Homomorphic Encryption on Deep Learning in accurate prediction of Brain Tumour

Arafat Sarwar
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
arafat.sarwar@g.bracu.ac.bd

MD. Shakhawat Hossain
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
shakhawat.hossain@g.bracu.ac.bd

Risum Ahmed Bhuiyan
Computer Science and Engineering
BRAC University)
Dhaka, Bangladesh
risum.ahmed.bhuiyan@g.bracu.ac.bd

Tanun Mahmud
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
tanun.mahmud@g.bracu.ac.bd

Shakila Zaman
Computer Science and Engineering
University of North Texas
Texas, USA
shakila.zaman@my.unt.edu

Muhammad Iqbal Hossin
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
iqbal.hossain@bracu.ac.bd

2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM) | 979-8-3503-1600-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/NCIM59001.2023.10212646

***Abstract*—**
Data breach and data theft has become a repulsive problem in the digital world and medical data is the most vulnerable in this case as it contains numerous information of patients which can hinder a patient's personal life. Moreover, due to patients' rights and confidentiality, hospital's are unwilling to share the medical data to third parties. Also, Brain tumour is one of the most occurring disease as everyday, around the globe vast amount of MRI image are taken. Many existing models have been introduced throughout the decades to detect brain tumors, mostly based on Neural Network models. To secure the MRI images and protect patient rights, we have worked on a technique based on Partial Homomorphic Encryption (PHE) which produces encrypted images while keeping most of the features intact. As a result, existing NN models can produce a higher accuracy by extracting features from the encrypted images and also reduces the time complexity of established Paillier cryptosystem. We were able to achieve max 82% accuracy using a VGG-19 model. Using our proposed model we can safely and securely read encrypted medical image data via our PHE method while being efficient enough to be used on a mass scale in the medical industry.

***Index Terms*—Deep Learning, Neural Networks, Homomorphic encryption, VGG16, VGG19, Paillier Cryptosystem, ResNet50**

I. Introduction

In the medical field, confidential information are crucial. The medical industry must create a trustworthy environment and a sense of privacy to entice patients to use services, but data theft can endanger patients lives. Our research aims to encrypt medical data using Paillier Cryptosystem, allowing hospitals to share any data without fear of data leakage. It also allows detection characteristics from encrypted data-sets using NN. Finally, merging with neural networks, our Homomorphic model can offer better accuracy in the detection of encrypted data. To completely secure and protect image detection, we created a Homomorphic encryption model which will enable businesses and people to use neural networks in the future.

In the medical industry, the use of medical images helps to speed up the detection process and saves the crucial diagnostic process. MRI offers the best, highest-resolution images out of all the medical imaging technologies, and CNN models like VGG16, VGG19, ResNet50, AlexNet and others are readily available. Medical images are more detailed and data-rich than other types of images, and each and every pixel is crucial for accurate detection. Sharing medical photos has become common due to the influence of the internet and the modernization of the medical industry.

We developed an encryption system based on the Paillier cryptosystem, adapting the Partial Homomorphic Encryption technique. Our model reduces complexity and accelerates image processing while maintaining high detection and prediction accuracy for encrypted photos.

Patients are wary of sharing medical records, while institutions lack resources for ML and NN models. Obtaining suitable data is challenging, and inadvertent disclosures to third parties lead to data loss and theft [1]. Despite privacy laws in various countries, third parties exploit loopholes to access personal data [2]. Thus data loss from the medical sector is still prevalent. As a result, trust between patients and the health industry is still frail.

The goal of this effort is to create a secure medical data analysis system that will allow the sharing of medical image data without concern for data theft or misuse.

The contributions of this work are listed below -

- Allow identification of characteristics from datasets and use of ML and NN on the encrypted data.
- Finally, merging with neural networks, our Homomorphic model can offer better accuracy in the detection of encrypted data

Here, we created a homomorphic encrypted model to offer total security and anonymity for image detection, enabling businesses and people to use neural networks in the future.

The paper includes a literature review (Section 2), our encryption technique (Section 3), and the superiority of our custom PHE model (Section 4).It also covers a literature review, our encryption technique, and the analysis of our custom PHE model's superiority.

II. Literature Review

A. *Homomorphic Encryption*

Homomorphic encryption enables computation on encrypted data without decryption, preserving privacy. Kim et al. (2022) achieved comparable accuracy with FHE [3].Homomorphic encryption enables secure computation and privacy-preserving modeling. Chen et al. (2021) developed FHE for machine learning, ensuring confidential patient EHRs and secure computation outsourcing. [4]. Partially homomorphic encryption (PHE) is an important part of machine learning.Wang et al. (2021) used PHE-based training to train a machine learning model for predicting COVID-19 infection using CT images, providing similar accuracy to a non-encrypted model while preserving CT image privacy. [5]. Additionally,Mohassel et al. (2019) conducted a study using PHE to encrypt raw images and CNN weights, and use a PHE-based prediction scheme to make predictions on fresh encrypted images. The results of the study were as accurate as a baseline model that wasn't encrypted. [6].
Moreover, homomorphic encryption (FHE) is used to securely aggregate model changes in federated learning, as demonstrated by Yang et al. (2020). [7]. FHE-based aggregation encrypts model updates without decryption, producing accuracy comparable to a benchmark model that isn't encrypted.
In addition, deep learning uses homomorphic encryption, differential privacy, secure multi-party computation, and functional encryption to protect data.

B. *Differential privacy*

is a technique used to protect data in deep learning, such as in Abadi et al. (2016) to train a deep learning model for speech recognition while preserving the privacy of the training data. [8].

C. *Secure multi-party computation*

It protects sensitive information in training data while allowing multiple parties to collaborate and analyze it [9]. developed a deep learning model for face recognition to protect sensitive information in training data while allowing multiple parties to collaborate on the training process. [10].

D. *Functional encryption*

is a technique that allows computation on encrypted data while preserving the privacy of the data. Kamm et al. (2020) used functional encryption to protect the privacy of a deep learning model for medical diagnosis without revealing the underlying data. [11].
Furthermore, one other technique which is the unsupervised learning is capable at detecting brain cancers.

E. *Unsupervised learning*

is a form of Machine learning detection where the algorithms can classify brain MRI images, analyze values, and cluster untagged datasets.K-means clustering, ICA, SOM, and unsupervised learning were used with an accuracy of 98.6% to accurately visualize brain segments from tumors and healthy brain tissue, with the goal of identifying grey brain tissue [12]. The Superpixel method can, however, detect brain tumors.

F. *The superpixel technique*

Ren and Malik's technique groups pixels that naturally belong in an image to reduce complexity and make it computationally lighter. [13]. FLAIR is a technique used to detect brain tumor segmentation and detection, using ELA to classify and compare with SVM. The result is above 90%, achieved. [14].A study was conducted using multi-parametric superpixel to compare the accuracy of three classesifier applications: RF, SVM, and Adaboost. On balanced data, the accuracy was 82%, 82%, and 89%, while on unbalanced data, it was 82%, 79%, and 91%. [14].
CNN is a deep learning system for classifying images that uses flexible learning methods and low-moderate-high priority features.. [15]. Feature extraction was used by AlexNet and the Visual Geometry Group (VGG-16, VGG-19) to lessen motion sickness and enhance image clarity. With 3064 images divided into three categories, AlexNet's accuracy score was 99.55%, while R-CNN's accuracy was 91.66%. [16]. Although increasing key is not effective and affordable as it may ensure a bit more secury but also increases noise when decrypting. With FHE to encrypt image datasets and convert them to CIFAR-10 format, the group used NN and CNN models to forecast leukemia with 80% accuracy. [17].

III. Methodology

The total work is divided into two sub sections. First part explains from where and why the data-set was gathered and selected etc and in the next part we described our proposed model, the equation used, and the total encryption and decryption methods.

A. *Data-set Description*

The most important part in our work was to find a well organized data-set to apply our encryption method. We picked

TABLE I
LITERATURE REVIEW

Technique	Application	Methodology	Author	Year	Research Purpose
SVM	Medical diagnosis	Supervised learning	K. Wongsuphasawat	2014	Develop a system for medical diagnosis
CNN	Image classification	Supervised learning	A. Krizhevsky	2012	Improve accuracy in image classification
Random Forest	Fraud detection	Supervised learning	Y. Liu	2014	Identify fraudulent transactions
Clustering	Tumor identification	Unsupervised learning	A. Thirumurugan	2015	Identify tumor in brain images
Deep Learning	Privacy-preserving learning	Differential privacy	M. Abadi	2016	Develop a system for privacy-preserving deep learning
Federated Learning	Secure aggregation	Fully homomorphic encryption	Y. Yang	2020	Develop a system for secure federated learning
Adversarial Learning	Model prediction	Secure prediction	P. Mohassel	2019	Develop a system for secure prediction of deep learning models under adversarial corruption
Superpixels, Clustering	Image Segmentation	Unsupervised learning	Ankur Gupta	2019	Improve accuracy in image segmentation
Image classification	CNN	Face recognition	Wang et al.	2023	Developing a reliable and accurate face recognition system using deep learning techniques
Privacy preserving Neural Network	Medical Imaging	Privacy Preserving Neural Network	Khilji et al.	2019	Acute leukemia prediction using PPNN

our data set from Kaggle, where the data was divided into two categories, with 253 images. From the total images, 98 images are of 'no tumour' category and the rest are in 'yes tumour' category. Finally, we further added more images to train our model thoroughly.

B. Proposed Homomorphic Encryption Model

Firstly, we converted all the MRI it into a gray-scale, resized it into 244*244 pixels, and used partial Homomorphic Encryption to generate private and public keys. only public keys can be given to 3rd parties as public and private keys are both needed to decrypt the image to the original one. After generating the keys, next we divided the images for testing and training in a ration of 1:3. Finally, we ran the encrypted image through different Neural Network models and SVM models to determine the accuracy of prediction based on training.

Among the other Homomorphic encryption techniques, Paillier Cryptosystem is more simple in terms of generating keys and encryption. As a result, we decided to used PHE, as our encryption technique. So, all the equations and calculations below for generating keys, encryption and decryption methods are based for Paillier Cryptosystem.

To generate keys, first, we took two large random prime numbers independently of the same binary size which was in our case a bit length of 15. This length needs to be higher because if we use lower length numbers, a lot of noise gets added while decrypting the data. Next, the following requirement must be met.

- GCD(p×q, (p−1)×(q−1)) = 1, where GCD means Greatest Common Divisor. It primarily ensures that two generates numbers are prime numbers.
- To calculate n, n = (p×q) and λ = LCM(p−1, q−1) where LCM means Least Common Multiplier between two variables. This is done in our private class part which calculates the public and private key.
- Then g = n + 1 as two primes are of equal binary length. These parameters are calculated and importantly related to each other for decryption purposes.
- we calculated the modular multiplicative inverse. $\mu = \left(L\left((g^{\lambda}) \bmod n^2\right)\right)^{-1}$
- Here, L(x) = $(x\text{-}1) \div n$.
- At last, we got the public key (n, g) and private key (λ, μ).

1) Encryption Method:

- To encrypt the images, the pixel size is 'm' which must be follow 0 <= m < n, it is to ensure modulus constraint.
- Then we calculate a random value r where 0 <= r < n.
- Finally we compute the cipher-text as: c = $(g^m \times r^n) \bmod n^2$,

2) Decryption Method:

- To decrypt, every value of pixel 'c' is returned one by one to decryption algorithm.
- To decrypt each pixel, the following equation is used : m = L($c^{\lambda} \bmod n^2$) × μ mod n)

3) Image Encryption: To pre-process the image data, we have taken the whole image as a matrix and later changed the matrix into a NumPy matrix to use the to.list() method so that we can make our matrix into a list. After that, we set our encryption method different from the commonly known Paillier Cryptosystem method. In the Paillier Cryptosystem model, to encrypt the image, for every pixel a random number r is generated while encrypting data as mentioned before. But in doing so, the encrypted image pixels are so randomized that the features that it once held as the real image gets lost. As a result, if we send the encrypted image to any existing NN, it won't be able to detect the difference between a brain that has a tumor and one which does not.

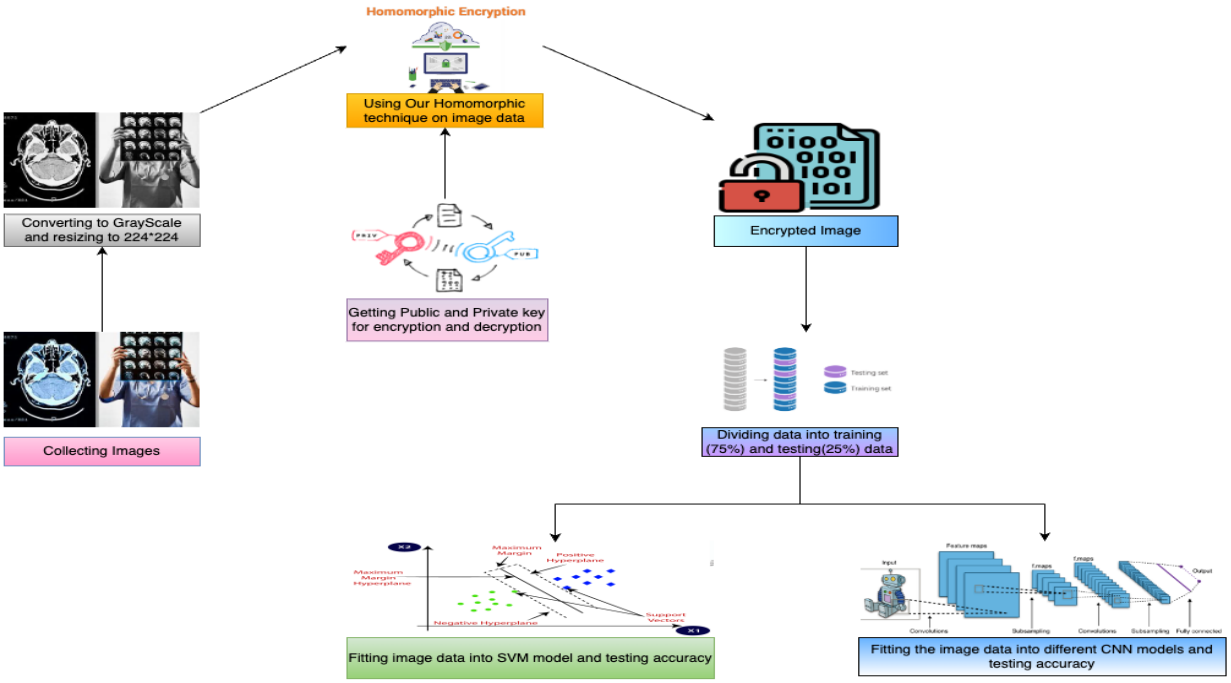

Fig. 1. Workflow of proposed model.

Inputs :
publickey: Public key used for Paillier Cryptosystem
image: input image
Outputs :
Cipherimage: encrypted image as cipherimage
# Generate a random value r such that gcd(e,n) = 1
r = random.radint(1, publickey.n-1)
while not xgcd(r.publickey.n)[0] == 1:
r = random.radiant(1, publickey.n-1)
# Compute the encryption of the image using the paillier Cryptosystem
a = pow(publickey.g, plainimage, publickey.nsq
b = pow(r, publickey.n, publickey.nsq
return cipherimage

Algorithm 1: Encryption method

To get the better from this pixel-related mess, we have encrypted every image with a random number each so that getting a list of encrypted images is possible with their features intact which will also provide higher accuracy using the old NN models. The benefits of Using a single random number for each encryption generates closer random numbers, making the encrypted images of brains with and without tumors look distinct. Thus the existing NN can determine the images correctly using their default weights. After that we send values starting from 0 to matrix size which is 224*224, meaning every image will go through the encryption method to encrypt the image. [**?**]

4) Encrypting the whole Image Data-sets: Firstly, we created a directory called Encrypted Brain in Google Colab where

Variable :
IMAGE_SIZE : Resizing image to 224
X : list of store encrypted image arrays
Y : list of store corresponding class labels
cls : if (an image contains **'Tumor'** cls : 1)
else cls : 0
For (each cls in classes)
For (each **i** in the list of files in the path directory)
Convert image to grayscale and resize it to 224
Encrypt using public key
If (cls is **'yes'**)
Appending numpy yes images in X
Appending class of X in Y
Converting numpy array into uint8 to save it as image
Saving the converted data in a folder which contains yes images
Else if (cls is **'no'**)
Appending numpy no images in X
Appending class of X in Y
Changing numpy array into uint8 to save it as image
Saving converted data into a no tumor folder
End If
End For
End For

Algorithm 2: Image reading, pre-processing and saving data

two folders were created for encrypted images, one for 'Yes Tumor', and other for the 'No Tumor' images. Also, we created two more folders in the content section for dividing our encrypted images into Train and Test data for later. The folders are named Image_Train_Validation, Image_Test_Validation.

Then using the in-built pathlib library, we read our image files from our drive and divided them into two classes. For the yes images, we have set the value 1 and for the no image, we set the value 0.

Before reading the images, we set all Image size to 224 as it is not mandatory to work with higher pixel size as it would increase the encryption time but will not contribute much in our detection part.

After that, we declared two arrays which are X and Y. X are the encrypted image array, and Y for yes: 1 or no : 0. After that we declared a loop that will iterate until we have read all the image files in our dataset. To read the images, we used the PILLOW library. From the PILLOW library, we imported PIL Image to read images in RGB and then convert the RGB images to Gray scale. After converting the images to grayscale, we resize our image to 224. Next, we send all the images one by one to the encryption method to encrypt our data. For the yes and no tumors, we save them in two formats, one is in jpg format using img.save() and another npy format using np.save(). Also, we appended the encrypted array image in X and their class type (zero or one) in Y. This was done for a regular CNN model because it's easier for the CNN model to work with arrays.

Finally, we sent the encrypted data into a regular CNN, VGG16, VGG19, ResNet50, SVM model for training, testing, and accuracy purposes.

IV. Result and Analysis

Normally FHE performs inefficiently despite having strong functionality and security [18]. Oppositely, PHE provides good security and efficiency, but its functionality is very constrained. So, our goal was to make a custom PHE model which will remain secured, efficient while maintaining a strong functionality also. Our PHE model, unlike conventional methods such as Paillier Cryptosystem, generates a random number for each pixel, resulting in encrypted images that retain structural information. Tumor regions, appearing as white circles, exhibit slightly higher values in the encrypted image array. This encryption scheme provides strong protection while enabling feature extraction and accurate prediction scores on encrypted images. Also, In the Paillier Cryptosystem model,

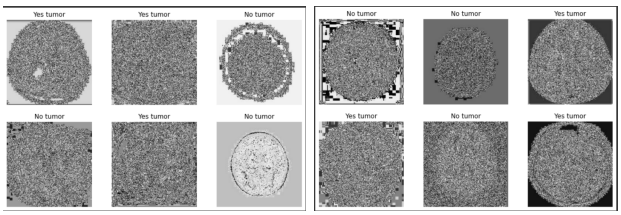

Fig. 2. Our Encryption Method

the encryption time takes longer due to the same reason of the random number being generated for each pixel, then for an image with 224×224 pixels, it takes $\Omega(n2)$ times to encrypt an image file. Our encryption scheme is faster than regular Paillier, In our encryption model random number is generated once for each image. This as a result, reduces the time complexity to $\Omega(n)$, which is less than the half time it took from the original existing model.

Here, we are comparing our encryption model (on the Right) with Paillier Cryptosystem model (on the Left)

A. SVM

In Machine Learning algorithms like SVM, we fitted the Numpy array and got the training accuracy to be exactly 100 percent, however, in the testing score, it was around 66 percent. Thus, we concluded, this is due to the over-fitting issue, as it is unexpected to get this much higher value only in training data from a Machine Learning algorithm.

B. VGG-16

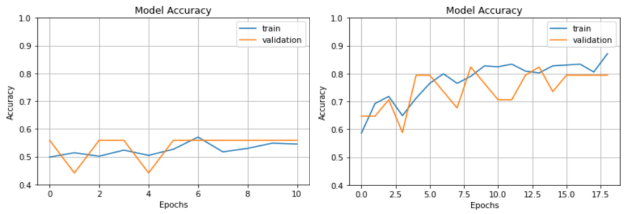

Fig. 3. Here, we are seeing by using the VGG16 model we get the training accuracy of our encrypted data is around 88 percent, and a validation accuracy of 80 percent around only 18 epochs. we can already see a better result and input compared with the traditional HE encryption.

C. ResNet-50

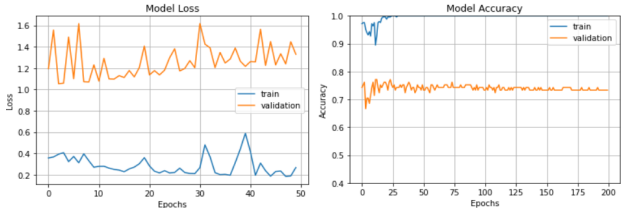

Fig. 4. Here, in ResNet-50, out of 200 epochs it has reached 100% accuracy in 25 epochs in training and got 75% accuracy throughout validation period, which is a quite decent result.

D. VGG-19

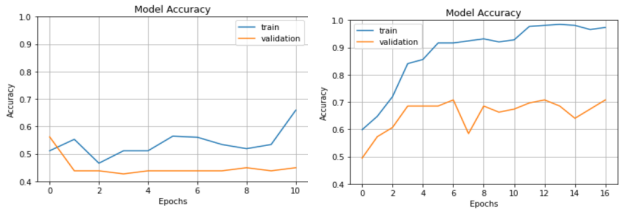

Fig. 5. Here also, in the shown graph, we can see the accuracy is better than using the normal HE. In only about 16 epochs, we have got an accuracy rate of above 95 percent, which is a great improvement over the previously tested models. We also got a validation accuracy of around 70 percent. The validation accuracy can be further increased if we can use a larger data set than the ones we are using now.

E. Training and Validation accuracy of NN and ML models

Table II and III shows the training accuracy and validation accuracy between our encrypted model and the Paillier-crypto model.

Training Accuracy	VGG16	ResNet50	VGG19	CNN	SVM
Paillier-crypto	58%	78%	68%	62%	100%
Our PH-Encryption	89%	100%	98%	70%	100%

TABLE II
TRAINING ACCURACY

Validation Accuracy	VGG16	ResNet-50	VGG19	CNN	SVM
Paillier-crypto	57%	60%	54%	54%	61%
Our PH-Encryption	82%	75%	71%	55%	76%

TABLE III
VALIDATION ACCURACY

From the Table III, VGG16 has the highest validation accuracy. VGG16 is a Neural Network model so the result is satisfactory. VGG19 and ResNet50 have better validation accuracy due to their weighted layers.

V. CONCLUSION

Our goal is to create a secure IoT system for encrypted brain tumor detection, enabling third-party use of encrypted image data while ensuring patient privacy and enhancing healthcare services.

- In forthcoming times, the implementation of our proposed model has the potential to significantly bolster the fortification of sensitive medical data.
- In the future, researchers and developers can utilize our framework as a foundation for creating new neural networks.
- This method applies to various domains like voting systems and defense, ensuring strong protection against unauthorized data acquisition.
- It will create a strong defense against unauthorized access and breaches, covering various areas of application.

In conclusion, our proposed IoT security system, detecting brain tumors with encrypted patient data, fortifies medical data, fosters neural network advancements, and provides multi-domain protection.

REFERENCES

[1] J. George and T. Bhila, "Security, confidentiality and privacy in health of healthcare data," *International Journal of Trend in Scientific Research and Development*, vol. Volume-3, 06 2019.

[2] "Patient Privacy Data Protection: Different laws around the world," https://www.brainlab.com/journal/patient-privacy-data-protection-different-laws-around-the-world/, jan 13 2021, [Online; accessed 2023-01-22].

[3] S. Kim, Z. Zhang, C. Liu, L. Huang, Y. Wang, and D. Song, "Secure prediction of deep learning models using fully homomorphic encryption," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2022, pp. 4834–4843.

[4] Y. Chen, H. Zhu, and D. Wu, "Privacy-preserving heart disease prediction with fully homomorphic encryption," *IEEE Transactions on Medical Imaging*, vol. 40, no. 8, pp. 2235–2246, 2021.

[5] Z. Wang, J. Xu, and H. Liu, "Privacy-preserving deep learning for covid-19 diagnosis using chest ct images," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 6, pp. 2027–2036, 2021.

[6] P. Mohassel, Y. Zhang, and A. Raghunathan, "Secure prediction of deep learning models under adversarial corruption," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2019, pp. 1193–1208.

[7] Y. Yang, Y. Liu, X. Song, and X. Huang, "Federated learning with secure aggregation using a fully homomorphic encryption scheme," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3435–3447, 2020.

[8] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, 2017, pp. 3–18. [Online]. Available: https://doi.org/10.1109/SP.2017.11

[10] ——, "Privacy-preserving deep learning," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2015, pp. 1310–1321.

[11] L. Kamm and D. Song, "Secure medical diagnosis using functional encryption," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8294–8302.

[12] V. Govindaraj, M. Rajasekaran, P. Subbaraj, and A. Vishnuvarthanan, "An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images," *Applied Soft Computing*, vol. 38, 10 2015.

[13] "Image segmentation, superpixels, and clustering," https://cran.r-project.org/web/packages/OpenImageR/vignettes/Image_segmentation_superpixels_clustering.html, accessed: March 26, 2023.

[14] S. Shah, Y. Huang, S. Suresh, and et al., "Non-invasive pressure difference mapping shows excellent correlation with invasively measured pulmonary vascular resistance in children with pulmonary hypertension," *International Journal of Cardiovascular Imaging*, vol. 33, no. 1, pp. 77–85, 2017.

[15] A. Gupta. (2019, January 7) Using convolutional neural network for image classification. [Online]. Available: https://towardsdatascience.com/using-convolutional-neural-network-for-image-classification-5997bfd0ede4

[16] L. Wang, H. Wei, J. Chen, X. Xie, Y. Xu, and D. Zhang, "A robust and effective face recognition system using deep convolutional neural networks," *Applied Sciences*, vol. 13, no. 6, p. 3680, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/6/3680

[17] I. Q. Khilji, K. Saha, J. A. Shonon, and R. Israq, "Prediction of acute lymphoid leukemia using privacy preserving neural network," Dec 2019. [Online]. Available: http://dspace.bracu.ac.bd/xmlui/handle/10361/15076

[18] "What is fully homomorphic encryption?" Apr 2021. [Online]. Available: https://inpher.io/technology/what-is-fully-homomorphic-encryption/