

# Homomorphic Encryption on Deep learning in accurate prediction of Brain Tumour

by

Arafat Sarwar

19101607

Shakhawat Hossain

19101611

Risum Ahmed Bhuiyan

19101556

Tanun Mahmud

19101593

A thesis submitted to the School of Data and Sciences  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science and Engineering

School of Data and Sciences

Brac University

January 2023

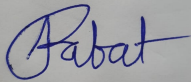
© 2023. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

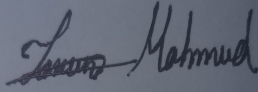
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:



---

Arafat Sarwar  
19101607




---

Tanun Mahmud  
19101593



---

Shakhawat Hossain  
19101611



---

Risum Ahmed Bhuiyan  
19101556

# Approval

The thesis/project titled “Homomorphic Encryption on Deep learning in accurate prediction of Brain Tumour” submitted by

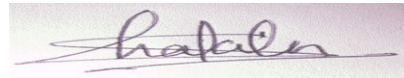
1. Arafat Sarwar (19101607)
2. Shakhawat Hossain (19101611)
3. Risum Ahmed Bhuiyan (19101556)
4. Tanun Mahmud (19101593)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on January 25, 2023.

## Examining Committee:

Supervisor:

(Member)



---

Shakila Zaman  
University of North Texas

Program Coordinator:

(Member)

---

Dr. Md. Golam Rabiul Alam  
Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:

(Chair)

---

Sadia Hamid Kazi  
Chairperson and Assistant Professor  
Department of Computer Science and Engineering  
Brac University

# Abstract

The brain is the most complicated organ that manages every bodily function as well including intellect, memory, emotion, taste, motor skills, vision, respiration, temperature, and appetite. Any type of disease or damage can obstruct the function of the brain and can change the daily lifestyle of a person in an instant or gradually. One of those diseases is a brain tumor, which is hard to detect as serious symptoms start to develop in the later stages of the disease. There are mainly non-cancerous and cancerous brain tumors To make it easier to detect brain tumors we have used the existing Neural Network model to identify tumors. Our objective is to keep patient data confidential as medical institutions are not willing to share patient information due to patients' rights. And so we have integrated our own Homomorphic encryption so that existing NN models can work and detect tumors from encrypted image datasets. Different Deep Learning and Neural Network techniques can enhance the tumor identification process. In our paper, we have built our custom-made Partial Homomorphic Encryption (PHE) which is based on Paillier Cryptosystem to encrypt the medical data and then used pre-built Neural Networks models (VGG16, VGG19, ResNet50) have been chosen to execute on the data-set consisting of encrypted images of different types of tumors. We have taken the characteristics from the encrypted photos of these brain tumors and extracted them using a pre-trained deep CNN model. First, we have used different Machine Learning algorithms and neural networks in deep learning to classify the images into two categories. Then, we compared the accuracy of various models to identify which algorithm performs the best. For our research, we have created a combined dataset by collecting images of the diseases mentioned above from different sources and applying data augmentation to them. Using our proposed model we can safely and securely read encrypted medical image data via our Partial Homomorphic Encryption method while being efficient enough to be used on a mass scale in the medical industry.

**Keywords:** Brain tumor, benign, malignant, Deep Learning, Neural Network, Homomorphic encryption, Convolutional Neural Network(CNN), VGG16, VGG19, Paillier Cryptosystem, Support Vector Machine(SVM), ResNet50

## **Dedication**

We would like to dedicate this paper to the Almighty, our family, our friends, and our allies.

# Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Shakila Zaman and co-advisor Muhammad Iqbal Hos-sain, PhD for there kind support and advice in our work. They helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Dedication</b>	<b>iv</b>
<b>Acknowledgment</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Nomenclature</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 Problem Statement . . . . .	3
1.3 Research Objective . . . . .	3
1.4 Thesis Structure . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Unsupervised Learning . . . . .	6
2.2 Super-pixel Technique . . . . .	6
2.3 Common Neural Network Model . . . . .	6
2.4 Related Works . . . . .	7
<b>3 Background Studies</b>	<b>9</b>
3.1 CNN . . . . .	9
3.2 Homomorphic Encryption . . . . .	9
3.3 SVM . . . . .	10
3.4 VGG-16 . . . . .	11
3.5 VGG-19 . . . . .	12
3.6 Res-Net50 . . . . .	12
<b>4 Methodology</b>	<b>14</b>
4.1 Dataset Description . . . . .	14
4.2 Our Homomorphic Encryption Model . . . . .	15
4.2.1 Decryption . . . . .	16

4.2.2	Encrypting and Decrypting the Images . . . . .	17
4.2.3	Encrypting the whole Image Data-sets . . . . .	18
<b>5</b>	<b>Experimentation and Result Analysis</b>	<b>20</b>
5.1	Comparison between Paillier Cryptosystem and Our FHE in terms of accuracy . . . . .	21
5.1.1	Accuracy based on Paillier Cryptosystem . . . . .	21
5.1.2	Accuracy based on our Homomorphic Technique . . . . .	22
	<b>Conclusion</b>	<b>26</b>



# List of Figures

3.1	CNN Network . . . . .	9
3.2	VGG-16 Architecture . . . . .	11
3.3	VGG-19 Architecture . . . . .	12
3.4	CNN Network . . . . .	13
4.1	Total Workflow . . . . .	15
4.2	Encryption Method . . . . .	16
4.3	Decryption Method . . . . .	17
4.4	Encrypting a whole image . . . . .	17
4.5	Decrypting a whole image . . . . .	18
4.6	Reading images, Preprocessing and Encrypting the data-sets . . . . .	19
5.1	Our Encryption Method . . . . .	21
5.2	Accuracy of Each Models (CNN,ResNet50,VGG16,VGG19,SVM) . . . . .	22
5.3	Training and validation accuracy graph for CNN . . . . .	22
5.4	VGG16 Training accuracy vs Validation accuracy graph . . . . .	23
5.5	VGG19 Training accuracy vs Validation accuracy graph . . . . .	23
5.6	ResNet50 Training accuracy vs Validation accuracy graph . . . . .	24
5.7	SVM training and testing score . . . . .	24

# List of Tables

5.1	Training Accuracy . . . . .	25
5.2	Validation Accuracy . . . . .	25

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

*CNN* Convolutional Neural Network

*FHE* Fully Homomorphic Encryption

*HE* Homomorphic Encryption

*MRI* Magnetic Resonance Imaging

*NN* Neural Network

*PHE* Partial Homomorphic Encryption

*R – CNN* Regions with Convolutional Neural Network

*SHE* Somewhat Homomorphic Encryption

*SVM* Support Vector Machine

*VGG16* Visual Geometry Group with 16 layers

*VGG19* Visual Geometry Group with 19 layers

# Chapter 1

## Introduction

The most important and time-consuming process for tumor identification is that used to find brain tumors. The identification of brain tumors is exceedingly challenging [4]. In the medical industry, the use of medical images helps to speed up the detection process and saves the crucial diagnostic process. The analysis of anomalies using MR images enables the detection of hidden brain illnesses. Other medical images exist as well, including MRI, CT scans, and X-rays. These pictures are helpful in identifying any type of brain issues or anomalies[5].

MRI offers the best, highest-resolution images out of all the medical imaging technologies. The original image is used as the input for the Deep Learning algorithm, which then outputs the results[35]. CNN offers a variety of convolutional layers in between several Deep Learning models to extract characteristics from the images. CNN functions better when handling large datasets. CNN models like VGG16, VGG19, ResNet50, AlexNet and others are readily available. These models identify visual data and use it to determine how to present it. These are therefore detectable and can be used for such. They can also use a tiny dataset to use their previously trainable model. It benefits the medical field because medical datasets aren't often particularly huge in size[34]. High accuracy on small datasets can be achieved with this strategy[12].

In order to identify a patient's disorders today, medical photographs are frequently used. Modern life makes extensive use of medical pictures like MRI, CT scans, and X-rays. Sharing medical photos has become common due to the influence of the internet and the modernization of the medical industry, which contains a lot of information about the patients for treatments[6]. Because of this, medical data must be protected with appropriate privacy and security measures. In the absence of such measures, the patient's information may be taken from the photos and utilized for any bad purposes. Medical images are more detailed and data-rich than other types of images. In addition, each and every pixel is crucial for accurate detection. Incorrect diagnosis results may be obtained from the entire set of images if there is even one incorrect pixel[30].

Homomorphic encryption performs exceptionally well in terms of data security. You can compute the encrypted data using this technique. Homomorphic encryption is a cryptosystem that uses a public key to encrypt the data and the same pri-

vate key for the decrypted data for specific individuals. This encrypted system's final destination performs an endless amount of additions and multiplications of encrypted data. As if the identical procedures were carried out on pre-existing plain texts and the output was encrypted, the outcome should be cipher-text[33]. Three main sorts of operations can be distinguished based on the various types and frequencies of mathematical operations that are carried out on ciphertext. These are Partial Homomorphic Encryption, Somewhat Homomorphic Encryption, and Fully Homomorphic Encryption. Without the usage of the encryption key, FHE enables anyone to read and utilize encrypted data[42]. PHE is used to allow a particular kind of encrypting operation to run indefinitely many times[33]. SHE uses public key encryption to encrypt data and supports a finite amount of additions and multiplications. It aids in protecting and guarantees the security and privacy of the data.

To secure the appropriate security and privacy of the medical picture data, we chose this as our research's objective. For the purpose of creating a novel encryption system, we attempted to adapt the partial Homomorphic encryption technique. We can attain great forecast accuracy and lower time complexity by doing this. Image encryption is often possible using a Homomorphic approach. Additionally, image processing models have the ability to deliver improved accuracy on real image data. The accuracy, however, was not very great in earlier studies when these two systems Homomorphic encryption and image processing—were combined. That's why we concentrated on finding a solution. On the other hand, a completely Homomorphic system is exceedingly complex because each and every pixel is given a large number of random integers. It resulted in a less accurate and unsatisfactory prediction. We built our model to reduce the complexity in order to solve this problem. We can shorten the time needed for image processing in our model. because each image receives a single random number. Additionally, the tumours' areas offer various benefits when the photos are encrypted at that point. As a result, when neural networks or image processing systems are used, they may quickly detect cancers by computing these various values. By doing this, our approach can ensure that encrypted photos are detected more quickly and with higher detection and prediction accuracy.

In today's world, most encryption technique provides security to sensitive data, and the NN model can give accurate predictions in the raw images. However, an encryption method with decent prediction results is rare to find. The motivation of our work is to make a secure medical data analysis system where medical image data can be shared without any fear of data theft or any sort of misuse in the industry. As all the data has been encrypted, no one except the user/authorities can decipher the encrypted data. So, our objective is to make an encryption model where decryption is not required on the image to run on different detection NN models which will ensure security as well as can deliver a good prediction score based on the encrypted image sets.

In the paper, first of all, we included an Introduction to generalize our topic. Then, we added a background study to show the types of existing NN models we implemented. Next, we followed by analyzing Literature Review, where we described various existing works related to ours which were collected from internet sources,

google scholar journals, and articles. Then we imposed our encryption technique in the methodology part where we described the dataset and algorithm. Then, in Result and Analysis section, we have described why our custom PHE model works better than the existing FHE and PHE models. Finally, we concluded our paper by highlighting our findings and future works.

## 1.1 Motivation

Data security is one of the hot topics in today's world. Medical images are the most intimate and delicate section in the security section. But research on encrypted medical images and diagnosing diseases at the same time are rare to find in this field. An algorithm with security and accurate prediction of diseases from encrypted data is rarely sighted because most of the time while encrypting parameters value that differentiates a behavior in pictures is lost due to encrypted processes and as a result, prediction cannot be effectively done on encrypted data using neural networks. Since better disease detection cannot be done using predictions made using encrypted data. Our motivation is to work in this area. So that our efforts and system can be used to fill this gap. Last but not least, our aim is to improve encryption data detection with accurate prediction.

## 1.2 Problem Statement

Patients are unwilling to share their medical data with medical professionals for in-depth evaluation of the data through third parties as most hospitals lack proper IT professionals to train their own NN or ML models. Getting hand on medical data that can be used to train ML and NN models for the detection of tumors as well as convincing patients to rely on the ML and NN models for tumor detection is extremely difficult. We know that routine and unintentional disclosures of patient data to third parties are the most prominent concerns to data loss and theft [20]. Though different countries in the world have introduced different laws to maintain the privacy of data, third parties have managed to use a number of loopholes to obtain such data [31]. Thus data loss from the medical sector is still prevalent. As a result, trust between patients and the health industry is still frail.

## 1.3 Research Objective

Our research aims to encrypt the medical data using Paillier Cryptosystem allowing patients to share any medical data without the fear of data leakage. Moreover, our system allows identification of characteristics from a data and use of ML and NN on the encrypted data.

There are a lot of confidential data and information in the medical area. To entice the patient to use medical services, the industry must first create a trustworthy environment and a sense of privacy. However, data theft and the selling of private information have become major problems, as those leaked data can be executed to

endanger people's lives. We developed a Homomorphic encrypted model to address all of these issues, allowing for complete security and privacy when doing detection.

Our concept allows for the encryption of medical images as well as the detection of the specific characteristic of the encrypted data. People will be able to implement various neural networks in the future utilizing our concept because image identification is a task better left to neural networks. By merging with neural networks, our Homomorphic model can offer better accuracy in the detection of encrypted data.

## 1.4 Thesis Structure

Chapter 1: Introduction where we have discussed motivation, problem statement, objectives, and contributions.

Chapter 2: In the literature review, we analyzed the previous work that is related to ours. We also described Homomorphic Encryption and how we selected its parameters and the encryption algorithms. Finally, The source of all papers we talked about are mostly from Google scholar and some from Internet Articles.

Chapter 3: Here, we have shown the previous research that we implemented in our work. Here, we briefed about Neural Networks and different architectures that we used in our system

Chapter 4: We presented our proposed model in methodology and provided a workflow diagram. We also provided a description data-set about how we pre-processed data and also performed feature selection. We also described our model in detail.

Chapter 5: We discussed our system performance and analyzed our results.

Chapter 6: Here, we briefly discuss and concluded our results and lastly talked about our plan for future works.

# Chapter 2

## Literature Review

Many photographs pertaining to healthcare are sent via open networks every day. These pictures can reveal sensitive patient information. However, a number of security attacks could be perpetrated on these medical images [[14], [15]]. As a result, numerous medical picture encryption models were put into practice.

Ding et al. created a deep learning-based algorithm for medical image encryption [1]. The photos were encrypted using a cycle-generative adversarial network (CGAN). Khedr and Glenn created a homomorphic encryption model that is GPU-accelerated [2]. This model can deliver encryption results quickly. A verified multi-keyword search (VMKS) encryption model was created by Liu et al. [37]. Medical picture anonymized key generation has been used. Electronic health records were scrambled using a convergent key.

Using Paillier and ElGamal cryptosystems (PECs), Yi et al. implemented statistical analysis on healthcare data [13] without jeopardizing patient privacy. Haddad et al. presented the JJJ joint watermarking encryption technique for medical photos [27]. The data was also encrypted with JPEG-LS bit substitution watermarking modulation. By combining a selective encryption model (SET) with fragmentation and dispersion, Qiu et al. created a secure communication model [28].

Somewhat homomorphic encryption (SHE) was used by Jiang et al [21] for homomorphic evaluation across one instruction of many data. Data can be encrypted with fewer overheads because of this. In order to encrypt the patient information, Bao et al. created a renewable, confidential data-sharing approach with a search query [36]. A pseudo/fake signature approach also was utilized to ensure the accuracy of the information.

1. Ciphertext-only: In this approach, cryptanalysts attempt to decrypt ciphertext in order to obtain the private key or plain text. Only a few combinations of cipher texts are available to them.
2. Known-plaintext: In this assault, the attacker attempts to discover the encryption's secret key while also having some knowledge of the related plaintext and ciphertext.
3. Known-plaintext: In this approach, the attacker chooses their chosen randomized raw images and puts it into an encryption algorithm, giving the matching



cipher image a useful analysis tool.

4. Noise: In this case, the attacker tries to saturate an encrypted image with noise in order to obliterate the informational value of the plain image. As a result, the intended user's effort to retrieve the raw image file following the decryption process is now unsuccessful. [40]

## 2.1 Unsupervised Learning

Brain tumors can be detected by unsupervised learning. Unsupervised learning is a form of machine learning detection algorithm where the algorithm can classify the different parts of a brain MRI image it can analyze the different values of the image and can cluster the untagged dataset[43]. Using image-preprocessing which includes different techniques like histogram equalization, noise filtering, etc, then extracting the feature using Independent Component Analysis (ICA) and merging with Self Organised Mapping (SOM) and k-means clustering an accuracy of 98.6% was achieved [9]. Another paper had a different approach using unsupervised learning by only segmenting the brain from a healthy image and an image that has a tumor. The total work was divided into two sub-parts. In the first part, the limitation was set to identify grey brain tissue, then slicing the brain from the rest of the unnecessary parts was done in the second part and lastly using the unsupervised technique, the detailed segmentation of the brain image was completed [3].

## 2.2 Super-pixel Technique

Another way of detecting brain tumors is using the Superpixel technique. The superpixel technique is mainly used to make algorithms more feasible that are computationally heavy while working with image datasets. As a result, Ren and malik introduced this technique which groups a certain amount of pixels that naturally belong in an image, which reduces the complexity of the image as well as makes the image computationally lighter[38]. Implementing this technique, for brain tumor segmentation and detection, applying several other methods such as Fluid- Attenuated Inversion Recovery (FLAIR) to calculate the feature of each pixel, Extremely Randomized tree (ERT) to classify and compare with SVM, a result of above 90%, was achieved [17]. Similarly to the previous paper, another research was done, using the multi-parametric superpixel technique, applying different classification technique like Random Forest (RF) classifier, SVM classifier, and Adaboost on FLAIR images, On balanced data, precision was 85%, 82%, and 89% and on unbalanced data, precision was 82%, 79%, 91% was found for RF, SVM and Adaboost respectively [25]. Here, balanced data is after applying the super-pixel method, normalization, and feature selection.

## 2.3 Common Neural Network Model

Neural Networks (NN) models are algorithms based on how humans think. It can be said as a simplified version of human neurons by how the brain stimulates and processes information. Neural Network models are divided into layers. The layers

are divided into mainly 3 parts: Input layers, hidden layers and output layers. The input layers are for inputting the target, hidden layers are for transferring weight or values that are used for detection or segmentation and finally, the output layer is used for getting the desired output [41].

Convolutional Neural Network (CNN) is a popular deep learning model used mainly for image classification. CNN can group pictures into classes using adaptive learning methods in a low to high prioritizing feature[19].

Using Region-based Convolutional Neural Networks which are Visual Geometry Group (VGG-16, VGG-19) and AlexNet. This work was divided into two parts which are localization and detection. For preprocessing, filtering was used to reduce noises and enhance image quality, then used feature extraction and lastly for training, testing, and evaluating purposes, again feature extraction was done to finally test the model. In the above-mentioned model, AlexNet was able to provide the best result with an accuracy rate of 99.55%.

Another work related to a faster R-CNN model was with a data set consisting of 3064 images and those images were divided into 3 different categories namely Glioma, Meningioma, and Pituitary. In their work, they successfully classified among these sub-classes with 91.66%[26].

From various works of literature, it is observed that the development of an efficient encryption approach for healthcare is a challenging problem. Increasing the key size is highly desirable. Therefore, the high-dimensional map can be designed to increase the key size. However, due to an upsurge in the processing power, it is not as efficient and inexpensive as the medical industries were seeking. Here, varieties of models and algorithms were introduced for encrypting sensitive images on cloud servers using different encryption schemes and detecting brain tumors based on CNN throughout the decades. One thing common in all those techniques is that they can analyze and label tumors from non-tumor using raw images, yet none were built to work on encrypted data.

## 2.4 Related Works

Before beginning and developing our research, we conducted some prior research to learn more about the approaches and algorithms we would employ. An author has demonstrated from relatively recent research that they feature a step length and a filter to indicate how many pixels will be added in between calculations as well as the size of the partial images they will be taking into account. So the image's proportions have been severely reduced.

After that follows the pooling layer. Depending on the application, the main distinction between this and the convolution layer is that the latter only takes the average or maximum value from the outcome. By doing this, tiny details that are crucial to finishing the task are saved in a few pixels.

There are no gaps in the final layer. Now that the size of the image has been greatly reduced, they can use the closely-meshed layers. The many sub-images are again

joined to establish the connections and finish the classification**r5**.

Another author showed that, in order to identify particular characteristics in the data-sets, the images category encompasses extracting features from the image. Since the trainable factors get to be very huge, using an ANN for picture segmentation may finish up being highly expensive in terms of computation**r2**. CNNs are filtered when we use them. According to their intended use, filters come in a wide variety of sorts. By creating a local connection pattern between neurons, filters allow us to take advantage of an image's spatial localization. in this case convolution generally includes multiplying two inputs point-wise to create a quantifiable result. According to a different author's security examination of Homomorphic encryption,

The known-plain-text assault, the cipher text-only attack, the quantitative attack, and numerous brute force attacks are only a few examples of known attacks that can be thwarted by a strong encryption scheme. The suggested Homomorphic image Cryptosystem's security is examined for digital images when subjected to the differential, statistical, and brute-force attacks. The suggested Homomorphic picture of Cryptosystem's security from a fundamentally Cryptographic perspective will be demonstrated. The outcomes demonstrate the proposed Cryptosystem's Secs-level security, which is sufficient.

Similarly to ours, in another paper, we found where a group worked on predicting Leukaemia using their own Privacy Preserving Neural Network. In their paper, they used Fully Homomorphic Encryption (FHE) for encrypting the image datasets, made them into CIFAR-10 format and finally used some pre-used NN and their own CNN model to compare their result. They were able to reach a maximum of 80% accuracy rate using their model[24].

# Chapter 3

## Background Studies

### 3.1 CNN

Deep learning models have accomplished staggering results in computer image processing, so they are becoming a new trend in image recognition and classification. In this context comes our used CNN algorithm. CNN is divided into 4 stages. these are, 1. convolution, 2. nonlinear (ReLU), 3. pooling or downsampling, and 4. classification (fully connected layer). These activities are very important to build a CNN system. Big convolutional layers, that are derived from the convolution of pictures through several tiny kernels, are the foundation of CNN's structure. These many cores serve as distinctive identifiers to categorize various aspects of the incoming data, which is typically photos. However, applying those methods requires activation and pooling functions, which makes it difficult. Coupled neural networks will be used to connect them to the output layer after extracting features. A full representation of the CNN structure for the rock image is shown, including an input layer, feature maps, and a fully connected network.

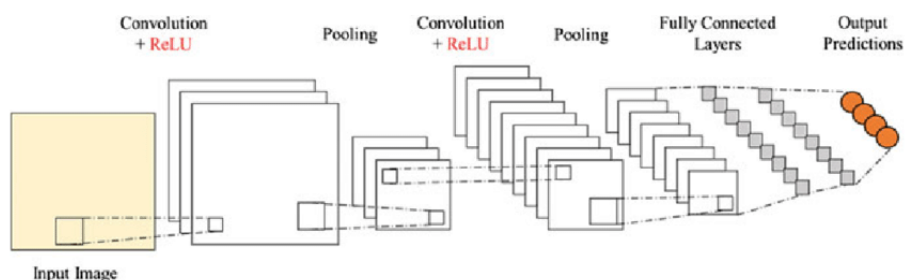


Figure 3.1: CNN Network

### 3.2 Homomorphic Encryption

Homomorphic encryption is an encryption technique that can convert a series of data into another form of data known as ciphertext [61]. Using this technique the values of all the data are changed completely so that if anyone other than the owner wants to access the original file, they are unable to do so without the keys used to encrypt the data. There are some major advantages of this encryption technique

that makes it stand out compared to other encryption methods. Based on what homomorphic technique is used, complex mathematical operations can be done on the data [11]. Basically, two types of operations can be done on ciphertext data. Using the relation between different parameters while encryption, multiplication, and addition operation can be on the cipher-text data infinite or finite amount of times depending on its type. Mainly homomorphic encryption can be divided into 3 subclasses. These are:

1. Partially Homomorphic Encryption (PHE)
2. Somewhat Homomorphic Encryption (SHE)
3. Fully Homomorphic Encryption (FHE)

Partially Homomorphic Encryption or PHE schemes have different types also. They are Unpadded RSA, ElGamal, Goldwasser-Micali, Benaloh and Paillier cryptosystems [10]. All the above-mentioned types can perform one operation at an infinite amount of times. For example, Unpadded RSA and ElGamal can encrypt the data in a computational way where multiplication on the cipher-text is possible. Whereas using Paillier cryptosystem, 5 parameters are created using dependencies, and working with that adding operation can be done in the cipher-text.

Somewhat Homomorphic Encryption or SHE encrypts the data in a way that it can make both addition and multiplication operations on the ciphertext but for a finite amount of time. The reason it can operate for a fixed amount of operation is that after every execution, the data grows. Once the data grows large enough, it starts to produce noise and the encrypted value starts to fade away [7]. As a consequence, at one time, the ciphertext can not be decrypted. Thus the goal of the encryption is lost. [8]

Fully homomorphic encryption of FHE is an upgraded version of SHE that can do addition and multiplication operations both and for an infinite amount of times [60]. The benefit of FHE is it does not drop the feature of any data to ensure total security. So, it may look like a dream come true but the worst case is its poor performance. The computation to create large numbers of keys is more complex than the PHE and as a result, it takes a heavy toll in the computation section while using a large number of datasets.

### **3.3 SVM**

The proposed Support vector machines model (SVM) in machine learning are supervised learning model featuring corresponding machine learning algorithms that evaluate datasets for classification and Identification. As these are the points nearest to the hyperplane, support vectors are a key component in determining the position of these hyperplanes. The machine learning approach is used to instruct an SVM (or support vector machine) in a linear method. We may estimate the error of our mode using K-fold cross-validation. As this will be utilized we can increase the size of our training dataset by adding the training and validation datasets [16]. Following feature extraction with the VGG16, the TensorFlow output is applied to the SVM model fitting. As a result, no separate extraction feature of pre-processed images is required. The model was applied about 100 times. Finally, we verify that the SVM

classifier has one hyper-parameter, which is the error term penalty of parameter C. [23]

### 3.4 VGG-16

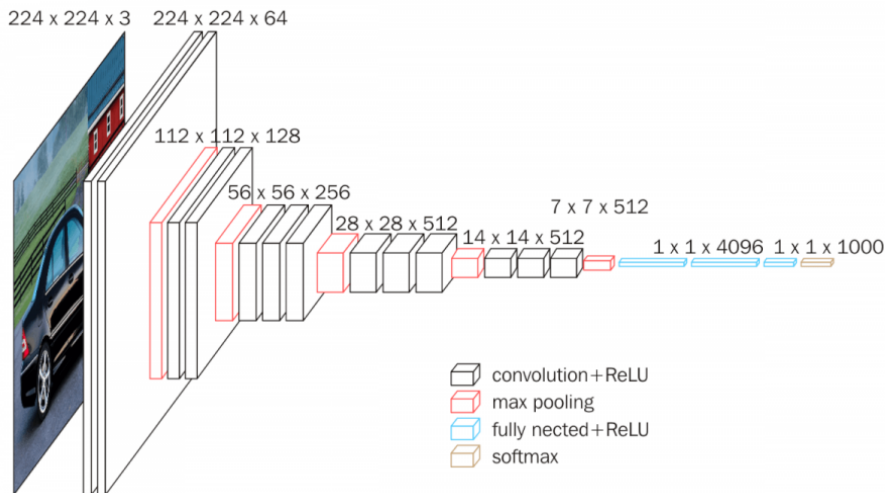


Figure 3.2: VGG-16 Architecture

The VGG model also referred to as VGGNet, is abbreviated as "VGG16." With its 16 layers and around 14 million training images, VGG16 can attain a test accuracy of 92.7 percent [18]. By deleting AlexNet's big filters and substituting a series of 3\*3 filters in their place, VGG16 enhanced AlexNet. Their affiliation was with the VGG at Oxford. It used a modest 3\*3 receptive field with a 1-pixel stride, which was the initial difference. VGG16 is a deep neural network with 16 layers, as its name suggests. Even by today's standards, VGG16 is a sizable network with 138 million total parameters. It is created using tiny convolutional filters. In this case, 13 convolutional layers and 3 completely connected layers. A 224x224 image has been uploaded to VGGNet. The model's creators for the ImageNet competition were able to keep a constant image input size by deleting a 224\*224 area of each image from the center. The shortest imaginable receptive field used by the VGG convolutional filters is 3\*3 [22]. An additional 1\*1 convolution filter is used in the linear transformation of the input for VGG. The main innovation of AlexNet for accelerating training is the ReLU Activation Function component. ReLU is a linear function that, for negative inputs, produces zero, and for positive inputs, produces the corresponding result. In all the hidden layers of the VGG network, ReLU is employed in place of Local Response Normalization, like in AlexNet. The latter requires more memory and extends training sessions, but overall accuracy isn't any better. Due to the rapid increase in the number of potential filters from 64 to 128, 256, and finally 512 in the last layers, pooling is crucial. VGGNet has three layers which are interconnected. The first two layers have exactly 4096 channels each and the third layer has 1000 channels with a channel set for each class. VGG's two main drawbacks are its protracted training period and its huge 500MB model size. The addition of skip connections and inceptions, which decrease the number of trainable parameters, improves the accuracy and training time of modern designs [18].



iterated 6 times. In addition, 9 more layers with  $1 \times 1,512$  cores,  $3 \times 3,512$  cores, and  $1 \times 1,2048$  cores iterated 3 times.[39] Finally, we have an average pooling layer with 1000 nodes.

Below is a diagram of Res Net50 architecture. The diagram shows a visual representation of different layers that are used to extract features from images without losing details.

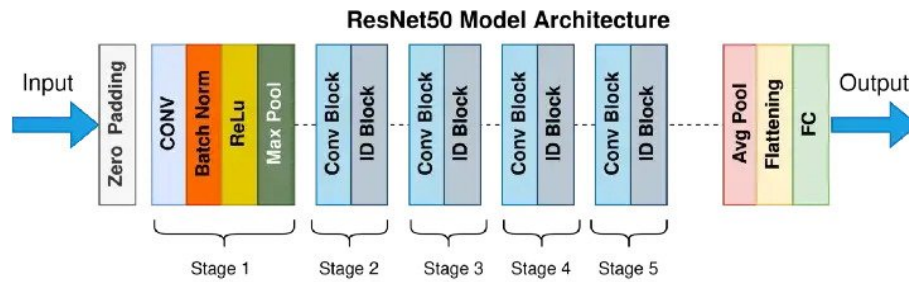


Figure 3.4: CNN Network



# Chapter 4

## Methodology

Our methodology section is divided into two sub sections. First part explains where we gathered the dataset from, why we used that particular dataset and dataset length etc. Then in the next part we described our model, the equation used, and the total encryption and decryption methods.

### 4.1 Dataset Description

For the dataset collection, we were looking for images which were structured well beforehand as we were trying to build an encryption method which can later be used on previous build R-CNN models. We are already aware of the fact that medical dataset were hard to collect due to confidentiality and patients rights which made our scouting for dataset even difficult. In our search, we were mainly looking for a well arranged dataset and so we had to compromise the image dataset size. For our case, a well arranged means all the images should be taken from the same perspective. We searched in kaggle because kaggle is the most widely used dataset platform. In kaggle we found a dataset with 253 tumour and non-tumour image files. Among the images, 98 of them were in the ‘no tumour’ category and the rest are in the ‘yes tumour’ category. We specially chose this dataset because all the images were of high quality and all images were taken from a top to bottom passion. Additionally, we didn’t want a dataset with more than two categories because as we were trying to build a new encryption scheme, the more the classes exist, the more difficult it will be to learn and classify encrypted images.

from ”<https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>”

Additionally, for training the NN models with more images, we increased our dataset size cautiously. For adding more images, we looked for images formatted precisely like the images we had earlier. Finally, after increasing the image files, the length changed from 253 to 353 in terms of total image number. In our latest findings we made our total number of ‘no tumour’ to 198 and the size of ‘yes tumour’ was the same as before which was 155.

## 4.2 Our Homomorphic Encryption Model

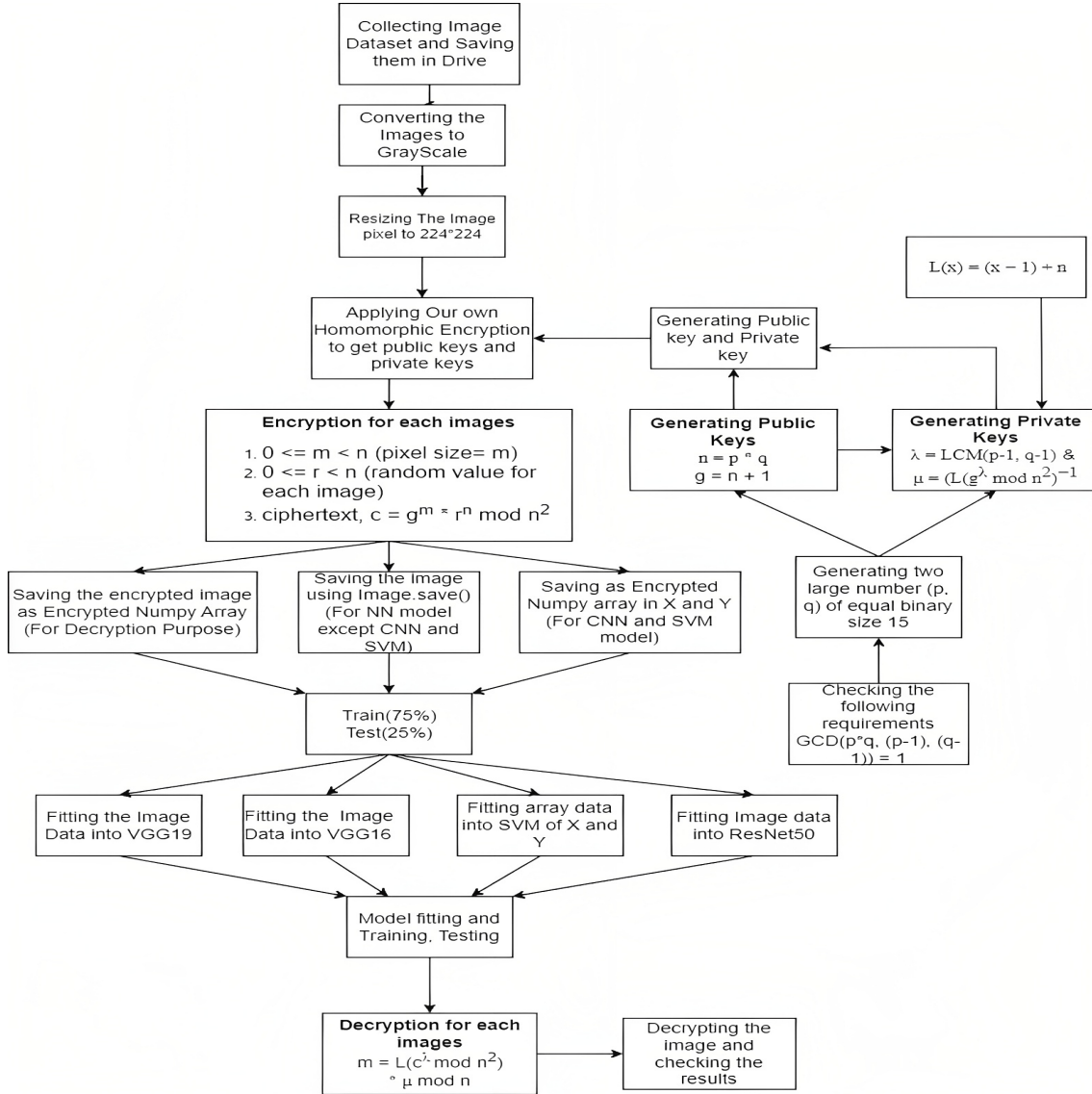


Figure 4.1: Total Workflow

We have provided the whole process in the workflow above.

As our main goal is to encrypt image data using Homomorphic encryption we tried to use the most efficient and time-saving way to encrypt our data. Among the 3 types of homomorphic encryption, we used Partial Homomorphic Encryption or PHE. Generating the keys in PHE is much more simple and the computation pressure reduces a lot in this algorithm.

So to encrypt our image, we had to create multiple methods. First, we needed to generate private and public keys. Private keys are used to decrypt the image to the original image. Public keys along with the encrypted data can be provided to the 3rd parties because without the private keys the encrypted image data cannot be decrypted. And then the encrypted image can go through different NN for prediction purposes and we can get a result based on the encrypted images.

· 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.

$\text{GCD}(p \times q, (p-1) \times (q-1)) = 1$ , where GCD means Greatest Common Divisor. It primarily ensures that two generated numbers are prime numbers.

After that, we need to calculate  $n$ , where  $n = (p \times q)$  and  $\lambda = \text{LCM}(p-1, q-1)$  where LCM finds out the Least Common Multiplier between two variables. This is done in our private class part which calculates the public and private key. Similarly, we calculated  $g$ , where  $g = n + 1$  as two primes are of equal 15 in binary numbers. These parameters are calculated and importantly related to each other for decryption purposes.

Next, we calculated the modular multiplicative inverse.  $\mu = (L(g^\lambda \bmod n^2))^{-1}$ .

Here,  $L(x) = (x-1) \div n$ . At last, we got the public key  $(n, g)$  and private key  $(\lambda, \mu)$ .

## Encryption

- To encrypt the images, the pixel size 'm' must be of  $0 \leq m < n$
- Then we calculate a random value  $r$  where  $0 \leq r < n$
- Finally we compute the cipher-text as:  $c = (g^m \times r^n) \bmod n^2$

```
def Encrypt(public_key, plaintext, r):
    """
    It encrypts the plaintext using the given public key.
    """

    #r = random.randint( 1, public_key.n-1)
    #while not xgcd( r, public_key.n)[0] == 1:
    #    r = random.randint( 1, public_key.n)

    a = pow(public_key.g, plaintext, public_key.nsq)
    b = pow(r, public_key.n, public_key.nsq)

    ciphertext = (a * b) % public_key.nsq
    return ciphertext
```

Figure 4.2: Encryption Method

### 4.2.1 Decryption

1. To decrypt, a value of pixel  $c$  is given.
2. To decrypt it, we need to use the following equation  $m = L(c^\lambda \bmod n^2) \times \mu \bmod n$

```

def Decrypt(public_key, private_key, ciphertext):
    """
    It decrypts the ciphertext using the given public key and private key.
    """

    x = pow(ciphertext, private_key.λ, public_key.nsq)
    l = lambda x: (x - 1) // public_key.n

    plaintext = (l(x) * private_key.μ) % public_key.n
    return plaintext

```

Figure 4.3: Decryption Method

## 4.2.2 Encrypting and Decrypting the Images

In these two methods, we have taken the whole image as a matrix. Then we change our matrix to a NumPy matrix to use the `tolist()` 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 in the encryption method mentioned before. But in doing so, the encrypted image pixels are so randomized that the feature 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.

To get the better of this pixel-related mess, we have encrypted every image with a random number each so that we still can get a list of encrypted images but with higher accuracy using the old NN models. The benefits of using a single random number each is that the random generated each time is closer and when encrypting the image, the feature of an encrypted image of a brain with a tumor looks similar and which does not is also different than the brain which does not have a brain. 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  $224 \times 224$ , meaning every image will go through the encryption method to encrypt the image.

```

def image_encryption(public_key, plain_image):
    """
    This function encrypt the given image using Paillier Cryptosystem.
    """

    r = random.randint( 1, public_key.n-1)
    while not xgcd( r, public_key.n)[0] == 1:
        r = random.randint( 1, public_key.n)
    cipher_image = np.asarray(plain_image)
    #print(cipher_image)
    #print(cipher_image.dtype)
    shape = cipher_image.shape
    cipher_image = cipher_image.flatten().tolist()
    for i in range(len(cipher_image)):
        cipher_image[i] = Encrypt(public_key, cipher_image[i], r)

    return np.asarray(cipher_image).reshape(shape)

```

Figure 4.4: Encrypting a whole image

```
def Encrypt(public_key, plaintext, r):
    """
    It encrypts the plaintext using the given public key.
    """

    #r = random.randint( 1, public_key.n-1)
    #while not xgcd( r, public_key.n)[0] == 1:
        #r = random.randint( 1, public_key.n)

    a = pow(public_key.g, plaintext, public_key.nsq)
    b = pow(r, public_key.n, public_key.nsq)

    ciphertext = (a * b) % public_key.nsq
    return ciphertext
```

Figure 4.5: Decrypting a whole image

### 4.2.3 Encrypting the whole Image Data-sets

At first, we made a directory called Encrypted Brain in the Google Colab content directory. Then We created one folder inside the Encrypted Brain named Images and another in Numpy. In both folders there were two folders, one was for the images and NumPy format each which had a tumor and named it 'Yes Tumor', and another folder which did not, and we saved that folder name as 'No Tumor'. Also, we created two folders in the content section for dividing our encrypted images into Train and Test 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 image we have set the value 1 and for the no image, we set the value 0.

Before reading the image, we set the Image size to 224 as we do not need a higher pixel size which will increase the encryption time but won't do much better in our detection part.

Then we declared two arrays, 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 Images to read images in RGB and images to convert the RGB image to Gray. After converting the images to grayscale, we resize our image to 224. Then 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.

```
import glob
IMAGE_SIZE = 224

X = []
Y = []

for cls in classes:
    path = '/content/drive/MyDrive/brain_tumor_dataset/' + cls
    c = 0
    for i in os.listdir(path):
        c += 1
        img = Image.open(path + '/' + i)
        #img = img.convert('RGB')
        img = ImageOps.grayscale(img)
        img = img.resize((IMAGE_SIZE, IMAGE_SIZE))
        #Saving as Numpy Array
        numpy_encrypted_image = image_encryption(publickey, img)
        if (cls == 'yes'):
            name = 'YES_' + str(c)

            #Saving as Numpy Array
            numpy_enc_yes_path = (f'/content/Encrypted brain/NumpyArray/Yes tumor/' + name + '.npy')
            np.save(numpy_enc_yes_path, numpy_encrypted_image)
            X.append(numpy_encrypted_image)
            #print(classes[cls])
            Y.append(classes[cls])

            #Saving as JPG image
            encrypted_image = Image.fromarray(numpy_encrypted_image.astype(np.uint8))
            img_enc_yes_path = (f'/content/Encrypted brain/Image/Yes tumor/' + name + '.jpg')
            encrypted_image.save(img_enc_yes_path)
        elif (cls == 'no'):
            name = 'NO_' + str(c)

            #Saving as Numpy Array
            numpy_enc_no_path = (f'/content/Encrypted brain/NumpyArray/No tumor/' + name + '.npy')
            np.save(numpy_enc_no_path, numpy_encrypted_image)
            X.append(numpy_encrypted_image)
            Y.append(classes[cls])

            #Saving as JPG image
            encrypted_image = Image.fromarray(numpy_encrypted_image.astype(np.uint8))
            img_enc_no_path = (f'/content/Encrypted brain/Image/No tumor/' + name + '.jpg')
            encrypted_image.save(img_enc_no_path)
    print('We have ' + str(c) + ' ' + cls + ' Images')
```

Figure 4.6: Reading images, Preprocessing and Encrypting the data-sets

# Chapter 5

## Experimentation and Result Analysis

In this section, we will be comparing our Homomorphic method with the conventional Paillier cryptosystem and FHE and briefly discuss how our encryption model retains its feature for NN models to use and finally show how much difference it creates while training and detecting brain tumors.

Normally FHE performs inefficiently despite having strong functionality and security[32]. 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.

Using the conventional Partial Homomorphic Encryption which is based on Paillier, a random number is generated for each pixel. As a result, a mess of pixels is created with no sense of feature extraction.

In compared to Paillier Crypto, using our PHE model, specific images represent the structure of the brain image. The reason is when we generated the random number ( $r$ ) for each images separately, after all the encryption computation, an array is produced for each images. In image format, tumour looks like a white circle. So, in the encrypted image array, the array value for the images in tumour part is slightly higher. After making the array into .jpg format the relation remains same although the image look encrypted at the same time. As a result of providing protection and retaining a huge percentage of its value, it enables us to use the existing Model to identify the features and generate a decent prediction score on the encrypted image.

Also, In the Paillier Cryptosystem model, the encryption time takes longer due to the same reason of the random number being generated for each pixel, then for an image with  $224 \times 224$  pixels, it takes  $\Omega(n^2)$  times to encrypt an image file.

We have even made our encryption scheme faster than 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.

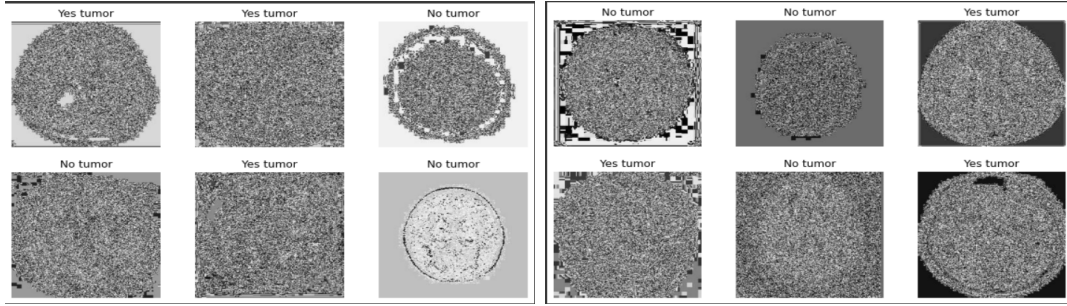


Figure 5.1: Our Encryption Method

## 5.1 Comparison between Paillier Cryptosystem and Our FHE in terms of accuracy

We have used a number of Neural Network models to check the accuracy of our model.

### 5.1.1 Accuracy based on Paillier Cryptosystem

In our experimentation part, we tried the conventional PHE and checked if we can get a better result using previously created Neural Network models like CNN, VGG16, VGG19, ResNet50, and SVM model. In our experiment, using the existing models, we checked with the raw images with no encryption at first. Here, All the results were quite impressive as typically using VGG16, VGG19, and ResNet50 models we got a validation accuracy of above 95 percent, the CNN model giving around 80 to 85 percent and the SVM model result was around 75 percent.

So, we were ensured that this model was good enough to compare our Homomorphic models and the Paillier Cryptosystem model.

Using the Paillier cryptosystem model in all the mentioned models, we have got an accuracy ranging from 38 to 60 percent. The reason behind getting such less accuracy rate is that this encryption technique randomizes each pixel value making it exceedingly difficult to read, as a consequence, the Neural Network models are unable to get the underlying value of its feature and weights which results in randomly detecting images.



Images of the accuracy rates are shown below:

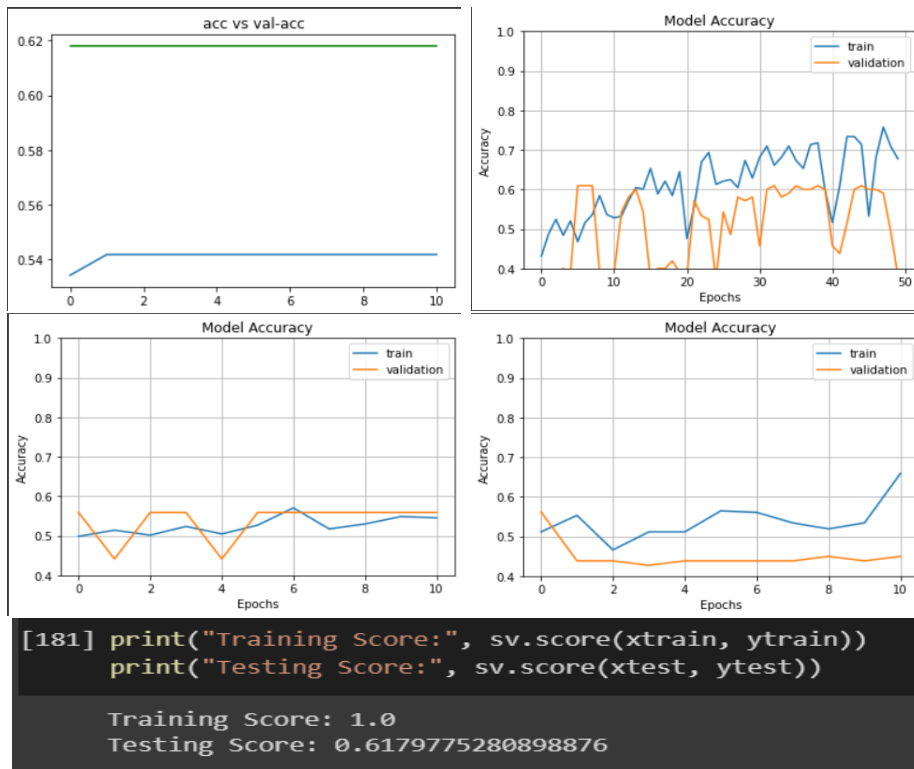


Figure 5.2: Accuracy of Each Models (CNN,ResNet50,VGG16,VGG19,SVM)

### 5.1.2 Accuracy based on our Homomorphic Technique

The training and validation accuracy graph using our custom encryption technique is given below. In a typical CNN model, we do not get a better accuracy as CNN does not have a learning feedback system. Here in the graph, the accuracy and validation accuracy comparison is given below:

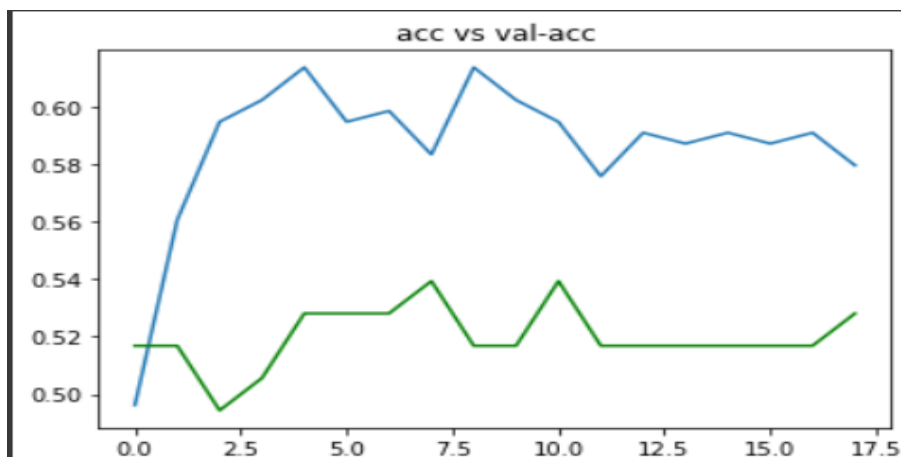


Figure 5.3: Training and validation accuracy graph for CNN

On the other hand, other models except for CNN gives a better result in fewer epochs.

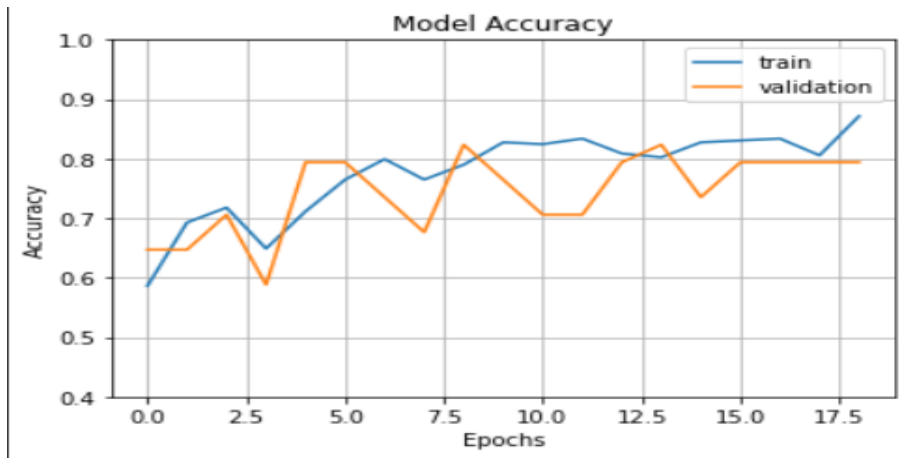


Figure 5.4: VGG16 Training accuracy vs Validation accuracy graph

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.

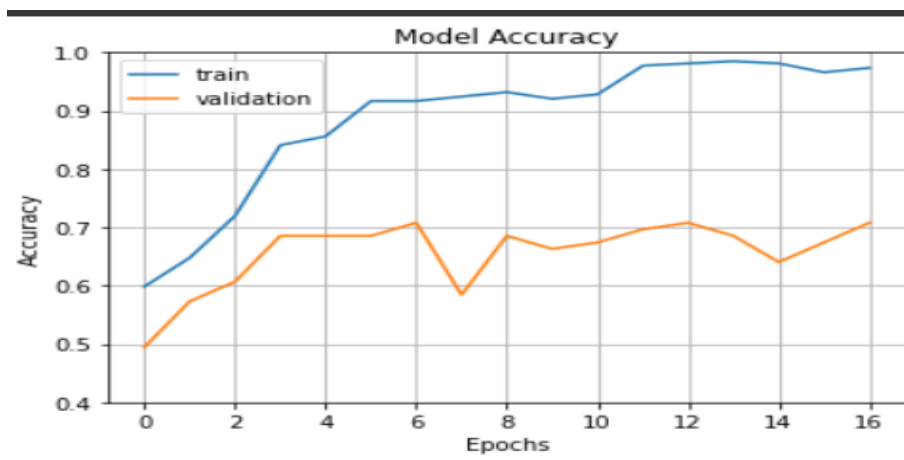


Figure 5.5: VGG19 Training accuracy vs Validation accuracy graph

Here also, in the above-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.

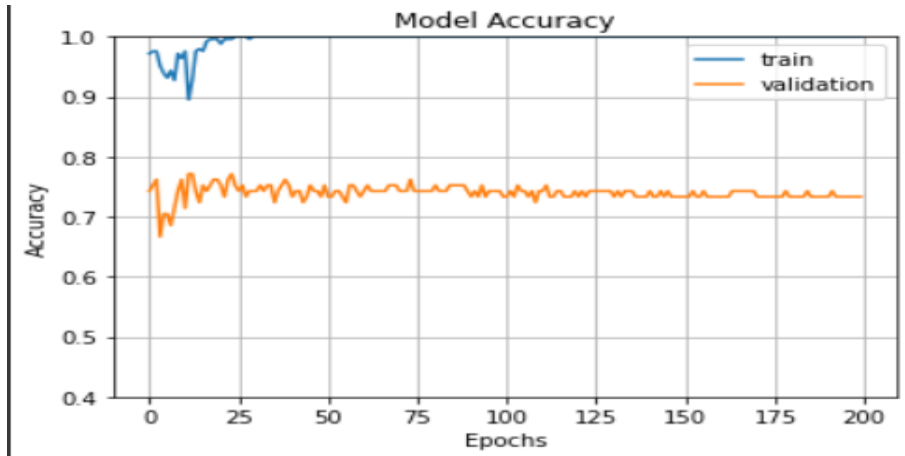


Figure 5.6: ResNet50 Training accuracy vs Validation accuracy graph

Only in the ResNet50 model, we have trained our data for 200 epochs to check the results. From the above graph, we can see that the ResNet50 model gave us an accuracy of 100 percent close to 25 epochs and the validation accuracy was around 75 percent throughout the validation period.

```
[ ] print("Training Score:", sv.score(xtrain, ytrain))
    print("Testing Score:", sv.score(xtest, ytest))

Training Score: 0.9924242424242424
Testing Score: 0.6067415730337079
```

Figure 5.7: SVM training and testing score

In the above picture and the one from using Paillier Cryptosystem, we can see the training accuracy is around 100 percent and the testing score is around 66 percent. To get this much higher value from a Machine Learning algorithm is unexpected. We determined that because of the over-fitting issue, this result was provided. So, we entirely avoided the result in our consideration.

The table below shows the training accuracy and validation accuracy between our encrypted model and Paillier-crypto model.

Training Accuracy	VGG16	ResNet-50	VGG19	CNN	SVM
Paillier-crypto	58%	78%	68%	62%	100%
Our PH-Encryption	89%	100%	98%	70%	100%

Table 5.1: Training Accuracy

Validation Accuracy	VGG16	ResNet-50	VGG19	CNN	SVM
Paillier-crypto	57%	60%	54%	54%	61.79%
Our PH-Encryption	82%	75%	71%	55%	76%

Table 5.2: Validation Accuracy

From the above table VGG16 has the highest validation accuracy. VGG16 is a Neural Network model so the result is satisfactory. Though we expected VGG19 and ResNet50 to have better validation accuracy, as they have more weighted layers. The result we got are not too far from VGG16. Though we have not implemented AlexNet model to our experiment, we plan to look and improve upon it in near future.

In our experiment we have trained our the neural network and machine learning model for 100 epochs and in the callbacks module if the value accuracy does not change much, we save the more accurate model as our best model. Only in ResNet50. We have trained the model for 200 epochs.

# Conclusion

We have used different types of values, parameters, and functions in our CNN and Homomorphic encryption model using the same database. We used various techniques and types of encryption algorithms to come up with our own encryption algorithm. Using our algorithm we trained Neural Network models and Machine Learning models. Our Encryption algorithm was able to keep the patient data secure through a version of Partial Homomorphic encryption and was able to be efficient in encrypting the features without losing valuable data. As a result of our algorithm, we managed to train the model even faster with better training and validation accuracy, thus saving valuable time. We used a total of 353 images out of which were 198 images of a normal brain and 155 images of a brain with a different tumor in it. The validation accuracy we got for after training the models we got validation accuracy of VGG16(82%), VGG19(71%), ResNet50(75%), CNN(55%), and SVM(76%). With more data we expect our model to outperform the current accuracy. The better accuracy we get the more we will be able to implement our algorithm in an everyday scenario. Moreover, our algorithm can be used in other medical fields and in other industries too.

Our primary aim is to construct a system that will be able to detect brain tumors but the patient data will be in encrypted form. Only the user can access the data and information using a unique private key. This will help to keep patient information secure and confidential. As a result, people will trust the healthcare system, which can improve the quality of health services.

# Bibliography

- [1] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980.
- [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [3] A. Capelle, O. Alata, C. Fernandez, S. Lefevre, and J. Ferrie, “Unsupervised segmentation for automatic detection of brain tumors in mri,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 1, 2000, 613–616 vol.1. DOI: 10.1109/ICIP.2000.901033.
- [4] L. M. DeAngelis, “Brain tumors,” *New England journal of medicine*, vol. 344, no. 2, pp. 114–123, 2001.
- [5] T. Kesavamurthy and S. SubhaRan, “Pattern classification using imaging techniques for infarct and hemorrhage identification in the human brain,” vol. 4, Nov. 2005.
- [6] J. G. Conde, S. De, R. W. Hall, E. Johansen, D. Meglan, and G. C. Peng, “Telehealth innovations in health education and training,” *Telemedicine and e-Health*, vol. 16, no. 1, pp. 103–106, 2010.
- [7] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, “Multiparty computation from somewhat homomorphic encryption,” in *Annual Cryptology Conference*, Springer, 2012, pp. 643–662.
- [8] J. Fan and F. Vercauteren, *Somewhat practical fully homomorphic encryption*, Cryptology ePrint Archive, Paper 2012/144, <https://eprint.iacr.org/2012/144>, 2012. [Online]. Available: <https://eprint.iacr.org/2012/144>.
- [9] S. Goswami and L. K. P. Bhaiya, “Brain tumour detection using unsupervised learning based neural network,” in *2013 International Conference on Communication Systems and Network Technologies*, 2013, pp. 573–577. DOI: 10.1109/CSNT.2013.123.
- [10] L. Morris, “Analysis of partially and fully homomorphic encryption,” *Rochester Institute of Technology*, pp. 1–5, 2013.
- [11] X. Yi, R. Paulet, and E. Bertino, “Homomorphic encryption,” in *Homomorphic encryption and applications*, Springer, 2014, pp. 27–46.
- [12] H.-C. Shin, H. R. Roth, M. Gao, *et al.*, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

- [13] X. Yi, A. Bouguettaya, D. Georgakopoulos, A. Song, and J. Willemson, "Privacy Protection for Wireless Medical Sensor Data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 369–380, May 2016.
- [14] W. Cao, Y. Zhou, C. P. Chen, and L. Xia, "Medical image encryption using edge maps," *Signal Processing*, vol. 132, pp. 96–109, 2017. DOI: 10.1016/j.sigpro.2016.10.003.
- [15] X. Chen and S. Zou, "Improved wi-fi indoor positioning based on particle swarm optimization," *IEEE Sensors Journal*, vol. 17, no. 21, pp. 7143–7148, 2017. DOI: 10.1109/jssen.2017.2749762.
- [16] P. Liu, K.-K. R. Choo, L. Wang, and F. Huang, "Svm or deep learning? a comparative study on remote sensing image classification," *Soft Computing*, vol. 21, no. 23, pp. 7053–7065, 2017.
- [17] M. Soltaninejad, G. Yang, T. Lambrou, *et al.*, "Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in flair mri," *International journal of computer assisted radiology and surgery*, vol. 12, no. 2, pp. 183–203, 2017.
- [18] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, IEEE, 2018, pp. 169–175.
- [19] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, *Convolutional neural networks: An overview and application in radiology - insights into imaging*, Jun. 2018. [Online]. Available: <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9#citeas>.
- [20] 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, Jun. 2019. DOI: 10.31142/ijtsrd23780.
- [21] L. Jiang, L. Chen, T. Giannetsos, B. Luo, K. Liang, and J. Han, "Toward Practical Privacy-Preserving Processing Over Encrypted Data in IoT: An Assistive Healthcare Use Case," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10177–10190, Dec. 2019.
- [22] T. Kaur and T. K. Gandhi, "Automated brain image classification based on vgg-16 and transfer learning," in *2019 International Conference on Information Technology (ICIT)*, IEEE, 2019, pp. 94–98.
- [23] I. Q. Khilji, K. Saha, J. A. Shonon, and R. Israq, *Prediction of acute lymphoid leukemia using Privacy Preserving Neural Network*, <http://hdl.handle.net/10361/15076>, [Online; accessed 2023-01-17], Jan. 2019.
- [24] 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>.
- [25] Z. U. Rehman, S. S. Naqvi, T. M. Khan, M. A. Khan, and T. Bashir, "Fully automated multi-parametric brain tumour segmentation using superpixel based classification," *Expert systems with applications*, vol. 118, pp. 598–613, 2019.

- [26] K. Salçin *et al.*, “Detection and classification of brain tumours from mri images using faster r-cnn,” *Tehnički glasnik*, vol. 13, no. 4, pp. 337–342, 2019.
- [27] S. Haddad, G. Coatrieux, A. Moreau-Gaudry, and M. Cozic, “Joint Watermarking-Encryption-JPEG-LS for Medical Image Reliability Control in Encrypted and Compressed Domains,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2556–2569, 2020.
- [28] H. Qiu, M. Qiu, M. Liu, and G. Memmi, “Secure Health Data Sharing for Medical Cyber-Physical Systems for the Healthcare 4.0,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2499–2505, Sep. 2020.
- [29] *Understanding the VGG19 Architecture*, <https://iq.opengenus.org/vgg19-architecture/>, [Online; accessed 2023-01-17], Feb. 2020.
- [30] S. Dhawan and R. Gupta, “Analysis of various data security techniques of steganography: A survey,” *Information Security Journal: A Global Perspective*, vol. 30, no. 2, pp. 63–87, 2021.
- [31] *Patient Privacy Data Protection: Different laws around the world*, <https://www.brainlab.com/j/privacy-data-protection-different-laws-around-the-world/>, [Online; accessed 2023-01-22], Jan. 2021.
- [32] *What is fully homomorphic encryption?* Apr. 2021. [Online]. Available: <https://inpher.io/technology/what-is-fully-homomorphic-encryption/>.
- [33] *What is homomorphic encryption, and why isn't it mainstream?* Jul. 2021. [Online]. Available: <https://www.keyfactor.com/blog/what-is-homomorphic-encryption/#:~:text=Partially%5C%20homomorphic%5C%20encryption%5C%20algorithms%5C%20allow,sum%5C%20of%5C%20the%5C%20two%5C%20plaintexts..>
- [34] M. F. Alanazi, M. U. Ali, S. J. Hussain, *et al.*, *Brain tumor/mass classification framework using magnetic-resonance-imaging-based isolated and developed transfer deep-learning model*, Jan. 2022. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8749789/>.
- [35] M. Arif, F. Ajesh, S. Shamsudheen, O. Geman, D. Izdrui, and D. Vicoveanu, “Brain tumor detection and classification by mri using biologically inspired orthogonal wavelet transform and deep learning techniques,” *Journal of Healthcare Engineering*, vol. 2022, 2022.
- [36] Y. Bao, W. Qiu, P. Tang, and X. Cheng, “Efficient, Revocable, and Privacy-Preserving Fine-Grained Data Sharing With Keyword Search for the Cloud-Assisted Medical IoT System,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2041–2051, May 2022.
- [37] X. Liu, X. Yang, Y. Luo, and Q. Zhang, “Verifiable Multikeyword Search Encryption Scheme With Anonymous Key Generation for Medical Internet of Things,” *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 315–22 326, Nov. 2022.



- [38] L. Mouselimis, *Image segmentation based on superpixels and clustering*, Dec. 2022. [Online]. Available: [https://cran.r-project.org/web/packages/OpenImageR/vignettes/Image\\_segmentation\\_superpixels\\_clustering.html#:~:text=%5C%E2%5C%80%5C%9CIn%5C%20computer%5C%20vision%5C%2C%5C%20image%5C%20segmentation,meaningful%5C%20and%20easier%5C%20to%5C%20analyze..](https://cran.r-project.org/web/packages/OpenImageR/vignettes/Image_segmentation_superpixels_clustering.html#:~:text=%5C%E2%5C%80%5C%9CIn%5C%20computer%5C%20vision%5C%2C%5C%20image%5C%20segmentation,meaningful%5C%20and%20easier%5C%20to%5C%20analyze..)
- [39] *Resnet-50: The Basics and a Quick Tutorial*, <https://datagen.tech/guides/computer-vision/resnet-50/>, [Online; accessed 2023-01-17].
- [40] W. Stallings, *Http://williamstallings.com/Crypto3e.html*, <http://williamstallings.com/Crypto3e.html>, [Online; accessed 2023-01-22].
- [41] *The neural networks model*. [Online]. Available: <https://www.ibm.com/docs/en/spss-modeler/18.0.0?topic=networks-neural-model>.
- [42] Venafi, *What is homomorphic encryption amp; how is it used*. [Online]. Available: <https://venafi.com/blog/homomorphic-encryption-what-it-and-how-it-used/>.
- [43] *What is unsupervised learning?* [Online]. Available: <https://www.ibm.com/topics/unsupervised-learning>.