







# IIMFCBM: Intelligent Integrated Model for Feature Extraction and Classification of Brain Tumors Using MRI Clinical Imaging Data in IoT-Healthcare

Amin Ul Haq , Jian Ping Li , Bless Lord Y Agbley, Asif Khan , Inayat Khan ,  
M. Irfan Uddin , and Shakir Khan 

**Abstract**—Accurate classification of brain tumors is vital for detecting brain cancer in the Medical Internet of Things. Detecting brain cancer at its early stages is a tremendous medical problem, and many researchers have proposed various diagnostic systems; however, these systems still do not effectively detect brain cancer. To address this issue, we proposed an automatic diagnosing framework that will assist medical experts in diagnosing brain cancer and ensuring proper treatment. In developing the proposed integrated framework, we first integrated a Convolutional Neural Networks model to extract deep features from Magnetic resonance imaging. The extracted features are forwarded to a Long Short Term Memory model, which performs the final classification. Augmentation techniques were applied to increase the data size, thereby boosting the performance of our model. We used the hold-out Cross-validation technique for training and validating our method. In addition, we used various metrics to evaluate the proposed model. The results obtained from the experiments show that our model

achieved higher performance than previous models. The proposed model is strongly recommended to be used to diagnose brain cancer in Medical Internet of Things health-care systems due to its higher predictive outcomes.

**Index Terms**—Algorithms, brain tumors, classification, clinical MRI data, convolution neural network, data augmentation, feature extraction, integration, long short term memory, Medical Internet of Things.

## I. INTRODUCTION

**B**RAIN cancer (BC) is a critical medical issue globally, resulting in many deaths. Among the different kinds of cancer, brain tumors are dangerous due to their critical nature. There are various forms of tumors of the brain, such as Meningioma, Glioma, Pituitary, and Acoustic Neuroma [1]. In clinical observation, the rates of Meningioma, Glioma, and Pituitary tumors are 15%, 45%, and 15%, respectively [2]. From the type of tumor, clinical experts can decide about the treatment and recovery process [3]. For the past several years, the Internet of Things (IoT) has witnessed a transformation that provides a mechanism to evaluate real-time and historical data using artificial intelligence (AI) and data mining (DM) methods. Different non-invasive methods for brain cancer diagnosis-related problems have been proposed for use in MIoT [4], [5]. These techniques primarily rely on artificial intelligence (AI), such as machine learning (ML) and deep learning (DL) approaches to classify brain tumors. CNN models have become more prevalent in recent years for medical image data analysis owing to the ability of CNN models to appropriately and automatically extract features from the given data eliminating manual feature engineering [6]–[9].

In the literature, various researchers proposed different techniques for BC detection employing ML and DL algorithms. Zacharaki *et al.* [10] designed a brain cancer diagnosis system to classify various grades of Glioma employing Support Vector Machine (SVM) and k-Nearest Neighbor (KNN) machine learning models and respectively achieved 85% and 88% classification accuracies. Cheng *et al.* [11] presented a method for the classification of brain tumors. They augmented the tumor region, thereby improving the classification performance. They employed three methods for extracting the features. These include bag of words, gray level co-occurrence matrix, and intensity

Manuscript received 5 January 2022; revised 4 March 2022 and 5 April 2022; accepted 26 April 2022. Date of publication 3 May 2022; date of current version 5 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61370073, in part by the National High Technology Research and Development Program of China, in part by the Project of Science and Technology Department of Sichuan Province under Grant 2021YFG0322, in part by the Project of Science and Technology Department of Chongqing Municipality, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K202114401, in part by Chongqing Qinchengxing Technology Company, Ltd., in part by Chengdu Haitian Digital Technology Company, Ltd., in part by Chengdu Chengdian Network Technology Company, Ltd., in part by Chengdu Civil-military Integration Project Management Company, Ltd., and in part by Sichuan Yin Ten Gu Technology Company, Ltd. (Corresponding authors: Amin Ul Haq; Jian Ping Li.)

Amin Ul Haq, Jian Ping Li, Bless Lord Y Agbley, and Asif Khan are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China (e-mail: khan.amin50@yahoo.com; jpli2222@uestc.edu.cn; agbleybless@outlook.com; asif05amu@yahoo.com).

Inayat Khan is with the Department of Computer Science, University of Buner, Buner, Khyber Pakhtunkhwa 19290, Pakistan (e-mail: inayat\_khan@uop.edu.pk).

M. Irfan Uddin is with the Institute of Computing, Kohat University of Science and Technology, Kohat, Khyber Pakhtunkhwa 26000, Pakistan (e-mail: irfanuddin@kust.edu.pk).

Shakir Khan is with the College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia (e-mail: sgkhan@imamu.edu.sa).

Digital Object Identifier 10.1109/JBHI.2022.3171663

histogram. Their proposed diagnostic technique resulted in a 91.28% accuracy.

Dahshan *et al.* [12] proposed a brain tumors classification method with 80 Brain Magnetic Resonance Imaging (MRI) images. They have used discrete wavelet transform and Principle Components Analysis (PCA) algorithms for reducing the dimensions of the data. To classify the tumors, they applied Artificial Neural Network (ANN) and KNN classifiers. The classifiers ANN and KNN obtained 97% and 98% classification accuracies. Afshar *et al.* [13] presented a brain tumor classification method employing a capsule network that combined MRI images of the brain and the coarse tumor boundaries. Their proposed method obtained 90.89% accuracy. Anaraki *et al.* [14] proposed an integrated framework for brain tumor classification, consisting of Convolutional Neural Network (CNN) and Genetic Algorithm (GA). Their GA-CNN framework obtained 94.2% accuracy. Ani *et al.* [4] designed an IoT-based system for patient monitoring and diagnosis. They used the ensemble learning approach in the proposed method, and the system achieved an accuracy result of 93 percent.

According to the literature reviewed, the existing brain cancer diagnosis methods do not result in high BC diagnosis accuracy. A new IoT-based automatic diagnostic system (ADS) is required to detect and administer treatment for brain cancer patients in a timely fashion [15].

This paper proposes an AI-based intelligent integrated framework (CNN-LSTM) for brain tumors classification and diagnosis in the IoT healthcare industry. In the integrated framework design, we have incorporated the CNN model to extract features from medical MRI data automatically. The extracted features are passed to our Long short-term memory (LSTM) model to learn the dependencies in the features and finally predict the class for the tumor. We applied brain MRI data sets for the assessment of the proposed integrated model. Massive data is one requirement for an effective deep learning model like ours. Since the size of our original data set is small, we utilized data augmentation approaches to increase the data set size, thereby improving the model result during training. We also used the train-test splits Cross-validation approach for hyperparameter tuning and best model selection to ensure proper model fitting. For model assessment, we used well-known evaluation measures. We compared the predictive outputs of our proposed CNN-LSTM model with previous methods in the Medical Internet of Things (MIoT) healthcare industry.

This study's innovations are summarised as follows:

- IoT-based Brain tumor classification and detection system with an AI model.
- CNN and LSTM based intelligent integrated framework (CNN-LSTM) was proposed to detect brain cancer using brain tumor image data in IoT healthcare environment. The CNN module automatically extracts relevant features from the image data, while the LSTM module analyses and classifies the tumor.
- To enhance the predictive capability of the CNN-LSTM model, we used augmentation approaches to transform the training examples, increasing the data sets' sizes for a more effective model training.

- The CNN-LSTM model outputs were measured using different model assessment measures and compared with previous models. Due to the excellent outcomes of our model, we recommend it for the classification of brain tumors in MIoT healthcare systems.

The following sections present the remaining parts of the paper: Section 2 is related to materials and methodology. We report our experiments in section 3. Section 4 is the conclusion.

## II. MATERIALS AND METHOD

In this section, we present the data sets used for training our model and demonstrate the proposed model's methodology in detail.

### A. Data

We have used three data sets for the evaluation of our model. The first one is the Brain Tumor Data Set (BTDS) provided by Tianjing Medical University, China, in different versions. A new version was published in 2017 [11]. This data set contains contrast-enhanced T1-weighted images of 233 subjects suffering from 3 kinds of brain tumors, namely Pituitary, Glioma as well as Meningioma [16]. The second data set was acquired from the cancer imaging archive repository [17]. The Molecular Brain Neoplasia Data Set (MBNDS) included multi-sequence MRI images of 130 subjects [18]. The contrast-enhanced T1-weighted images consist of various grades of glioma (2, 3, 4). The third data set used is the Brain MRI Images Data Set (BMIDS), which includes 253 MRI brain images. The tumor class in the data set is composed of 155 images, while the non-tumor class comprises 98 images.

### B. Method

The proposed model's methodology is presented, and the theoretical and mathematical foundations of Convolutional Neural Network and Long Short Term Memory models are thoroughly discussed in this section.

1) *Convolutional Neural Network*: CNN's are advanced feed-forward neural networks [19], [20] that utilize the unique convolution property. The convolutions are able to scan for translation invariance, as the filters are not dependent on positions. The number of parameters greatly reduced after the convolution operation. There are usually three different layers present in a typical CNN model. They include the convolutional layer, which applies kernels to reduce the parameter sizes, a pooling layer to capture a representation of selected portions of the data and a fully connected layer. Also, different functions accomplished by these layers include reducing the dimensions, extracting appropriate features, and classifying the resulting features. Filters are used to convolve the input data in the convolution step, computing the corresponding activation map. The sliding operation is linear, hence the use of dot product to speed up the deployment. Considering  $x$  and  $w$  as an input vector and weight function respectively, we can express the convolution  $(x * w)(a)$  on each time  $t$  formally in all dimensions as shown by (1).

$$(x * w)(a) = \int x(t)w(a - t)da \quad (1)$$

Where  $a$  is in  $\mathbf{R}^n$  for a given  $n \geq 1$  and  $t$  is a discrete parameter. Here, we express the discrete convolution as shown by (2):

$$(x * w)(a) = \sum_a x(t)w(t-a) \quad (2)$$

Two or three-dimensional convolutions are usually employed in CNN models. Hence in the scenario of 2-D input image  $I$ , and given that  $K$  represents a 2-D kernel, we can express the convolution process mathematically as shown by the following equation:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n) \quad (3)$$

In the case of 3-dimensional image data, the convolution process can be written mathematically as in (4):

$$(I * K)(i, j, k) = \sum_m \sum_n \sum_l I(m, n, l)K(i-m, j-n, k-l) \quad (4)$$

In order to introduce non-linearity in the model, activation functions, Sigmoid and ReLU can be incorporated. The sigmoid activation can be formally presented as follows in (5):

$$\theta(x) = \frac{1}{1 + \exp(-x)} x \in R. \quad (5)$$

The sigmoid activation function is ideal for cases where there is a need to express the predictions in the range  $[0, 1]$ . Furthermore, sigmoid function is monotone increasing which means  $\lim_{n \rightarrow +\infty} \theta(x) = 1$  and  $\lim_{n \rightarrow +\infty} \theta(x) = 0$ . However, this fact may result in the vanishing gradients issue. That is, the gradient of  $\theta(x)$  approximates to zero when neurons become so large due to the input  $x$  being very far from zero. Subsequent optimization becomes very hard to achieve.

The next activation function, relu, is mathematically defined in (6):

$$ReLU(x) = \max(0, x) x \in R \quad (6)$$

The resulting gradient  $relu(x) = 1$  when  $x > 0$  and  $relu^-(x) = 0$  when  $x < 0$ . The convergence ability of relu is better than that of sigmoid.

The pooling layer in a CNN model is used to compute the statistical summary of the given inputs as well as to reduce the dimensionality without missing important information. Several types of pooling methods exist. In max pooling, the layer generates the maximum values considering each rectangular point surrounding data given a 2-D point  $(i, j)$  or a 3-D point  $(i, j, k)$  of each input feature. The average pooling computes the mean of the data surrounding the point.

Given a fully connected layer where there are  $n$  inputs and  $m$  outputs, the output of the layer is determined by the parameters:  $W \in M_{m,n}$  which is the weight matrix where  $m$  indicates the rows and  $n$  indicates the columns as well as  $b \in \mathbf{R}^m$ , a bias vector. More formally, for a vector of input,  $x \in \mathbf{R}^n$ , When an activation function  $f$  is applied to the fully connected  $FC$  layer, the resultant output can be defined as in equation 7:

$$FC(x) := f(Wx + b) \in \mathbf{R}^m \quad (7)$$

From (7),  $Wx$  represents matrix product.  $f$  is used component-wise. When designing classification models, fully-connected

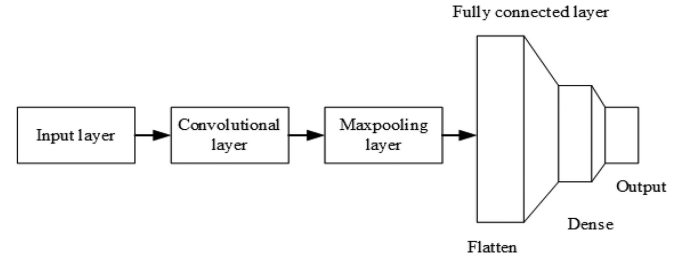


Fig. 1. CNN architecture.

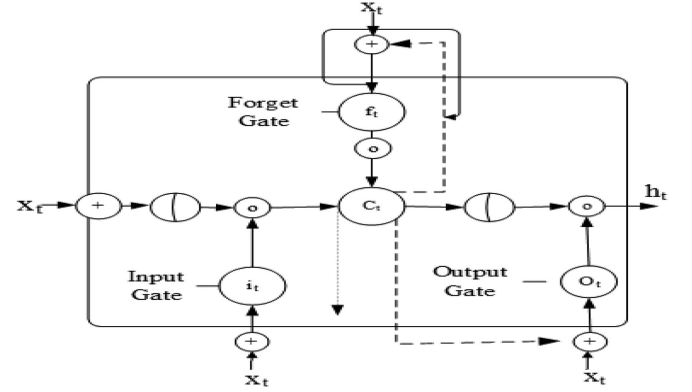


Fig. 2. LSTM architecture.

layers usually get stacked at the top of a preceding CNN as the last layers. The features extracted by the CNN are flattened into the shape of a single vector. The CNN model architecture is given in Fig. 1.

**2) Long Short Term Memory:** LSTM model first introduced by Hochreiter *et al.* [21] has been very successful in many sequence-based tasks. In practice, LSTM is the predominant Recurrent Layer applied in training sequence and time-series data. Fig. 2 depicts the LSTM algorithm's architecture and shape. According to Fig. 2, in the LSTM model, the Forget Gate initially computes the Long Term State  $c_{(t-1)}$ , and the Input Gate deletes or adds memories. The output of  $c_{(t-1)}$  is  $c_t$ . Also, a copy of  $c_{(t-1)}$  is passed via a Tanh function and then filtered with the Output Gate to obtain the resulting Short Term State  $h_t$ . This is also the cell's output at  $t$  time stamp,  $y_t$ . The RNN cell consists of one fully connected layer, which generates results  $g_t$ . However, LSTM Cell contained three additional Gate Controllers Layers. After activation with the logistic function, the output is ranged from zero to one inclusive.

The results of the element-wise multiplication operation determine the states of the gates. When the output zeros out, then the state of the Gate is set to off, and when the results are ones, the state of the Gate becomes on. The input gate decides on the information to include in the Long Term State, while the removal of information from the Long Term-State is determined by the forget Gate. The output Gate selects which sections to read and return as output at any given time step. Consider an input vector of the sequence  $x$  as an illustration, and  $W$  are the weights related to each matrix element. The stages included in the LSTM cell are presented in (8)–13.

$$i_t = \sigma(W_{xi}^T x_t + W_{hi}^T h_{(t-1)} + b_i) \quad (8)$$

$$f_t = \sigma(W_{xf}^T x_t + W_{hf}^T h_{(t-1)} + b_f) \quad (9)$$



$$o_{(t)} = \sigma(W_{xo}^T x_{(t)} + W_{ho}^T h_{(t-1)} + b_o) \quad (10)$$

$$g_{(t)} = \tanh(W_{xg}^T x_{(t)} + W_{hg}^T h_{(t-1)} + b_g) \quad (11)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \quad (12)$$

$$y_{(t)} = h_t = o_{(t)} \otimes \tanh(c_{(t)}) \quad (13)$$

**3) Integrated Framework (CNN-LSTM):** We have proposed an integrated framework (CNN-LSTM) by combining the unique capabilities of CNN and LSTM deep learning models to classify brain tumors. The integrated architecture (CNN-LSTM) employs CNN layers to extract deep, complex features of brain MRI and these extracted features are embedded into the LSTM model for the final classification. In the framework, the convolution layers are 12, max-pooling layers are 5, and there is 1 fully connected layer followed by 1 LSTM layer and 1 final output layer with softmax function. Each convolution block is integrated with 2 or 3 2D convolution layers, 1 Max-pooling layer and a drop-out layer. For feature extraction, the convolutional layer with a size of  $3 \times 3$  kernels is applied along with Relu activation. Max-pooling layer of  $2 \times 2$  kernel size is incorporated to reduce the input image size. The resulting function map is then passed to the LSTM layer for information extraction.

**4) Data Augmentation:** To improve the predictive performance of our CNN-LSTM model, we have used data augmentation techniques. The data augmentation can resolve the problem of insufficient data for model training. We utilized augmentations such as rotation, zooming, and brightness on the original images to create new sample data bearing the same label. We applied rotation from right to left by 45 degrees and increased the zooming and brightness of all images. The augmented data sets were utilized to effectively train the proposed CNN-LSTM model.

**5) Model Cross-Validation Criteria:** We used the hold-out Cross-validation technique for validating our proposed model [22]–[24]. In this study, we split each brain tumor MRI data set into training and testing sets of 70% and 30%, respectively.

**6) Model Assessment Criteria:** The evaluation metrics used to assess the predictive output of the proposed model include accuracy(Acc), precision(Pre), specificity(Sp), sensitivity(Sn), F1-score(F1), Matthews' correlation coefficient(MCC), as well as Area Under the Curve (AUC) [25], [26].

**7) Proposed Integrated (CNN-LSTM) Classification Model:** We have designed an integrated framework (CNN-LSTM) based on CNN and LSTM deep learning models to classify brain cancer using clinical brain MRI image data in the MIoT industry. In the integrated architecture, convolutional layers in the CNN part were used for deep, complex features extraction from brain MRI image data, and these extracted features were embedded into our LSTM network for classification. Brain MRI image data sets were utilized to evaluate the CNN-LSTM model.

The data set  $X(I, T)$  with the original images were fed to the CNN model, which performs a deep feature extraction to produce a new set  $X'(I, T)$ , where  $I$  and  $T$  are input images and target output labels, respectively. The expression is shown in (14).

$$X(I, T) = \{(I_i, T_i) | I_i \in R^n, T_i \in \{0, 1, 2\}\}_{i=1}^k \quad (14)$$

---

**Algorithm 1: SGD.**


---

**Input:** Training data  $D_i$ , regularization parameter  $\lambda$ , learning rate  $\eta$ , initialization  $\sigma$   
**Output:** Parameters  $\Theta = (\omega_0, w, V)$

```

1  $\omega_0 \leftarrow 0; w \leftarrow (0, \dots, 0); V \leftarrow \mathcal{N}(0, \sigma);$ 
2 repeat
3   for  $(x, y) \in S$  do
4      $\omega_0 \leftarrow \omega_0 - \eta \left( \frac{\partial}{\partial \omega_0} \ell(\hat{y}(x|\Theta), y) + 2\lambda^0 \omega_0 \right);$ 
5     for  $i \in \{1, \dots, p\} \wedge x_i \neq 0$  do
6        $\omega_i \leftarrow \omega_i - \eta \left( \frac{\partial}{\partial \omega_i} \ell(\hat{y}(x|\Theta), y) + 2\lambda_{\pi(i)}^\omega \omega_i \right);$ 
7       for  $f \in \{1, \dots, k\}$  do
8          $\nu_{i,f} \leftarrow \nu_{i,f} -$ 
9            $\eta \left( \frac{\partial}{\partial \nu_{i,f}} \ell(\hat{y}(x|\Theta), y) + 2\lambda_{f, \pi(i)}^\nu \nu_{i,f} \right);$ 
10        end
11      end
12 until stopping criteria;
```

---

Where  $I_i$  can be written as in (15).

$$I_i = \{I_1, I_2, \dots, I_n\} \quad (15)$$

Before passing the data set  $X(I, T)$  into the CNN model, we applied some data transformations to increase the data set size so as to train the model. Additionally, the number of epochs  $E$ , model parameters  $w$ , Learning Rate (LR)  $\eta$ , size of batch  $b$ , and the number of layers in both CNN and LSTM were adjusted accordingly as described in section II-B3. For the optimization of our model parameters, we have used the Stochastic Gradient Descend (SGD) [27]. This optimization algorithm is known to obtain good performance across a broad range of deep learning architectures and has served as the foundation for a number of different optimization approaches. The SGD uses the same learning rate for all parameters without performing any form of adaptation. It does the computation on mini-batches (subsets of the data) at a time. When all the mini-batches are processed by performing both forward and backward passes via the network, then an epoch is completed. Tuning of hyper-parameters in order to find the most optimal for any given task is usually very difficult to carry out as it requires testing a very wide range of possible values and their combinations [28]. The pseudocode for the SGD optimization technique is given in algorithm 1. Model Performance was computed using different evaluation metrics with test data under the hold-out Cross-validation. The proposed approach is formally presented in algorithm 2 and graphically illustrated in Fig. 3. In the figure, C, L, and D represent CNN, LSTM, and dense layers, respectively.

### III. EXPERIMENTS

#### A. Setup

We implemented our method and performed several experiments. We used three brain MRI data sets for testing our model. The performance of the CNN-LSTM model after augmenting the original data sets was also checked. Using the hold-out approach, the data set was divided into 70% and 30% for training and testing, respectively, for all the experiments. We used the SGD

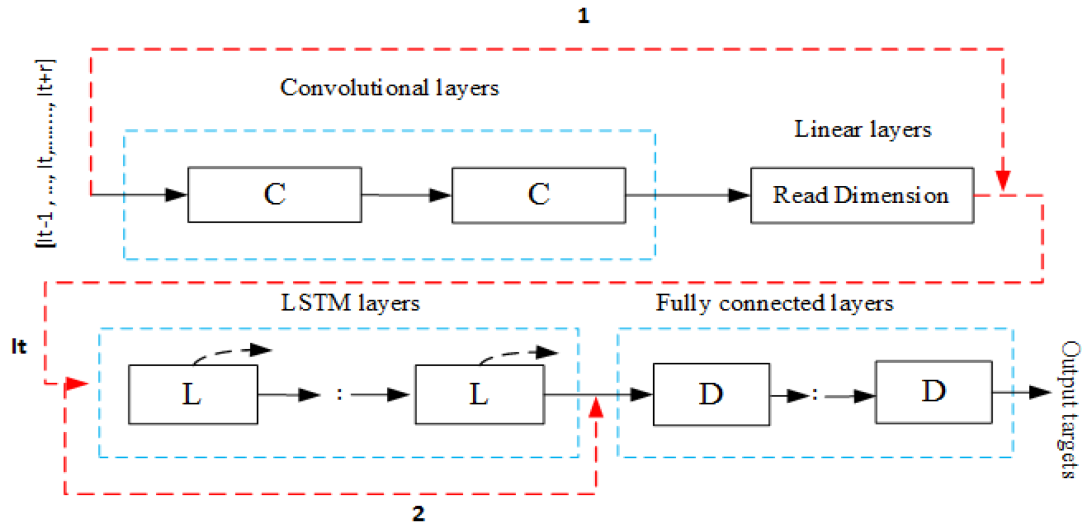


Fig. 3. Integrated CNN-LSTM diagnosis model, where C, L and D represent CNN, LSTM and dense layers respectively.

algorithm [27] to optimize the training parameters. Furthermore, other hyper-parameters were set as follows: The learning rate (LR)  $\alpha$  for the SGD was set to 0.0001. The experiments were run for 140 epochs with a batch size of 120. The Mini-batch size used was 9. Softmax was used as the model's outer activation function, while Relu was utilized for the inner activation in all the experiments.

The hardware setup for every experiment carried out consisted of a computer running a windows 8 operating system and equipped with a GPU. The software's requirements for all experiments were python v3.7, and the CNN-LSTM model was implemented using the Keras v2.2.4 framework as a high-level API and Tensorflow v1.12 for the back-end. Each experiment was iterated several times to obtain results that were stable.

## B. Results Analysis

1) *Data Pre-Processing Results:* We have used three data sets for our model validation. The first data set is the BTDS [11]. This data set comprises contrast-enhanced T1-weighted images of 233 subjects organized into 3 brain tumor classes; Pituitary, Meningioma, and Glioma. A total of 3064 slices were taken. The Meningioma subjects are 82 with 708 slices, and glioma subjects are 91 with 1426 slices. At the same time, pituitary subjects are 60 with 930 slices. There is an imbalance considering the proportions of the various classes, which can lead to overfitting. Hence to balance the Meningioma, Glioma, and pituitary portions in the data set, we have incorporated data augmentation [27] methods to transform the original samples in the data sets using rotation, zooming, brightness. Due to this augmentation process, the data set is balanced. Additional data is created to train the model properly so as to increase the model performance by rotation along the x-axis from right to left with a degree of 45 and increasing the zooming and brightness of all types of images of the original data set. Due to these augmentation processes, the size of the newly created data set is 21,448.

The second data is the MBNDS, which includes multi-sequence MRI images of 130 subjects, [18]. The contrast-enhanced T1-weighted images consist of different grades of Glioma (2, 3, 4). The number of slices for Glioma grades with the

TABLE I  
IMAGES IN ORIGINAL AND AUGMENTED DATA SETS

Data set name	original data set images	augmented data set images
BTDS	3064	21448
MBNDS	516	3612
BMIDS	253	1771

corresponding number of patients and slices is as follows; grade 2 has 33 patients with 205 slices, grade 3 have 19 patients with 129 slices, and grade 4 has 21 patients with 182 slices. Thus MBNDS data set contains a total of patients 73, and the total number of images is 516. We also applied data augmentation techniques such as rotation, zooming, and brightness of all types of images of the original data set. By applying rotations along the x-axis from right to left with degree 45 and zooming and brightness, the new data set is created of 3612 images.

The third data set BMIDS includes 253 MRI brain images. In the data set, 155 images belong to the tumor class, and 98 images belong non-tumor class. The same augmentation techniques have been applied, and a new data set of size 1771 was created. The original and augmented data set sizes are reported in Table I.

2) *Results of the CNN and LSTM Models on Original and Augmented Data Sets:* The individual results of the CNN and LSTM models on original data sets are presented in Table II. The CNN model obtained 95.01%, 96.91%, and 93.55% accuracy with data sets BTDS, MBNDS, and BMIDS, respectively, with original data sets. While the LSTM model obtained an accuracy of 96.60%, 95.90% and 95.10% with data sets BTDS, MBNDS and BMIDS, respectively, as presented in Table II. The other performance evaluation measures are also reported in Table II of both models.

Similarly, both models' predictive performances with augmented data sets are recorded in Table III under the same parameters. Table III presented that CNN and LSTM obtained improved performance in terms of accuracy with augmentation data sets. The CNN model with augmented data set BTDS obtained 96.98% accuracy, and 97.50% accuracy was obtained with MBNDS data set. While with the augmented BMIDS data set, the CNN achieved 95.44% accuracy. Table III show that the

---

**Algorithm 2:** CNN-LSTM Intergrated Framework for Deep Feature Selection and Classification of Brain Tumors.

---

**Input:**  $E$ : Epochs;  $w$ : Model parameters;  $\eta$ : Learning rate;  $b$ : Size of batch;  $X$ : Brain MRI images data set;  $X_{train}$ : Brain MRI images training data set;  $X_{test}$ : Brain MRI images validation data set;  $CNN$ : Feature extraction;  $X'$ : extracted features;  $LSTM$ : target classification

**Output:**  $P_{test}$ : CNN+LSTM model assessment using validation data set

- 1 pre-processing of the Brain MRI images data using augmentation methods transformation (rotation, zooming, brightness)
- 2 CV hold-out: 70% for training and 30% for valuation;
- 3  $X_{train} \leftarrow preData(X_{train})$
- 4  $X_{test} \leftarrow preData(X_{test})$
- 5 Initialize CNN model parameters  $w$
- 6 **Feature extraction:**
- 7 **for** Local epoch  $e \leftarrow 1$  to  $E$  **do**
- 8     **for**  $b = (I, T) \in$  random batch from  $X_i$  **do**
- 9         model parameters optimization
- 10          $w_i \leftarrow w_i - \eta(\nabla(\mathcal{L}(w_i; \mathbf{b})))$
- 11         Conv2D: Convolution operations;
- 12         MaxP2D: MaxPooling operations;
- 13          $X'$ : extracted features = Flatten the features;
- 14     **end**
- 15 **end**
- 16 Initialize upper layers of LSTM parameters  $\theta$  with extracted feature  $X'$  and parameters  $w$
- 17 **Classification training:**
- 18  $X_{train}^- \leftarrow preData(X_{train}^-)$
- 19  $X_{test}^- \leftarrow preData(X_{test}^-)$
- 20 **for** local epoch  $e \leftarrow 1$  to  $E$  **do**
- 21     **for**  $s = (I, T) \in$  random batch from  $X_{train}^-$  **do**
- 22         Update model parameters
- 23         Time distributed(cnn ...)
- 24         LSTM layers
- 25          $\theta_i \leftarrow \theta_i - \eta(\nabla(\mathcal{L}(\theta_i; \mathbf{s})))$
- 26     **end**
- 27     LSTM output layer softmax
- 28 **end**
- 29 Model assessment with test set:
- 30      $P_{test} \leftarrow computeMetrics(\theta, X_{test}^-)$
- 31 **return**  $P_{test}$

---

LSTM model obtained 96.78%, 97.80 and 95.93% accuracy with augmented data sets BTDS, MBNDS and BMIDS respectively.

Hence, [Table II](#) and [Table III](#) presented that with data augmentation process the CNN model improved in terms of accuracy from 95.01% to 96.98%, from 96.91% to 97.50% and from 93.55% to 95.44% respectively with datasets BTDS, MBNDS and BMIDS. The accuracy of the LSTM model also increased from 96.60% to 96.78% with the augmented BTDS data set. Also, with the augmented MBNDS data set, accuracy increased from 95.90% to 97.80%. Finally, with BMIDS augmented data set, the LSTM improved accuracy from 95.10% to 95.97%.

From [Table II](#) and [Table III](#) we concluded that the predictive output of deep learning models could be improved with enough training data. So data augmentation process has significant importance in obtaining high predictive performance.

**3) Results of the Integrated CNN-LSTM Model on Original Data Sets:** The results of the CNN-LSTM model when evaluated on the original data sets are discussed in this section. Essential hyper-parameters such as the SGD optimization technique

with a LR of 0.0001, 140 training epochs, and batch sizes of 120 for each forward and backward pass were used to configure the CNN-LSTM model. Also, each data set was partitioned into 70% training as well as 30% validation sets. Different evaluation metrics were utilized in evaluating the performance of the approach. All these hyperparameter values and the output of the experimental outcomes are reported in [Table IV](#).

The outcomes shown in the table indicate that the CNN-LSTM model obtained 98.78% accuracy, 88.45% specificity, 97.87% sensitivity, 96.82% precision, MCC of 97.98%, F1-score of 98.40%, and AUC of 98.10% when trained with the original BTDS. Further, the proposed CNN-LSTM model obtained 97.91% accuracy, 99.17% specificity, 87.92% sensitivity, and 98.00% precision. Other results are MCC of 94.67%, F1-score of 96.45% and AUC of 97.90% on the original MBNDS. On the other hand, with BMIDS, the proposed model obtained 96.70% accuracy, 97.25% specificity, 92.23% sensitivity, 98.76% precision, 95.15% MCC, 97.42% F1-score, and 97.36% AUC.

According to the experimental outcomes, we concluded that the proposed CNN-LSTM model obtained very high performance with all three data sets. With respect to the accuracy metric, the proposed model attained higher performance on the BTDS compared to the accuracy achieved when the MBNDS and BMIDS data sets were used.

**4) Results of the Integrated CNN-LSTM Model on Augmented Data Sets:** The proposed CNN-LSTM model performance has been evaluated on augmented data sets. The CNN-LSTM model was configured with essential hyperparameters like the SGD optimizer with a learning rate of 0.0001, 140 training epochs, and a batch size of 120. 70% of each data set was set for training, and 30% was set to validate the model. Different evaluation metrics were used for model performance evaluation. All these hyperparameter values and experimental results were tabulated in [Table V](#).

In [Table V](#), it is shown that the proposed CNN-LSTM model obtained 99.93% accuracy, 99.32% specificity, 95.36% sensitivity, and 98.76% precision. Also, MCC of 98.10%, F1-score of 99.09% as well as AUC of 99.00% were attained when trained using the augmented BTDS. Further, our proposed CNN-LSTM model obtained 99.22% accuracy, 100.00% specificity, 98.10% sensitivity, and 93.98% precision. Also, MCC of 97.45%, F1-score of 99.45%, and AUC of 99.31% were attained on the augmented MBNDS. On the other hand, with augmented BMIDS, the proposed model obtained an accuracy of 98.67%, a specificity of 99.87%, a sensitivity of 100.00%, a precision of 98.65%, MCC of 97.77%, F1-score of 98.33% and AUC of 98.87%.

From the outcomes of the experiments, we concluded that the CNN-LSTM approach resulted in very high performance with all three augmented data sets. According to [Tables IV](#) and [V](#), the accuracy of the CNN-LSTM model increased from 98.78% to 99.93% when the model was trained using the augmented BTDS data. Also, when the augmented MBNDS data set was used, the CNN-LSTM model attained improvement from 97.91% to 99.22% accuracy. Similarly, the model accuracy increased from 96.70% to 98.67% when trained with the augmented BMIDS data set. The CNN-LSTM model obtained high performance with data augmentation, demonstrating that data augmentation is necessary when the original data set size is small. The high performance of the proposed model might be due to the proper setting of hyperparameters of the model and data augmentation.

**TABLE II**  
CNN AND LSTM MODEL INDIVIDUAL PERFORMANCE ON ORIGINAL DATA SETS

Model	Data set	Parameter		Performance Evaluation Metric						
		Optimizer	LR	Acc (%)	Sp (%)	Sn/Rec (%)	Pre(%)	MCC (%)	F1 (%)	AUC (%)
CNN	BTDS	SGD	0.0001	95.01	83.23	93.11	92.00	79.34	92.32	94.44
	MBNDS	-	-	96.91	96.11	89.12	96.03	93.22	92.32	91.23
	BMIDS	-	-	93.55	97.42	90.12	96.34	90.10	95.00	94.19
LSTM	BTDS	SGD	0.0001	96.60	90.19	95.67	97.01	98.09	95.32	96.00
	MBNDS	-	-	95.90	100.00	81.45	88.70	92.97	93.88	96.00
	BMIDS	-	-	95.10	99.12	99.00	91.86	93.09	94.00	95.76

**TABLE III**  
CNN AND LSTM MODELS INDIVIDUAL PERFORMANCE ON AUGMENTED DATA SETS

Model	Data set	Parameter		Performance Evaluation Metric						
		Optimizer	LR	Acc (%)	Sp (%)	Sn/Rec (%)	Pre(%)	MCC (%)	F1(%)	AUC (%)
CNN	BTDS	SGD	0.0001	96.98	99.91	96.12	95.98	98.90	95.10	96.93
	MBNDS	-	-	97.50	100.00	89.02	95.03	97.08	88.43	96.98
	BMIDS	-	-	95.44	99.12	95.50	97.23	93.89	98.00	97.06
LSTM	BTDS	SGD	0.0001	96.78	100.00	96.81	93.00	91.87	96.37	97.62
	MBNDS	-	-	97.80	97.60	99.00	92.80	96.32	98.00	98.11
	BMIDS	-	-	95.93	100.00	97.87	93.79	97.99	95.00	96.00

**TABLE IV**  
CNN-LSTM MODEL PERFORMANCE ON ORIGINAL DATA SETS

Data set	Parameter		Performance Evaluation Metric						
	Optimizer	LR	Acc (%)	Sp (%)	Sn/Rec (%)	Pre (%)	MCC (%)	F1(%)	AUC (%)
BTDS	SGD	0.0001	98.78	88.45	97.87	96.82	97.98	98.40	98.10
MBNDS	-	-	97.91	99.17	87.92	98.00	94.67	96.45	97.90
BMIDS	-	-	96.70	97.25	92.23	98.76	95.15	97.42	97.36

**TABLE V**  
CNN-LSTM MODEL PERFORMANCE ON AUGMENTED DATA SETS

Data set	Parameter		Performance Evaluation Metrics						
	Optimizer	LR	Acc (%)	Sp (%)	Sn/Rec (%)	Pre (%)	MCC (%)	F1 (%)	AUC (%)
BTDS	SGD	0.0001	99.93	99.32	95.36	98.76	98.10	99.09	99.00
MBNDS	-	-	99.22	100.00	98.10	93.98	97.45	99.45	99.31
BMIDS	-	-	98.67	99.87	100.00	98.65	97.77	98.33	98.87

Furthermore, the performance on other metrics such as AUC and F1-Score also improved with data augmentation processes. The AUC with augmented data set (BTDS) increased from 98.10% to 99.00%, and with augmented BMIDS data set, the AUC increased from 97.90% to 99.31%. Lastly, with the augmented BMIDS, the AUC increased from 97.36% to 98.87%. Similarly, the F1-score values of the proposed model increased with data augmentation.

**5) Comparison With Baseline Methods:** We have compared the accuracy performance of our model (CNN-LSTM) with baseline methods in Table VI. The results in the table show that our proposed model (CNN-LSTM) reached 99.89% accuracy with the BTDS data set, while the model obtained 99.22% accuracy with the MBNDS data set and 98.76% with the BMIDS

data set as compared to existing models. The outcome of our method demonstrates that it correctly classifies brain tumors and can be deployed in MIOThealthcare systems.

**6) Discussion:** The brain cancer diagnosis is critically essential for on-time treatment and recovery in MIOThealthcare. IoT consists of various devices connected via different network mediums transmitting to a centralized server or exchanging such data to achieve particular purposes. Likewise, MIOThealthcare can include a vast range of medical technologies linked together to collect relevant data for analysis and the provision of fast and more effective prognosis, diagnosis, treatment, logistics, and other services in a timely manner. The success achieved by Artificial Intelligence over a wide range of application domains is likewise extended to making MIOThealthcare smarter and more intelligent at connecting



**TABLE VI**  
ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS

Data set	Method of classification	Accuracy (%)	Reference
BTDS	FS+FS+Classification	91.28	[11]
	CNN	90.89	[13]
	GA-CNN	94.2	[14]
	CNN	96.13	[7]
	Proposed method (CNN-LSTM)	99.93	2022
MBNDS	SVM and KNN	85, 88	[10]
	CNN	98.7	[7]
	KNN	98	[12]
	Proposed method (CNN-LSTM)	99.22	2022
BMIDS	Proposed method (CNN-LSTM)	98.67	2022

medical experts, patients and the devices. AI-based computer-aided diagnosis (CAD) systems with well-trained models accurately diagnose diseases than medical professionals because the medical experts are humans and may not always correctly interpret medical data such as Chest X-rays, CT Scans, and MRI to diagnose the disease at the early stages [4], [29], [30]. With the appropriate IoT devices and communication infrastructure, the AI-based CAD can be easily integrated into MIIoTs systems. This can further provide services like Tele-monitoring, allowing medical institutions to obtain real-time medical information about the conditions of their patients.

The deep learning-based technique for brain cancer diagnosis with MRI images significantly improves brain cancer diagnosis in IoT healthcare systems. The CNN component of the model extracts valuable features from the images and these features aid in classifying cancerous and non-cancerous images. This study proposed an intelligent integrated IoT framework for brain cancer diagnosis employing CNN and LSTM models. We used the CNN model to extract deep features, which we fed to the LSTM model for the final classification into the tumor class. Three brain cancer image data sets were used for validating the model. The data sets consisted of small samples, hence not very effective when used to train the model. In order to improve the predictive performance, we used data augmentation [27] techniques to add additional samples by transforming the existing ones during training. By this, the data sizes increased, which resulted in a much more effective training process, and the model achieved excellent performance. The results from the experiments demonstrate that the model attained better performance when run using the augmented data sets.

The proposed CNN-LSTM model obtained 98.78% accuracy, 88.45% specificity, 97.87% sensitivity, 96.82% precision, 97.98% MCC, 98.40% F1-score and 98.10% AUC respectively on the original BTDS. Further, the proposed CNN-LSTM model obtained 97.91% accuracy, 99.17% specificity, 87.92% sensitivity, and 98.00% precision. Also, MCC of 94.67%, F1-score of 96.45%, and AUC of 97.90% were reported on the original MBNDS. On the other hand, with BMIDS, the proposed model obtained 96.70% accuracy, 97.25% specificity, 92.23% sensitivity, 98.76% precision, 95.15% MCC, 97.42% F1-score, and 97.36% AUC. The high 98.78% accuracy of the CNN-LSTM model with the BTDS data set shows that this data set is more suitable for training our model than the BMIDS and the MBNDS. The model's accuracy shows its overall correct performance, so this metric is essential for model evaluation. The proposed model specificity and sensitivity results on all

**TABLE VII**  
GPU MEMORY CONSUMED BY MODELS DURING TRAINING PROCESS

Model	Training parameters (millions)	GPU Memory (Mbs)
CNN-LSTM	29	1221

the data sets also show high performance, indicating that the proposed model effectively classifies brain cancer and healthy subjects.

In addition, the proposed CNN-LSTM model obtained 99.93% accuracy, 99.32% specificity, 95.36% sensitivity, 98.76% precision, 98.10% MCC, 99.09% F1-score and 99.00% AUC respectively with the augmented BTDS and obtained 99.22% accuracy, 100.00% specificity, 98.10% sensitivity, 93.98% precision, 97.45% MCC, 99.45% F1-score and 99.31% AUC with the augmented MBNDS. On the other hand, with augmented BMIDS, the proposed model obtained 98.67% accuracy, 99.87% specificity, 100.00% sensitivity, 98.65% precision, 97.77% MCC, 98.33% F1-score, and 98.87% AUC.

The accuracy of the proposed CNN-LSTM model increased from 98.78% to 99.93% when the model was trained using the augmented BTDS data set. There was also an improvement in the accuracy from 97.91% to 99.22% when the model was trained using the augmented MBNDS data set. Similarly, the model's accuracy increased from 96.70% to 98.67% when the augmented BMIDS data set was used. The improvement of the model as a result of the application of augmentations indicates that large data sets are required to attain high performance during training. Apart from the augmentations, the high-performance outcome of our proposed model might also result from the proper setting-up of hyperparameters during training. Furthermore, the other metrics such as AUC and F1-Score also improved due to the augmentation process. The AUC increased from 98.10% to 99.00% with the augmented BTDS data set. Likewise, using the MBNDS data set, the model achieved an increase in AUC from 97.90% to 99.31%. Lastly, the augmented BMIDS data set also improved the AUC from 97.36% to 98.87%. The F1-score values also improved in a similar fashion when the training was boosted with the augmented data sets. The proposed method achieved higher performance than other state-of-the-art models, as shown in Table VI. Thus, we reached the conclusion that our proposed CNN-LSTM approach is highly suitable for diagnosing brain cancer in the IoT healthcare industry.

The CNN-LSTM model training time and memory consumption increased with an increase in the number of epochs and decreasing the learning rate. It also resulted in very high consumption of GPU resources with higher computational complexity. The memory consumed when training the proposed CNN-LSTM model without data augmentation was lower than training with the augmented data. The GPU memory usage by the CNN-LSTM model is reported in Table VII. Table VII demonstrates the proposed integrated diagnosis framework (CNN-LSTM) memory complexity.

#### IV. CONCLUSION

AI techniques, specifically deep learning algorithms, are primarily used to analyze medical image data in MIIoT healthcare systems. The AI-based computer automated diagnosis (CAD)



system can more effectively diagnose the disease than medical professionals. Researchers have proposed different diagnostic systems, however, these systems still need further improvement to diagnose the disease correctly. To tackle this problem, we proposed a new intelligent integrated model (CNN-LSTM) based on deep learning techniques to diagnose brain cancer accurately. Our model uses a CNN model to extract deep features from image data and pass the features to an LSTM model for classification into tumor and non-tumor classes. Three data sets have been used for the model evaluation. Further, we performed some data augmentations to increase each data set size to train the model effectively. Also, the hold-out Cross-validation method was incorporated to select the best model and for hyperparameters tuning. A training split of 70% and a validation split of 30% were used. We computed the performance of the model on standard medical evaluation metrics. The results of the experiments demonstrate that the proposed integrated diagnosis framework (CNN-LSTM) obtained very high performance compared to the baseline methods in the IoT industry. The resulting high performance of the model might be due to the effective pre-processing of data and the adjustment of other parameters of the model, such as the number of layers, optimizer and activation functions, and data augmentation. As a result of the high performance of our method (CNN-LSTM), we recommend its usage in MIIoT to detect brain cancer.

The proposed model will be used to diagnose other diseases in the future, such as heart disease, breast cancer, Parkinson's disease, diabetes, and lung cancer. In addition, we will use deep learning, transfer learning, and federated learning techniques to classify tumors and diagnose brain cancer in MIIoT-healthcare systems.

## REFERENCES

- [1] D. N. Louis and Ohgaki, "The 2016 World Health Organization classification of tumors of the central nervous system: A summary," *Acta Neuropathologica*, vol. 131, no. 6, pp. 803–820, 2016.
- [2] Z. N. K. Swati *et al.*, "Content-based brain tumor retrieval for MR images using transfer learning," *IEEE Access*, vol. 7, pp. 17809–17822, 2019.
- [3] S. Pereira, R. Meier, V. Alves, M. Reyes, and C. A. Silva, "Automatic brain tumor grading from MRI data using convolutional neural networks and quality assessment," in *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Berlin, Germany: Springer, 2018, pp. 106–114.
- [4] R. Ani and S. Krishna, "IoT based patient monitoring and diagnostic prediction tool using ensemble classifier," in *Proc. Int. Conf. Adv. Comput., Commun. Informat.*, 2017, pp. 1588–1593.
- [5] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote e-health," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 54–61, 2021.
- [6] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
- [7] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-classification of brain tumor images using deep neural network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019.
- [8] S. Kumar *et al.*, "Deep learning framework for recognition of cattle using muzzle point image pattern," *Measurement*, vol. 116, pp. 1–17, 2018.
- [9] M. Elhoseny, G.-B. Bian, S. Lakshmanaprabu, K. Shankar, A. K. Singh, and W. Wu, "Effective features to classify ovarian cancer data in internet of medical things," *Comput. Netw.*, vol. 159, pp. 147–156, 2019.
- [10] E. I. Zacharaki *et al.*, "Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme," *Magn. Reson. Med., Official J. Int. Soc. Magn. Reson. Med.*, vol. 62, no. 6, pp. 1609–1618, 2009.
- [11] J. Cheng, Z. Huang, and Q. Feng, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLoS One*, vol. 10, no. 10, 2015, Art. no. e0140381.
- [12] E.-S. A. El-Dahshan, T. Hosny, and A.-B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digit. Signal Process.*, vol. 20, no. 2, pp. 433–441, 2010.
- [13] P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries," in *Proc. ICASSP 2019-2019 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 1368–1372.
- [14] A. K. Anaraki, M. Ayati, and F. Kazemi, "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms," *Biocybernetics Biomed. Eng.*, vol. 39, no. 1, pp. 63–74, 2019.
- [15] K. Yu *et al.*, "Securing critical infrastructures: Deep-learning-based threat detection in IIoT," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 76–82, Oct. 2021.
- [16] J. Cheng, "Brain Tumor Dataset," *Discov. Res. Figshare*, Accessed: Jan. 10, 2021, doi: [10.6084/m9.figshare.1512427.v5](https://doi.org/10.6084/m9.figshare.1512427.v5).
- [17] K. Clark and Vendt, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, 2013.
- [18] L. Scarpace, A. E. Flanders, R. Jain, T. Mikkelsen, and D. W. Andrews, "Data from REMBRANDT. The cancer imaging archive," 2015, <https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT#35392299515cc672b974080a1394cbe9c649c74>
- [19] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, 2019, Art. no. 5.
- [20] A. U. Haq *et al.*, "Stacking approach for accurate invasive ductal carcinoma classification," *Comput. Elect. Eng.*, vol. 100, 2022, Art. no. 107937.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [22] A. u. Haq and J. P. Li, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Inf. Syst.*, vol. 2018, 2018, Art. no. 3860146.
- [23] A. U. Haq and J. P. Li, "Diagnostic approach for accurate diagnosis of COVID-19 employing deep learning and transfer learning techniques through chest X-ray images clinical data in e-healthcare," *Sensors*, vol. 21, no. 24, 2021, Art. no. 8219.
- [24] A. K. Singh, B. Kumar, M. Dave, and A. Mohan, "Multiple watermarking on medical images using selective discrete wavelet transform coefficients," *J. Med. Imag. Health Informat.*, vol. 5, no. 3, pp. 607–614, 2015.
- [25] A. U. Haq, J. P. Li, and W. Zhou, "Detection of breast cancer through clinical data using supervised and unsupervised feature selection techniques," *IEEE Access*, vol. 9, pp. 22090–22105, 2021.
- [26] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C.-W. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2698–2707, Feb. 2021.
- [27] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. vol. 1. Cambridge, MA, USA: MIT Press, 2016.
- [28] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305 2012.
- [29] A. U. Haq *et al.*, "3DCNN: Three-layers deep convolutional neural network architecture for breast cancer detection using clinical image data," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, 2020, pp. 83–88.
- [30] M. H. Memon, J. P. Li, and A. U. H. Haq, "Breast cancer detection in the IOT health environment using modified recursive feature selection," *Wireless Commun. Mobile Comput.*, vol. 2019, 2019, Art. no. 5176705.