attacks. Initially, collected datasets are stored in the cloud using the cloud sim tool, and collected datasets are transferred into the developed FNNHMDH-PHE framework. At first, generate the key for each dataset and separate the private key for all datasets. Moreover, convert the plain text into ciphertext using the FNN and deer fitness function in PHE. Finally, cloud-stored data are encrypted successfully and the attained performance outcomes of the developed framework are associated with other existing techniques in terms of decryption time, encryption time, efficiency, and throughput. The figure 1. illustrate the process of the proposed methodology.

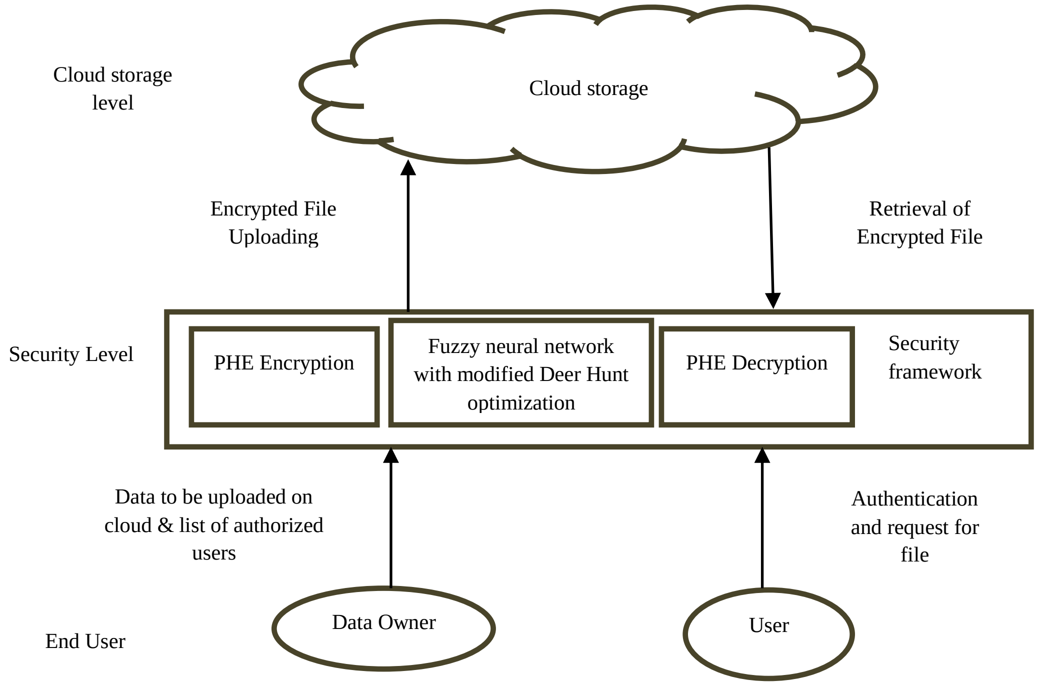
Cloud storage level
Cloud storage
Encrypted File Uploading
Retrieval of Encrypted File
Security Level
PHE Encryption
Fuzzy neural network with modified Deer Hunt optimization
PHE Decryption
Security framework
Data to be uploaded on cloud & list of authorized users
Authentication and request for file
End User
Data Owner
User

Figure 1. the process of the proposed methodology

3.1. Security Model

The entire system security structure model has been separated into three layers, which are depicted in Figure 1.

(a) Level of the End User

This level includes both the owner and those who make use of the data. The data owner uploads encrypted data together with a list of all authorised users to the security environment.

(b) Security Level

The security level of the proposed system includes SM-PHC security framework, which operates using the following workflow:

1. File uploading: The user initiates the process by selecting a file from their local machine and uploading it through the interface of the security framework. This file is then stored in the local database for safekeeping.

2.File Encryption: The security framework employs Pallier Homomorphic Encryption (PHE) to convert the plain text of the uploaded file to cipher text. This conversion process happens during the encryption phase, and the resultant cypher text is secure and unreadable to any unauthorized parties.

3.Secure File Uploading: The encrypted file is then uploaded to the public cloud, where it is stored securely to ensure that no unauthorized persons can access it.

4.File downloading: When a user needs to retrieve a file, they submit the file name to the cloud. The cloud system searches for the file and returns it to the user in an encrypted form.

5.File decryption: The encrypted file is decrypted using the PHE decryption algorithm. This algorithm requires the private key that was used during the encryption phase to convert the cypher text back to plain text.

6.Optimal Key generation: The security framework generates a pair of keys optimized by the spider monkey optimization method (SMO). These keys are used to ensure secure encryption and decryption of files, and the optimization process helps to improve the efficiency of the encryption algorithm.

(c) Cloud storage level

The highest level element of the security concept outlined in this study is a storage cloud. The security framework uploads encrypted user data via the cloud interface, where it is then stored. Users can use the security framework to download an encrypted file when they need it. The file is sent to the user after being decrypted by the security framework. This system concept can also be applied to a multi-cloud setting. However, the spider monkey-Paillier homomorphic encryption algorithm is the primary emphasis of the study. This technique encrypts and decrypts data before it is uploaded to the cloud in order to offer the best possible solution for data security in cloud computing.

3.2. Pallier Homomorphic Encryption (PHE)

HE is a method that can process ciphertext information. It is an encryption technology that allows computing operations on ciphertext and generates encryption results [23]. The calculation result obtained in the ciphertext is decrypted and matched with that in plaintext as if the same calculation operation has been performed on plaintext. The processing flow in encrypted and unencrypted states is shown in Figure 2.

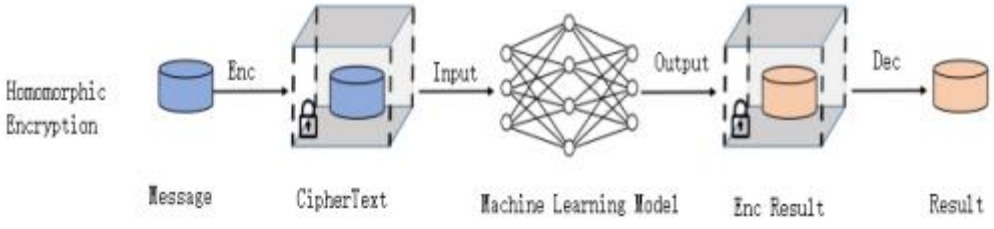

Figure 2. Process of PHE

As a method that can process ciphertext without decrypting ciphertext, HE is the most commonly used privacy protection mechanism nowadays. HE mechanism can compute the ciphertext without decrypting the ciphertext so that the computation party does not need to know the contents of the plaintext, but only needs to obtain the ciphertext, which is a good way to protect sensitive data and information while performing computation operations. HE can efficiently process cryptographic information and achieve specific algebraic operations on the encrypted content. The HE cryptosystem is composed of quaternions, as shown in Eq (1).

$$H = \{Ho_{key}, Enc, Dec, Eval\} \quad (1)$$

where Ho_{key} represents the key generation function, Enc represents the encryption function, Dec represents the decryption function and $Eval$ represents the evaluation function. $Enc_{pub}(.)$. A secure cryptosystem such as Eq (2) can be called a homomorphic operation. Using represents the encryption function that uses the public key pub as the encryption key, M represents the plaintext space and C represents the ciphertext space.

$$\forall m_1, m_2 \in M, Enc_{pub}(m_1 \odot_M M_2) \leftarrow Enc_{pub}(m_1) \odot_c Enc_{pub}(m_2) \qquad (2)$$

$\odot_M$ and $\odot_c$ represent the operator on the plaintext space M and the ciphertext space C, respectively. Eq (2) shows that, for any two elements m_1 and m_2 in the plaintext space M, after performing the $\odot_M$ *operating* on them, the obtained result is encrypted; the result is the same as if m_1 and m_2 were encrypted first and then the operators were executed. The symbol "□" indicates that the left-hand term is equal to or can be computed directly from the right-hand term without any intermediate decryption operation. To simplify the expression, we can use [[v]] to represent the result of HE for the plaintext v. The two basic operations of HE are defined below, namely, addition HE and multiplication HE.

Definition 1: Additive homomorphic operation. For any two elements u and v in the plaintext space, with the encryption results are respectively [[u]] and [[v]], with Dec_{pri} indicating that the private key is used for decryption if Eq (3) is satisfied:

$$Dec_{pri}([[u]] + [[v]]) = Dec_{pri}([[u + v]]) = u + v \qquad (3)$$

Definition 2: Multiplicative homomorphic operation. For any two elements u and v in the plaintext space, with the encryption results are respectively [[u]] and [[v]], with Dec_{pri} indicating that the private key is used for decryption if Eq (4) is satisfied:

$$Dec_{pri}([[u]] \times [[v]]) = Dec_{pri}([[u \times v]]) = u \times v \qquad (4)$$

3.2. Paillier algorithm

In the Paillier algorithm, the generation steps of the public-private key pair and the principle of encryption and decryption are as follows.

Key generation: First, randomly select two large prime numbers a and b (ensure that a and b are of equal length, Next, calculate n=ab and $\lambda = lcm(a-1, b-1)$, where lcm is a function to find the least common multiple. Define $L(x) = \frac{x-1}{n}$, and then randomly select a positive integer g less than $n^2$to satisfy Eq (5):

$$gcd\ gcd\ (L(g^{\lambda} mod\, n^2), n) = 1, u = (L(g^{\lambda} mod\, n^2))^{-1}, mod\, n) \qquad (5)$$

gcd is a function to find the maximum common divisor. By the above operation, we can get the public key (n,g) and private key (λ, u).

Encryption process: For any plaintext message m , choose any random number r satisfying 0<r< n ; the ciphertext c is calculated by Eq (6):

$$c = g^m r^n modn^2 \qquad (6)$$

Decryption process: For the ciphertext c, the plaintext message m is obtained from Eq (7):

$$m = L(c^{\lambda} modn^2) * u\ mod\ n \qquad (7)$$

The Paillier algorithm is an implementation of an asymmetric encryption algorithm, which can operate on encrypted data under encryption and then decrypt the encrypted result. The obtained result is the same as the result of directly operating on the plaintext. However, the Paillier algorithm does not satisfy the multiplicative homomorphic operation. Although the Paillier algorithm is not fully HE, its computational efficiency is high, so it is widely used in the industry. In this paper, the Paillier algorithm is used as the simulation algorithm of HE.

- **Calculation of the encryption loss function**

In the machine learning model that uses the Paillier algorithm to protect the training parameters, the public and private keys are generally generated randomly on the server side. The public key is mainly for encrypting data, and the private key is for decrypting data. In machine learning models, a loss function is usually first defined, and then an optimization algorithm such as a stochastic gradient that minimizes the value L(⎝;x). The parameter ⎝* descent is used to find the minimum value of L(⎝;x) is optimal. Taking logistic regression as an example, let the current set of n sample data points be $T = (x_1, y_1), (x_2, y_2), \dots \dots \dots (x_n, y_n)$, and use the logarithmic loss function as its target loss function as shown in Eq (8):

$$L = \frac{1}{n} \sum_{i=1}^{n} \square \, log\ (1 + e^{-y_i \theta^r x_i}) \qquad (8)$$

The model parameters are updated by taking the partial derivative ⎝ in the above Eq (8) and bringing the obtained gradient values into the gradient descent equation as shown in Eq (9)

$$\theta = \theta - lr * \frac{\partial L}{\partial \theta} \qquad (9)$$

The above computation process is repeated until the value of the loss function L(;) x is no longer decreasing or the maximum number of iterations is reached; then, the iteration is stopped. The above computation process, including the parameters and data information, is computed in the explicit state, and there is a risk of data leakage in the machine learning scenario. Machine learning based on HE requires that the parameters are solved in the encrypted state, i.e., ; the transmitted parameter ⎝ is usually an encrypted value [[⎝]] the loss function is shown in Eq (10).

$$L = \frac{1}{n} \sum_{i=1}^{n} \square \, log\ (1 + e^{-y_i [[\theta]]^r x_i}) \qquad (10)$$

The calculation of the loss function involves exponential and logarithmic operations on the encrypted data, but the Paillier algorithm only supports addition homomorphism and scalar multiplication homomorphism; it does not support multiplication homomorphism and complex exponential and logarithmic operations. Therefore, it is not possible to solve the above Eq (10) in the encrypted state. Here, we use the Taylor loss function to approximate the original logarithmic loss function instead, i.e., by Taylor expansion of the original logarithmic loss function, the logarithmic loss function is approximated by polynomials, and after Taylor expansion, the loss function is transformed into only scalar multiplication and addition operations so that Paillier can be applied directly to the cryptographic solution.

When using the Paillier algorithm for encryption and decryption operations, a large number of large prime power operations are involved, so intermediate results may be out of bounds and usually result in overflow errors. Therefore, we design the Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) algorithm to re-encrypt the data using the server-side key when the number of local training iterations reaches a certain number of rounds.

3.3. Hybrid Fuzzy Neural Network (FNN) and Modified Deer Hunt Optimization (HMDH))

3.3.1. Fuzzy Neural Network (FNN)

Neural networks and fuzzy logic are rapidly growing technologies that could be applied to the formulation and processing of health data for the prediction of breast cancer. Evolutionary algorithms are effectively used in combination with artificial neural networks (ANNs) for predicting and optimizing the formulation conditions. Fuzzy logic provides a natural bridge between first-order logic and neural networks, and indeed fuzzy-neural systems seem to have flourished perhaps more than other approaches to symbolic connectionism. A fuzzy neural network can be viewed as a three-layer feed forward network, with a fuzzy input layer (fuzzification), a hidden layer containing the fuzzy rules, and a final fuzzy output layer (defuzzification) [24]. Fuzzy sets are contained within the (fuzzy) connections between layers, though sometimes a five-layer network with sets contained in the second and fourth layers can be found. The input layer represents the input membership functions for the fuzzy rules, with sufficient input causing a rule in the hidden layer to fire. The weights between the layers represent the fuzzy sets, with membership in each set determined by the relative weights – these can be altered using particular training algorithms as per a normal neural system. Transfer functions are usually continuous and pass real values through the network to the output layer to be interpreted as degrees of membership in fuzzy sets based on the firing of fuzzy rules in the hidden layer.

One of the most popular neural networks is the layered feed forward neural network with a Back Propagation (BP) least mean- square learning algorithm. Its topology is shown in Fig. 3. The network edges connect the processing units called neurons. With each neuron input there is associated a weight, representing its relative importance in the set of the neuron's inputs. The inputs' values to each neuron are accumulated through the net function to yield the net value: the net value is a weighted linear combination of the neuron's inputs' values. For the purpose of multicriteria analysis a hierarchy of criteria is used to determine an overall pattern evaluation. The hierarchy can be encoded into a hierarchical neural network where each neuron corresponds to a criterion. The input neurons of the network correspond to single criterion. The hidden and output neurons correspond to complex criteria. As evaluation function it can be used as the net function of the neurons. However, the criteria can be combined linearly when it is assumed that they are independent.

But in practice the criteria are correlated to some degree. The linear evaluation function is unable to capture relationship between the criteria. In order to overcome this drawback of Standard Back Propagation (SBP) algorithm

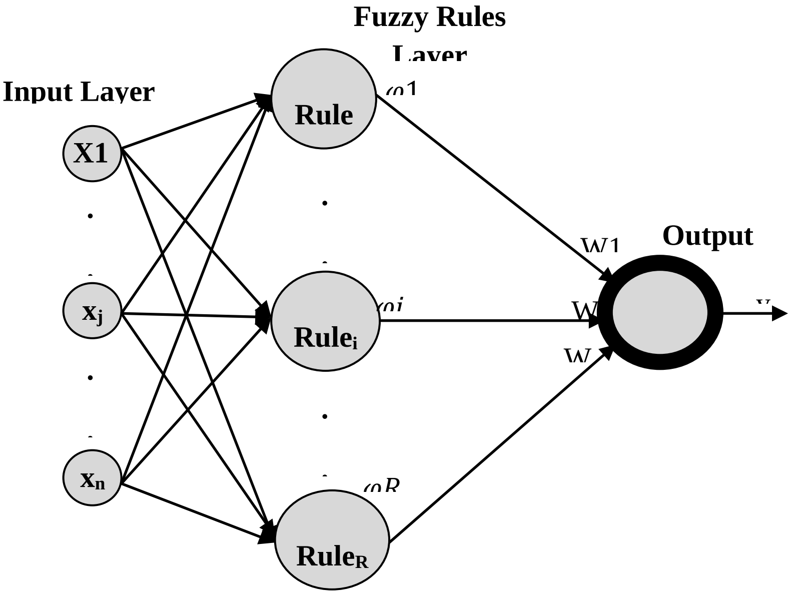

Fig.3. Example neural network with single-output neuron and a hidden layer.

In this work propose a fuzzy extension called Fuzzy Backpropagation (FBP) algorithm. It determines the net value through the LR type fuzzy number and thus does not assume independence between the criteria. Another advantage of FBP algorithm is that it reaches always forward to the target value without oscillations and there is no possibility to fall into local minimum. Necessary and sufficient conditions for convergence of FBP algorithm for single-output networks in case of single- and multiple-training patterns are proved.

- **Fuzzy Back propagation algorithm (FBP algorithm)**

Recently, many neuro-fuzzy models is introduced for yielding the net value net i the inputs values of the ith neuron are aggregated. The mapping is mathematically described by the fuzzy integral of Sugeno that relies on psychological background

Step1: Randomly generate the initial weight sets w for the input hidden layer where each $w_{ji} = (w_{mji}, w_{\alpha ji}, w_{\beta ji})$ is an LR type fuzzy number. Also generate the weight set w' for the hidden output layer

Where $w'_{kj} = (w'_{mkj}, w'_{\alpha kj}, w'_{\beta kj})$

$$w_{ji} = (w_{mji}, w_{\alpha ji}, w_{\beta ji})$$

$$w'_{kj} = (w'_{mkj}, w'_{\alpha kj}, w'_{\beta kj})$$

Step2: Let $(I_p, D_p)\ p = 1, 20 \ldots N$ input-output pattern set, that fuzzy back propagation needs to be trained with. Here $I_p = (I_{p0}, I_{p1}, I_{p1})$ where each I_{pi} is an LR-type fuzzy number.

Step3: Assign values for α and η; Alpha=0.1 Neta =0.9

Step4: Get next pattern set (I_p, D_p) Assign $(O_{pi} =\ I_{pi}$, i=1,2,3..1

Step5: Compute the input to hidden neurons

$$O'_{pj} = f\left(NET_{pj}\right), j = 1,2 \ldots., m; O'_{p0} = 1$$

Where $NET_{pj} = CE\ (\sum_{}^{} W_{ji}O_{pi})$

Step6: compute the hidden to output neurons

$$O''_{pk} = f\ (NET'_{pk})\ , k = 1,2, . . n;$$

Where $NET'_{pk} = CE\ (\sum_{}^{} W_{ji}O'_{pj})$

Step7: compute change of weights Δ w'(t) for the hidden output layer as follows

Compute

$$\Delta E_p(t) = (\partial E_p/\partial w'_{mkj}\ , \partial E_p/\partial w\alpha kj, \partial Ep/\partial w'_{\beta kj})$$

Compute

$$\Delta w'(t)\ = -\eta \Delta E_p(t) + \alpha \Delta w'(t-1)$$

The update weight i of hidden to output neuron is

$$W'(t)\ = W'(t-1)\ + \Delta W'(t)$$

Step 8: Compute change of the weights Δ w'(t) for the input hidden layer as follows

Let

$$\delta_{pmk} = -\left(D_{pk} - O''_{pk}\right)O''_{pk}(1 - O''_{pk}).1$$

$$\delta_{pmk} = -\left(D_{pk} - O''_{pk}\right)O''_{pk}\left(1 - O''_{pk}\right).\left(-\frac{1}{3}\right)$$

$$\delta_{pmk} = -\left(D_{pk} - O''_{pk}\right)O''_{pk}\left(1 - O''_{pk}\right).\left(\frac{1}{3}\right)$$

Compute

$$\Delta E_p(t) = (\partial E_p/\partial w'_{mji}\ , \partial E_p/\partial w\alpha_{ji}, \partial Ep/\partial w'_{\beta ji})$$

Compute

$$\Delta w'(t)\ = -\eta \Delta E_p(t) + \alpha \Delta w'(t-1)$$

Step 9: updateweight for the input-hidden-output layer as

$$W(t)\ = W(t-1)\ + \Delta W(t)$$

$$W'(t)\ = W'(t-1)\ + \Delta W'(t)$$

$Step10: p = p + 1;$

if (p<=N) go to step 5

Step 11: count_of_itrns=count_of_itrns+1;

if count_of_itrns<itrns

{

```
        Reset pointer of first pattern in the training set;

        P=1;

        Go to step 5;

        }
```

The value of a weight, which is multiplied with the input, decides whether a certain input needs to be taken seriously or not. No change in value means the network isn't learning anything. In short, there is no point in training. So this research work introduced a modified deer hunt optimization for solving this problem. The below section describes the modified deer hunt optimization based improvement for enhancing the performance of the prediction rate.

3.3.2. Modified Deer Hunt Optimization algorithm

Deer has special features that make the hunting process difficult for the predators. The visual sense is a prominent feature in a deer. This sense is five times stronger than the human visual sense. Another feature is its olfactory sense. The olfactory sense is sixty times stronger than humans [25]. Once a deer senses danger, it alerts other deer by sniffing loudly and treading heavily. Another specific feature of a deer is its ability in the detection of ultra-high-frequency sounds. In the following, the method of deer hunting is illustrated in detail.

- **initialization**

Like any meta-heuristic algorithm, deer hunting optimization starts with a set of the random population called hunters. This initializing can be determined as follows:

$$X = [y_1, y_2, \ldots\ldots y_m], \qquad 1 < i \leq m \qquad (11)$$

where, m describes the number of hunters population (solutions) and X is the total population.

- **Initializing the Parameters**

In the second step, the deer's position angle (position angle) and wind angle as the main parameters of the algorithm should be initialized. Because the search space is assumed as a circle, the wind angle is formulated in the circumference of a circle.

$$\theta_i = 2\pi\lambda \qquad (12)$$

where, λ describes a random value between 0 and 1 and i is at the present iteration. In addition, the position angle of the deer is modeled as follows:

$$\phi_i = \pi + \theta \qquad (13)$$

where, θ represents the wind angle.

- **Position propagation**

Finding the best solution to the algorithm at the first iteration is almost impossible. However, after evaluating the value of the cost function based on the randomly generated values, the best value is assumed as the candidate optimum solution value. In this part, two parameters are considered; the first one is the leader position (X^L) that describes the first best position of the hunter and the second parameter is the successor position (X^S) that is the succeeding hunter position.

(a) Propagation based on the leader's position:

By applying the first iteration for achieving the best positions, all the population attempts to obtain the best position by updating the position of them. Therefore, encircling behavior can be formulated as follows:

$$X_{i+1} = X^L - Y \times S_w \times |L \times X^L - X_i| \qquad (14)$$

where, X_i and X_{i+1} describe the current and the next positions, S_w describes a random value based on the wind speed in the range [0, 2], and L and Y illustrate the coefficient vectors which can be formulated as follows:

$$Y = 0.25 \times log\ log\left(I + \frac{1}{I_{max}}\right)\beta \qquad (15)$$

$$L = 2 \times \tau \qquad (16)$$

where, I_{max} describes the maximum iteration, β and is a random parameter in the range -1 and 1, and τ is a random value in the range [0, 1]. Fig. 4. shows the position updating, where (X, Z) describes the initial position of the hunter which can be updated based on the prey position. The position updating will be continuing once the best position (X^*, Z^*) is achieved based on L and Y. Hunters move to the direction where the leader is positioned. If the leader has an unsuccessful movement, the hunter will remain in the prior position. Position update follows Eq. (16) only when $S_w < 1$, i.e. the hunters can randomly move in any direction without respect to the position angle. So, based on Eqs. (15) and (16), the hunters can update their position in any random location within the space.

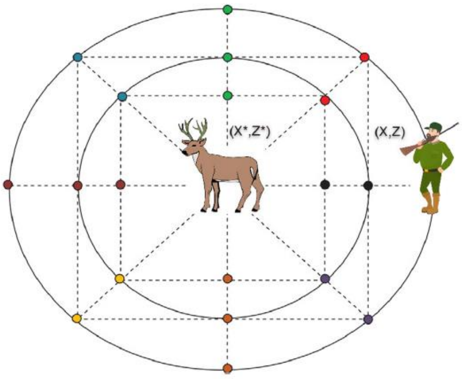

Fig. 4. Updating the best position X∗.

(b) Propagation based on the position angle

With considering the position angle in the update rule, we can develop the solution space. The evaluating of the angle is important for assigning the position of the hunter such that prey is uninformed of the attack which makes the hunting process successful. The angle of visualization for the prey (deer) is formulated as follows:

$$a_i = \frac{1}{8} \times \pi \times \lambda \qquad (17)$$

By considering the difference between the prey visual angle and the wind angle, we can evaluate a parameter that helps for updating the position angle**.**

$$d_i = \theta_i - a_i \qquad (18)$$

where, θ_i represents the wind angle. Afterward, for updating the position angle,

$$\phi_{i+1} = \phi_i + d_i \qquad (19)$$

And finally, by using the achieved position angle, the new position is achieved by the following equation,

$$X_{i+1} = X^L - S_w \times \left| cos\ cos\ (\phi_{i+1}) \times X^L - X_i \right| \qquad (20)$$

Since the hunter is out of the visual angle of the deer, the hunter is not in the view of it.

(c) Propagation based on the position of the successor

For applying the exploration term to the encircling behavior, the vector L can be adjusted. Based on assuming an initial random search, the value of the vector L is not considered more than 1. Hence, the successor position is utilized for updating instead of the first achieved best solution. This process makes a global search for the algorithm as the following equation:

$$X_{i+1} = X^S - Y \times S_w \times |L \times X^L - X_i| \qquad (21)$$

where, X^S describes the successor position of the hunters from the current population. In each iteration, the algorithm updates the position of the hunters by considering the best solution. The best solution has been selected when $|L| \geq 1$. Once $|L| < 1$, a hunter will be chosen randomly.

This procedure gives a switch called L which can change the algorithm between exploitation and exploration phases. A disadvantage of the original DHOA is it is the problem of premature convergence. In the next part, a new mechanism is proposed for removing this shortcoming.

- **The Modified DHOA**

In this part, Lévy flight (LF) is adopted to develop the DHOA. Lévy flight is a mechanism for resolving the problem of premature convergence drawback. Lévy flight generate a random walk mechanism for suitable controlling of the local search as follows:

$$Le(w) \approx w^{-1-\xi} \qquad (22)$$

$$w = \frac{A}{|B|^{\frac{1}{\xi}}} \qquad (23)$$

$$\sigma^2 = \left\{ \frac{\Gamma(1+\xi)}{\xi\Gamma\left(\frac{1+\xi}{2}\right)} \frac{sinsin\left(\frac{\pi\xi}{2}\right)}{2^{\frac{1+\xi}{2}}} \right\}^{\frac{2}{\xi}} \qquad (24)$$

where, $0 < \tau \leq 2$, $A \sim N(0,\sigma^2)$ and $B \sim N(0,\sigma^2), \Gamma(.)$ describes the Gamma function, w represents the step size, ξ represents Lévy index, $A/B \sim N(0,\sigma^2)$ means that the samples generated from a Gaussian distribution in which mean are zero and variance is σ^2, respectively. In this work, $\xi = 3/2$. Based on the Lévy flight mechanism, the new improved position for the hunters is:

$$X_{i+1}^{l} = X_{i+1} + (X * -AD) \times Le(\delta) \qquad (25)$$

where, X_{i+1}^{l} represents the new position of search agent X_{i+1} and

$$A = a(2 \times r - 1) \qquad (26)$$

$$D = CX'(t) - X(t) \qquad (27)$$

where, a is in the range 0 and 2, r is a random value in the interval [0, 1], and $X'(t)$ describes a random position vector selected from the present population. To guarantee the best solution candidates, fitter agents are kept:

$$\vec{D}_{el} = \{X_{i+1}^{l} \qquad F\left(X_{i+1}^{l}\right) > F(X_{i+1})\ X_{i+1} \qquad otherwise \qquad (28)$$

So the proposed model provides the best security for the patient medical data using the proposed hybrid deep learning with encryption scheme.

4. Results and Discussion

Initially, a medical diabetic dataset is obtained from the internet and used to train the system using 800 samples and the key is generated using paillier homomorphic encryption (PHE), which can be optimized using optimization. As a result, which gives optimal keys for encryption and decryption. Finally, the selected dataset is encrypted using a public key, which is then stored securely in real time database of public cloud for further accessing. To maintain confidentiality and to prevent unauthorized access by third parties, the data is provided in an encrypted form when accessed from the public cloud by the user and data owner. The proposed mechanism established in this study provides protection against malicious activity and unauthorized access through the use of encrypted data. Performance metrics are compared with those of other current methods to evaluate the effectiveness of the suggested procedure. This section presents a comparison between the proposed technique and several existing homomorphic encryption methods, Key Homomorphic Encryption (KHE), Energy Efficient Dynamic Homomorphic Security (EE-DHS) spider-monkey wıth Paillier homomorphic encryption model (SM-PHE), and proposed Hybrid Fuzzy Neural Network with Modified Deer Hunt optimization based Pallier Homomorphic Encryption (HFNNMDH-PHE).

Encryption time: To calculate the encryption time, you can measure the time taken by the encryption algorithm to encrypt the plaintext into ciphertext. Generally, the encryption time can be calculated by subtracting the start time from the end time of the encryption process. Encryption time is a critical factor in evaluating the efficiency of an encryption algorithm, as it reflects the amount of time required to convert plaintext to ciphertext. The shorter the encryption time, the more efficient the algorithm. This Section compares the encryption time of proposed with that of other homomorphic encryption algorithms.

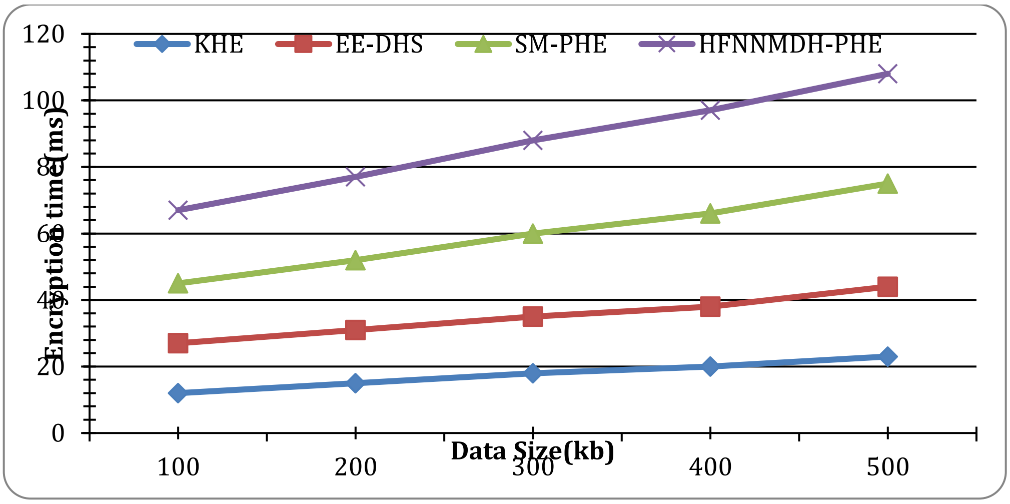

Figure 5. Encryption Time

The results of figure1. demonstrate that proposed model takes less time to encrypt the same plaintext compared to the other algorithms, indicating its superior efficiency in this aspect. The intended HFNNMDH-PHE model achieved 4 ms for 100kb using this metric. As a result, the EE-DHS and SM-PHE methods achieved 8, and 12 ms, respectively, while the KHE approach reached 18 ms for 100kb, as shown in Fig 5.

Decryption Time: The length of time required to decrypt encrypted data or information is referred to as decryption time. Decryption is the process of turning encrypted or encoded material back to its original form so that the intended recipient can read and understand it. The amount of time it takes to decrypt data depends on the encryption algorithm employed, the size of the data being decrypted, and the computational capability of the device used for decryption. In general, more advancedencryption algorithms and higher data volumes will necessitate more decoding time. Using this criteria, the proposed model accomplished 5 ms for 100kb**.**

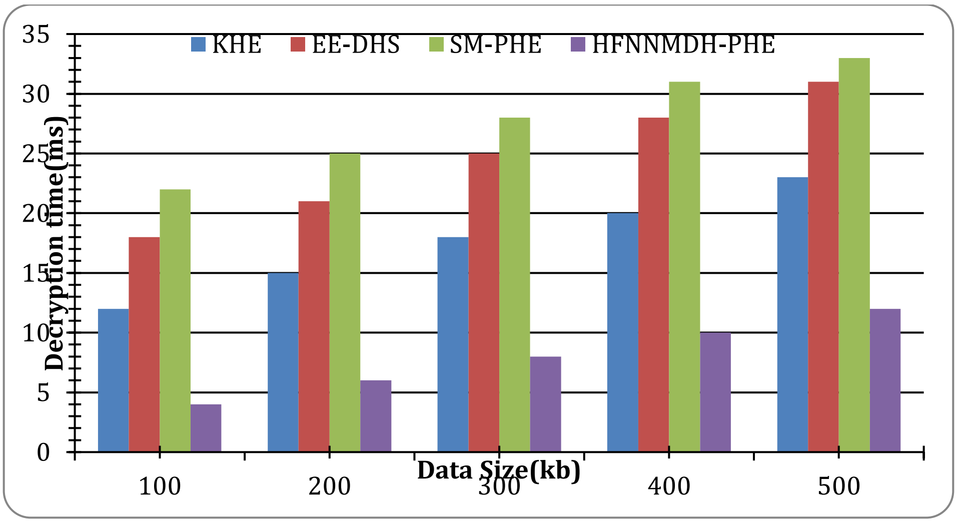

Fıgure 6. decryption time

As a result, for data sizes of 200 kb, 300 kb, 400 kb, and 500 kb, the decryption time is 7, 9, 11, and 14 milliseconds.The proposed HFNNMDH-PHE technique gained less decryption time when compared to other models expressed in fig 6.

Execution Time: The execution time of an algorithm is the amount of time it takes for the algorithm to complete its task or solve a problem. Execution time is typically measured in units of time, such as seconds, milliseconds, or microseconds.

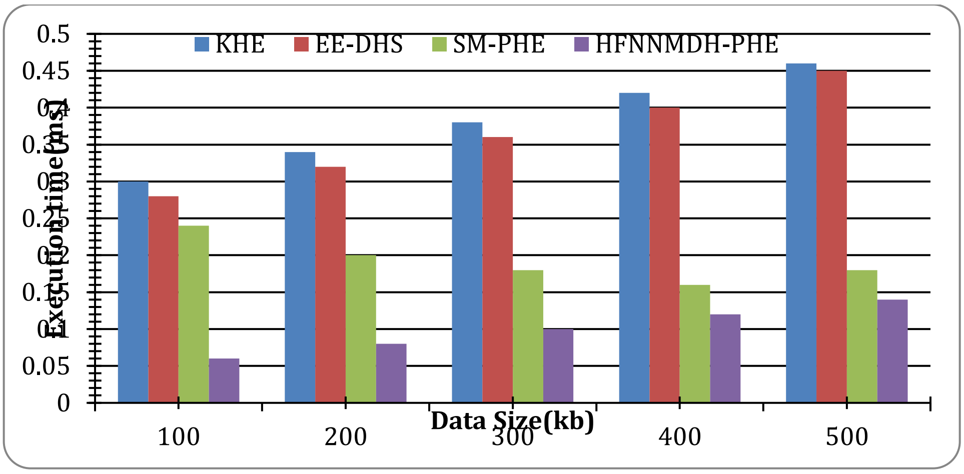

Figure 7. Execution time

The specific unit used may depend on the granularity required to accurately measure the execution time of the operations, which can be expressed in fig 7. The proposed model takes the less time for the execution compared with the other models.The proposed model achieves shorter execution time of 0.06 ms for 100 kB, 0.08 ms for 200 kB, 0.10 ms for 300 kB, 0.12 ms for 400 kB, and 0.14 ms for 500 kB of data size as a result in fig 3.

Efficiency: The efficiency of an algorithm can be evaluated based on several factors, including computational complexity, memory requirements, power consumption, and communication overhead. These factors determine how quickly and efficiently the algorithm can perform its cryptographic operations while using the minimum possible resources and proposed model is more efficient than other homomorphic techniques.

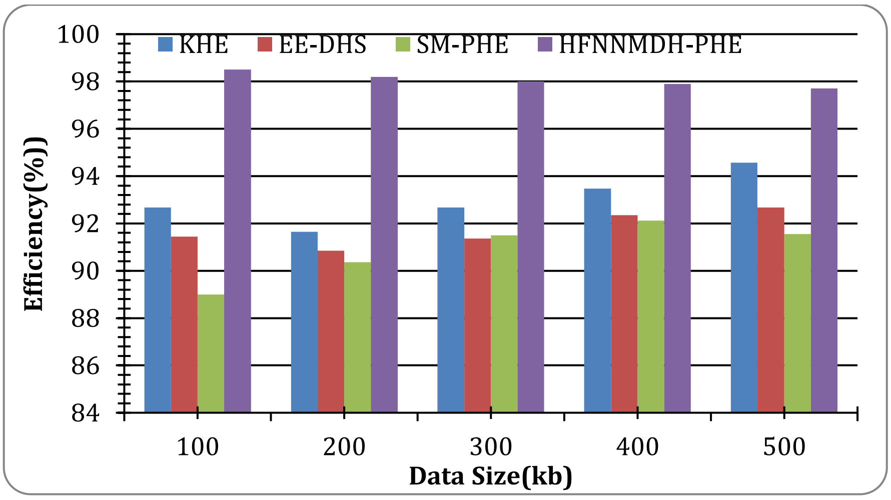

Figure 8. efficiency

In this case, the suggested model's efficiency was higher than that of the other models, as a consequence, proposed model's efficiency is higher than the other models' in the following cases: 98.5% for 100 kb, 98.2% for 200 kb, 98% for 300 kb, 97.9% for 400 kb, and 97.7% for 500 kb in the fig 8.

Power consumption: The power consumption can be estimated based on several factors, including the size of the public key, the length of the plaintext and ciphertext, and the number of operations required to perform the homomorphic computation. The less power consumption leads to secure and energy-efficient system.

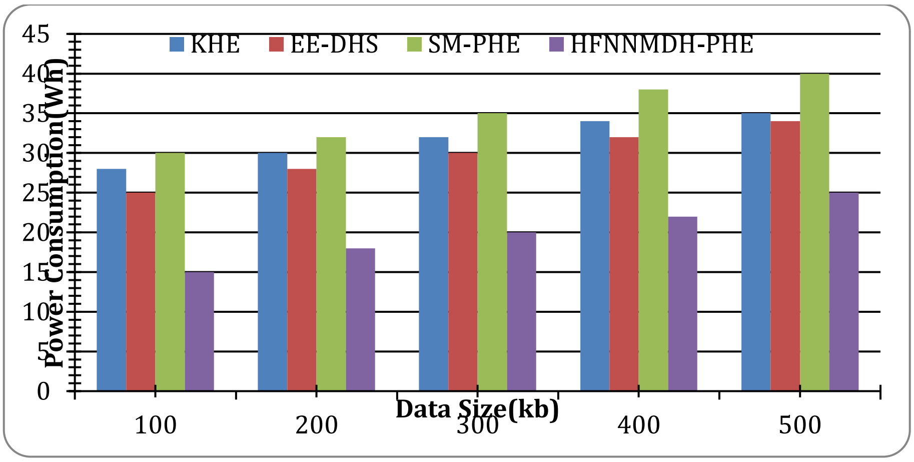

Figure 9. power consumption

The suggested model uses less energy than the current models, such as HFNNMDH-PHE, which uses 15Wh for 100kb of data and 18Wh, 20Wh, 22Wh, and 25Wh for 200kb, 300kb, 400kb, and 500kb, respectively in fig.9.

5. Conclusion

The field of adversarial machine learning has received significant research attention in recent years. In particular, many research works have explored adversarial attacks on machine learning models in the context of computer vision and image recognition, natural language processing and cybersecurity. Several defense methods have also been proposed with research efforts focused on evaluating the effectiveness of these defense methods against the continually evolving adversarial attacks. This research work introduced a novel Hybrid Fuzzy Neural Network (FNN) with Modified Deer Hunt Optimization (HMDH) based Pallier Homomorphic Encryption (PHE) scheme for enhancing the data security of the cloud from malware and attacks. Initially, collected datasets are stored in the cloud using the cloud sim tool, and collected datasets are transferred into the developed FNNHMDH-PHE framework. At first, generate the key for each dataset and separate the private key for all datasets. Moreover, convert the plain text into ciphertext using the FNN and deer fitness function in PHE. Using real-world datasets, evaluated the performance of the proposed model and demonstrated its feasibility in encryption, decryption and power consumption. For example, it took approximately 116 minutes to obtain the training model from the homomorphically encrypted for given dataset. In addition, it gives fairly accurate predictions on the testing dataset. While these implementations remain difficult for non-experts to navigate, current and future works will provide more high-level, user-friendly technologies. These would enable researchers from a variety of backgrounds to utilize FHE for their own research.

REFERENCES

1. Singh, A. K., Anand, A., Lv, Z., Ko, H., & Mohan, A. (2021). A survey on healthcare data: a security perspective. *ACM Transactions on Multimidia Computing Communications and Applications*, *17*(2s), 1-26.
2. Aiswarya, R., Divya, R., Sangeetha, D., & Vaidehi, V. (2013, July). Harnessing healthcare data security in cloud. In *2013 International Conference on Recent Trends in Information Technology (ICRTIT)* (pp. 482-488). IEEE.
3. Chandra, S., Ray, S., & Goswami, R. T. (2017, January). Big data security in healthcare: survey on frameworks and algorithms. In *2017 IEEE 7th International Advance Computing Conference (IACC)* (pp. 89-94). IEEE.
4. Benaloh, J., Chase, M., Horvitz, E., & Lauter, K. (2009, November). Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (pp. 103-114).
5. Boussif, M., Aloui, N., & Cherif, A. (2018). Secured cloud computing for medical data based on watermarking and encryption. *IET Networks*, *7*(5), 294-298.

6. Elhoseny, M., Ramírez-González, G., Abu-Elnasr, O. M., Shawkat, S. A., Arunkumar, N., & Farouk, A. (2018). Secure medical data transmission model for IoT-based healthcare systems. *Ieee Access*, *6*, 20596-20608.
7. Kaufman, L. M. (2009). Data security in the world of cloud computing. *IEEE security & privacy*, *7*(4), 61-64.
8. Tamilarasi, K., & Jawahar, A. (2020). Retracted article: Medical data security for healthcare applications using hybrid lightweight encryption and swarm optimization algorithm. *Wireless Personal Communications*, *114*(3), 1865-1886.
9. Elhoseny, M., Shankar, K., Lakshmanaprabu, S. K., Maseleno, A., & Arunkumar, N. (2020). Hybrid optimization with cryptography encryption for medical image security in Internet of Things. *Neural computing and applications*, *32*, 10979-10993.
10. Christo, M. S., Jesi, V. E., Priyadarsini, U., Anbarasu, V., Venugopal, H., & Karuppiah, M. (2021). Ensuring improved security in medical data using ecc and blockchain technology with edge devices. *Security and Communication Networks*, *2021*(1), 6966206.
11. Daman, R., Tripathi, M. M., & Mishra, S. K. (2016, March). Security issues in cloud computing for healthcare. In *2016 3rd international conference on computing for sustainable global development (INDIACom)* (pp. 1231-1236). IEEE.
12. Li, H., Yang, Y., Dai, Y., Yu, S., & Xiang, Y. (2017). Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data. *IEEE Transactions on Cloud Computing*, *8*(2), 484-494.
13. Denis, R., & Madhubala, P. (2021). Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems. *Multimedia Tools and Applications*, *80*(14), 21165-21202.
14. Pushpa, B. (2020, March). Hybrid data encryption algorithm for secure medical data transmission in cloud environment. In *2020 Fourth international conference on computing methodologies and communication (ICCMC)* (pp. 329-334). IEEE.
15. Abdelfattah, S., Baza, M., Badr, M. M., Mahmoud, M. M., Srivastava, G., Alsolami, F., & Ali, A. M. (2021). Efficient search over encrypted medical data with known-plaintext/background models and unlinkability. *IEEE Access*, *9*, 151129-151141.
16. Vizitiu, A., Niţă, C. I., Puiu, A., Suciu, C., & Itu, L. M. (2020). Applying deep neural networks over homomorphic encrypted medical data. *Computational and mathematical methods in medicine*, *2020*(1), 3910250.
17. Alzubi, J. A., Alzubi, O. A., Beseiso, M., Budati, A. K., & Shankar, K. (2022). Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis. *Expert Systems*, *39*(4), e12879.
18. Pandey, H. M. (2020). Secure medical data transmission using a fusion of bit mask oriented genetic algorithm, encryption and steganography. *Future Generation Computer Systems*, *111*, 213-225.
19. Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*.
20. Bost, R., Popa, R. A., Tu, S., & Goldwasser, S. (2014). Machine learning classification over encrypted data. *Cryptology ePrint Archive*.
21. Bokhari, M. U., Shallal, Q. M., & Tamandani, Y. K. (2019). Reducing the required time and power for data encryption and decryption using K-NN machine learning. *IETE Journal of Research*, *65*(2), 227-235.
22. Jäschke, A., & Armknecht, F. (2018, August). Unsupervised machine learning on encrypted data. In *International conference on selected areas in cryptography* (pp. 453-478). Cham: Springer International Publishing.
23. Fazio, N., Gennaro, R., Jafarikhah, T., & Skeith, W. E. (2017). Homomorphic secret sharing from paillier encryption. In *Provable Security: 11th International Conference, ProvSec 2017, Xi'an, China, October 23-25, 2017, Proceedings 11* (pp. 381-399). Springer International Publishing.
24. de Campos Souza, P. V. (2020). Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied soft computing*, *92*, 106275.
25. Brammya, G., Praveena, S., Ninu Preetha, N. S., Ramya, R., Rajakumar, B. R., & Binu, D. (2019). Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm. *The Computer Journal*, bxy133.