Heliyon 9 (2023) e21520

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Healthcare As a Service (HAAS): CNN-based cloud computing model for ubiquitous access to lung cancer diagnosis ☆,☆☆

Nuruzzaman Faruqui [a,b], Mohammad Abu Yousuf [a,*], Faris A. Kateb [c], Md. Abdul Hamid [c], Muhammad Mostafa Monowar [c]

[a] Institute of Information Technology (IIT), Jahangirnagar University, Savar, Dhaka, 1342, Bangladesh
[b] Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka, 1216, Bangladesh
[c] Department of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

ARTICLE INFO

Keywords:
Lung cancer classification
Convolutional neural network
Computer tomography images
Internet of medical things
Cloud computing
Optimization algorithms
CNN
IoMT
CT

ABSTRACT

The field of automated lung cancer diagnosis using Computed Tomography (CT) scans has been significantly advanced by the precise predictions offered by Convolutional Neural Network (CNN)-based classifiers. Critical areas of study include improving image quality, optimizing learning algorithms, and enhancing diagnostic accuracy. To facilitate a seamless transition from research laboratories to real-world applications, it is crucial to improve the technology's usability—a factor often neglected in current state-of-the-art research. Yet, current state-of-the-art research in this field frequently overlooks the need for expediting this process. This paper introduces Healthcare-As-A-Service (HAAS), an innovative concept inspired by Software-As-A-Service (SAAS) within the cloud computing paradigm. As a comprehensive lung cancer diagnosis service system, HAAS has the potential to reduce lung cancer mortality rates by providing early diagnosis opportunities to everyone. We present HAASNet, a cloud-compatible CNN that boasts an accuracy rate of 96.07%. By integrating HAASNet predictions with physio-symptomatic data from the Internet of Medical Things (IoMT), the proposed HAAS model generates accurate and reliable lung cancer diagnosis reports. Leveraging IoMT and cloud technology, the proposed service is globally accessible via the Internet, transcending geographic boundaries. This groundbreaking lung cancer diagnosis service achieves average precision, recall, and F1-scores of 96.47%, 95.39%, and 94.81%, respectively.

1. Introduction

Early detection of lung cancer significantly reduces the risk of fatality [1], with Computer Tomography (CT) scans being the predominant imaging diagnosis technology for this purpose. Consequently, computer-aided automatic lung cancer classification from

☆ This research work was funded by Institutional Fund Projects under grant no. (IFPIP:663-611-1443). The authors gratefully acknowledge the technical and financial support provided by the Ministry of Education and King Abdulaziz University DSR, Jeddah, Saudi Arabia.

☆☆ The authors acknowledge the National Cancer Institute and the Foundation for the National Institutes of Health, and their critical role in the creation of the free publicly available LIDC/IDRI Database used in this study.

* Corresponding author.
E-mail addresses: faruqui.swe@diu.edu.bd (N. Faruqui), fakateb@kau.edu.sa (F.A. Kateb), mabdulhamid1@kau.edu.sa (M. Abdul Hamid), mmonowar@kau.edu.sa (M.M. Monowar).
URL: https://faculty.daffodilvarsity.edu.bd/profile/swe/faruqui.html (N. Faruqui).

https://doi.org/10.1016/j.heliyon.2023.e21520
Received 27 May 2023; Received in revised form 27 September 2023; Accepted 23 October 2023

Available online 27 October 2023

CT scans has emerged as a vibrant and rapidly growing research area. Moreover, advancements in Convolutional Neural Networks (CNNs) [2] empower these models to identify cancerous nodules from CT scans with a proficiency comparable to expert radiologists [3]. The potential for reduced mortality rates, increasing demand within the medical sector, and technological maturity all contribute to the surging interest in CNN-based lung cancer classifier development. However, there is a distinction between enhancing the performance of CNNs and harnessing their capabilities to build an automatic diagnosis system. Focusing on developing and fine-tuning technology without leveraging its potential limits its overall effectiveness [4]. It has been observed that the architectural development and performance optimization of CNN-based classifiers have garnered more attention than their application in developing practical diagnosis systems [5]. This paper aims to bridge this research gap by introducing the Healthcare-As-A-Service (HAAS) model, which provides ubiquitous access to CNN-based lung cancer diagnosis systems. By harnessing the potential of cloud computing and the Internet of Medical Things (IoMT), the HAAS model offers a comprehensive solution for effectively deploying and utilizing CNN-based classifiers in diagnosing lung cancer.

Traditional lung cancer diagnosis involves visiting diagnostic centers, completing the CT scan, and getting the report from radiologists [6]. The physicians analyze the patient's condition based on the radiologist's report. That means the radiology report plays the most significant role. CNNs demonstrate the potential to automate this role and reduce dependency on radiologists. Physicians combine radiology reports and symptomatic data to analyze the current status of lung cancer and prescribe treatments. IoMT potentially scans the symptomatic data of patients and makes them accessible over the internet. Combining CNN and IoMT eliminates the need to visit physicians and radiologists, saving time and money. However, such a system must be accessible from anywhere. The cloud computing Software-As-A-Service model is an appropriate solution to beat this challenge [7]. However, cloud application development differs from traditional application development [8]. Cloud applications must be scalable with simple architecture to reduce resource consumption. The state-of-the-art sophisticated CNN architectures do not meet these criteria for their computational complexities [9]. A simple CNN, named HAASNet, has been developed in this paper which is ideal for cloud applications. The ubiquitous access to lung cancer diagnosis through the HAAS model uses HAASNet to predict lung cancer from CT images. It is an alternative to an analysis by radiologists. The physio-symptomatic data further guide this prediction. It resembles face-to-face interaction with physicians. The internet-enabled access makes the HAAS available 24×7 regardless of geographical boundaries.

The HAASNet has been designed with careful mathematical analysis to ensure maximum performance with minimal computational cost. It has made the lung cancer diagnosis service through cloud computing possible. Moreover, the combination of the IoMT data includes additional dimensions in the innovative design of the proposed HAAS. The contributions of this research have been listed below:

- Designing and implementing HAASNet through mathematical interpretation, gaining 96.07% validation accuracy.
- Evidential learning optimization through exploratory analysis of four optimization algorithms.
- Ubiquitous lung cancer diagnosis service development using cloud service model incorporating symptomatic data obtained through IoMT.
- Performance evaluation through 15 evaluation metrics from three exclusive domains: IoMT, CNN, and Cloud Computing
- Conceptualization of HAAS architecture to reduce lung cancer mortality rate by improving accessibility

The remainder of this paper is structured into seven sections. Section two offers a comprehensive literature review of cutting-edge lung cancer classification techniques employing CNNs. The proposed approach is detailed in section three, while section four provides the experimental results and an analysis of the method's performance. The ethical implications of the proposed system have been discussed in section five. Section six emphasizes the present limitations and potential future developments related to this study. Lastly, the seventh section concludes the paper by synthesizing the findings.

2. Literature review

According to the review paper published by S. Huang et al. [3], M. M. Warrier et al. [5], and S. H. Hosseini et al. [10], cloud-based high performing CNN-assistant lung cancer diagnosis service with ubiquitous access like the proposed Healthcare-As-A-Service model has not been attempted before. However, it has the potential to reduce lung cancer diagnosis expenditure significantly. Furthermore, it makes healthcare services more accessible regardless of geographical barriers. Lung cancer diagnosis from a Computed Tomography (CT) scan using a Convolutional Neural Network (CNN) is a challenging and vibrant field of research [11]. It is the key technology behind computer-aided automatic lung cancer diagnosis [12]. Sophisticated network architecture development [13], applying advanced image processing algorithms for feature enhancement [14], and forming hybrid classifiers [15] are the major research topics in this field. [16]. Analyzing the effect of transfer learning and bettering the performances of the classifiers is also a deeply explored research topic of computer-aided automatic lung cancer diagnosis system [16]. However, Software-As-A-Service (SAAS)-based application development using CNN-based lung cancer classifier is an underexplored field of research. It is the pivot point of translating the remarkable research advancement in automatic lung cancer classification into ubiquitous diagnosis service to serve mankind by making healthcare service available to the mass population at a marginal cost.

C. Venkatesh et al. [17] introduces a hybrid lung cancer detection and classification approach using Region-based Convolutional Neural Networks (RCNNs) [18] and cuckoo search [19]. RCNN using two-channel CNN extracts disease size and geographic features. The core focus of this research is to improve classification accuracy. The methods developed by A. R. Bushara et al. [20] use the combination of Convolutional Neural Networks (CNN) and Capsule Neural Networks (CapsNet) to classify lung cancer CT images. This framework achieves 95% classification accuracy with a very complicated system architecture. H. Mkindu et al. [21] develop a 3D

Table 1
Literature review summary.

Authors	Objective(s)	Method/Algorithm	Dataset	Limitation	Proposed Novelty
C. Venkatesh et al. [11]	Developing hybrid model for detecting and classifying lung cancer	RCNN and Cuckoo Search	CT Image	Computational cost and service models are ignored	Considers the computational cost and develops a feasible service model
A. R. Bushara et al. [20]	Determining risk of cancer development	CNN and Capsule Neural Networks (CapsNet)	LIDC-IRDI	Computational complexity for feasible solution development	Simple design with acceptable performance
H. Mkindu et al. [21]	Multilevel nodule candidate detection	CNN and Channel Attention Mechanisms	LUNA16	Evaluated using sensitivity and CPM	Performance evaluation form 15 different parameters
D. Kawahara et al. [22]	Developing a prediction model for Radiation pneumonitis (RP) grade ≥ 2	CNN	Non-Small-Cell Lung Cancer (NSCLC) Dataset	Dependent on Radiation pneumonitis (RP) and low poor performance	Effective on any lung cancer CT images
K. Barbouchi et al. [23]	Classify and detect lung cancer from CT images	Deep Learning	Lung-PET-CT-Dx	Assessment without feasible application analysis	Potential assessment and service model development
N. Maleki et al. [24]	Early diagnosis of lung cancer	CNN, ANN, GB, RF, and SVM	CT Image Dataset	Complicated analysis methods combining multiple algorithms	Simpler and more efficient, yet accurate approach
W. Liu et al. [25]	Developing PiaNet for lung cancer classification	CNN	LIDC-IRDI	Complex network architecture and evaluated on one criteria only	Lightweight and simple design with better performance on multiple criteria
C. Shankara et al. [26]	Lung cancer nodule classification	CNN	LIDC-IRDI	Poor performance on true negative prediction	Better performance in true negative prediction
E.A. Siddiqui et al. [27]	Enhancing the lung cancer classifier's performance	Gabor Filters and Enhanced Deep Belief Network (E-DBN)	LIDC-IRDI &LUNA16	Complex approach dependent on multiple different algorithms	Straightforward approach applicable to practical solution development

U-shaped encoding and decoding deep convolutional neural network (CNN). This unique approach uses channel attention techniques to classify lung cancer nodules. The k-fold cross-validation at $k = 10$ shows 98.65% sensitivity and a competition performance metric (CPM) of 0.911.

D. Kawahara et al. [22] proposes a Radiation Pneumonitis (RP) centered classification method using CNN. It classifies Non-Small-Cell Lung Cancer (NSCLC) nodules with 75.7% accuracy. K. Barbouchi et al. [23] used deep learning to assess the ability of PET/CT scans to categorize and diagnose lung cancer. This approach automates lung cancer anatomical localization from PET/CT scans and classifies tumors with an average accuracy of 95.5%. N. Maleki et al. [24] combined CNN with Artificial Neural Network (ANN) for initial lung cancer classification from CT images. Later, they applied Gradient Boosting (GB), Random Forest (RF), and Support Vector Machine (SVM) on numerical features of the same image dataset as the second approach. Finally, they compared the results from both approaches. This combined methodology achieved 95% accuracy.

A unique CNN named PiaNet was introduced by W. Liu et al. [25]. It detects ground-glass opacity nodules, abbreviated as GGO in 3D computed tomography (CT) images. The PiaNet uses multi-scale processing and is trained through a bi-stage transfer learning method on the LIDC-IDRI dataset. It achieves a sensitivity of 93.6%. A simple CNN-based lung cancer classifier developed by C. Shankara et al. [26] achieves 92.96% accuracy with 97.45% sensitivity. However, it suffers in the performance of predicting true negatives correctly. This CNN trained on the LIDC-IDRI dataset demonstrates 86.08% specificity. E.A. Siddiqui et al. [27] conducted an experiment using Gabor filters with an enhanced Deep Belief Network (E-DBN) to improve the performance of computer-aided lung cancer classifiers. The method uses Support Vector Machines (SVM) with E-DBN to classify the lung cancer nodules.

2.1. Literature review summary

The literature review summary has been listed in Table 1. It suggests that the core research focus of the computer-aided automatic lung cancer diagnosis system using CNN is developing and improving the CNN architecture. Optimization and dataset processing are key research fields as well. However, improving the performance has been the significant research interest in innovating CNN architecture development and incorporating various dataset processing and feature enhancement algorithms. The feasibility of CNN's architecture from a computational complexity perspective for widespread adoption in practical application is absent from the recent literature review [28]. In addition, several pieces of research analyzed the performance improvement on a few evaluation criteria, leaving the scope of questioning the performance from other contexts. The proposed methodology addresses the limitation of the current state-of-the-art literature review and presents a simple and lightweight CNN architecture. It has been developed to implement it on the cloud to offer a lung cancer diagnosis service. It combines the concept of IoMT to make the service accessible from anywhere. Furthermore, the proposed system has been evaluated from 15 different evaluation criteria.

Table 2
Research gap identification and proposed innovation.

Research Gap	Proposed Innovation
Advancement in CNN-based lung cancer classifier without focusing on the application domain	Development of HAAS: A full-fledged solution to lung cancer diagnosis application
CNN architectural complexity with nominal performance-computational cost trade-offs	Simpler and efficient CNN architecture with acceptable performance
Absence of symptomatic data in lung cancer diagnosis	Application of IoMT-based symptomatic data along with CNN-based diagnosis
Evaluation on limited number of metrics	Complete performance analysis using 15 evaluation metrics

2.2. *Research gap analysis*

The research gap has been analyzed from the context of cloud application usability of the lung cancer classifier from CT images. Lightweight and computationally inexpensive designs are two recommended criteria for SAAS application development [29]. Complex network architecture depends on multiple intermediate algorithms designed to increase accuracy and is unsuitable for cloud applications because of excessive resource consumption [30]. The state-of-the-art approaches in the literature review show that the research direction is moving towards the sophistication of CNN architecture in corporations with various image enhancement and classification algorithms. The study by M. M. Warrier et al. [5], S. Huang et al. [3], and M. Sachdeva et al. [31] support the research direction inferred in this paper. This research direction leaves a research gap in developing CNN architectures suitable for efficient SAAS applications. The Table 2 lists the research gaps and proposed solutions to abridge the research gaps presented in this paper's methodology section.

The proposed study has been designed based on the research gaps discovered in the state-of-the-art literature review in Table 2. It highlights four research gaps that guided the research methodology to fill these gaps. It has been observed research in this field mainly focuses on the development of a classifier instead of designing an applicable solution using them. This research gap has inspired the application model developed in this paper. Another research gap is the CNN architectural complexities. In pursuit of increasing the performance of CNNs, recent studies ignore the computational cost of CNNs, which makes it incompatible with a cloud environment. This shortcoming in the recent literature encourages the development of a computationally efficient CNN architecture. Physicians combine both CT image reports and physio-symptomatic data to diagnose lung cancer. However, recent research involves only CNN-based classification, focusing on CT images and ignoring symptomatic data. The proposed methodology combines both CT image features and physio-symptomatic data. The last research gap identified in our review is the limitation of evaluation metrics. This abridges this gap, 15 different evaluation metrics have been used in this paper.

3. Proposed methodology

The proposed methodology, illustrated in Fig. 1, combines IoMT, CNN, and Cloud Computing. It has been designed by carefully analyzing the research gaps listed in Table 2, which motivated the novelties proposed in Table 1. The methodology presented in this paper has been carried out by mathematical analysis guided by observational analysis of the performance of the implemented and motivated by the enthusiasm of abridging the research gaps of Table 2 by attaining the novelties mentioned in Table 1. Furthermore, the dataset used and data obtained from the experiments conducted in this paper have been analyzed through various statistical approaches mathematically interpreted in this section. The physio-symptomatic data of the patient in this research are obtained over the internet using IoMT. It allows physicians to analyze lung cancer symptoms without physical inspection. The lung cancer CT scans are classified using a carefully designed, simple, and computationally inexpensive CNN named HAASNet. The entire system is operated through the Healthcare-As-A-Service (HAAS) model, which ensures ubiquitous access to lung cancer diagnosis services from anywhere. The training, testing, and diagnosis are overlapping operations identified as Figs. 1(a), 1(b), and 1(d), respectively. Fig. 1(c) shows how IoMT data are secured, processed, and integrated with the diagnosis. It scans the physio-symptomatic data and CT images for further utilization by the HAAS model. The physicians get the diagnosis data on the Physician's User Interface, illustrated in Fig. 1(f). The decision about the patient's condition is made based on the prediction from HAASNet in combination with the IoMT data. Finally, the patients receive the diagnosis report and prescription on the Patient's User Interface, illustrated in Fig. 1(e).

The study combines the Internet of Medical Things (IoMT), HAASNet architecture design and implementation, and HAAS Model architecture design. It is a unique study, to our best knowledge, the first of its kind, which combines IoMT, CNN, and Cloud computing to develop a full-fledged lung cancer diagnosis service that is accessible from anywhere in the world regardless of geographic boundaries.

3.1. *Internet of medical things (IoMT) architecture*

The Internet of Medical Things (IoMT) is a sub-branch of the Internet of Things (IoT) [32]. The proposed HAAS represents a universal healthcare service model designed for accessibility from any location. However, to avail oneself of this healthcare service via the Internet, access to an IoMT device is necessary [33]. From this observation, a study related to IoMT has been carried out. The IoMT architecture has been illustrated in Fig. 2. The proposed IoMT consists of four sensors and a healthcare application. The

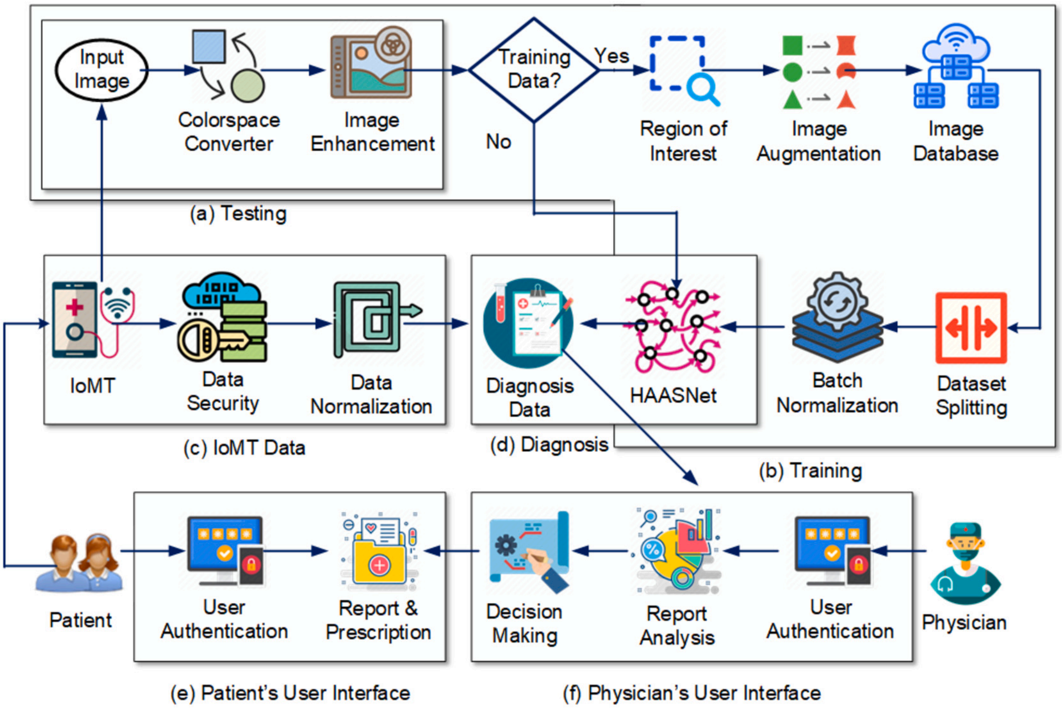

Fig. 1. The detailed overview of the proposed methodology with different elements.

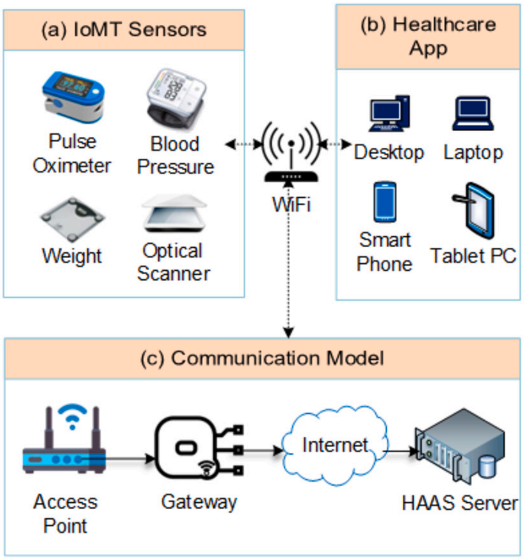

Fig. 2. IoMT sensors, application access, and communication model of HAAS Server.

application receives and processes the signals from the sensors. Along with sensor data, it has options to input data related to anorexia, anxiety, depression, pain, insomnia, constipation, and fatigue. The application processes these data before sending them to the HAAS server.

The sensor data come in different scales. Normalizing the data on the same scale is essential to train the classifier. Linear Scaling, defined in equation (1), and Z-Score normalization, defined in equation (2), have been studied in this research. Because of higher stability from experimental data, the Z-Score normalization has been applied for data normalization [34].

$$x' = \frac{(x - x_{min})}{x_{max} - x_{min}} \tag{1}$$

$$x' = \frac{(x - \mu)}{\sigma} \tag{2}$$

Table 3
IoMT normalized data from the sensors.

Parameter	Sensor	S1	S2	S3	S4
Breathing Problem	Pulse Oximeter	0.05-0.5	0.4-0.9	0.9-0.95	0.96-0.99
Heart Bit Rate	Heart Bit Rate	0.10-0.60	0.15-0.71	0.90-0.95	0.94-0.99
Blood Pressure	Sphygmomanometer	0.20-0.35	0.32-0.40	0.41-0.45	0.46-0.62
Weight	Weight Scale	0.33-0.62	0.44-0.58	0.90-0.95	0.94-0.99

Table 4
The normalized data generated by the IoMT application.

Parameter	S1	S2	S3	S4
Anorexia	0	0	0.4-65	0.4-0.8
Anxiety	0.32-0.5	0.45-0.65	0.9-0.95	0.94-0.99
Temperature	0.5-0.6	0.3-0.8	0.9-0.95	0.94-0.99
Depression	0.2-0.3	0.25-0.5	0.38-0.78	0.45-0.85
Pain	0.28-0.45	0.3-0.6	0.35-0.75	0.45-0.85
Insomnia	0.2-0.4	0.5-0.6	0.62-0.85	0.86-0.95
Constipation	0.15-0.2	0.2-0.28	0.3-0.46	0.5-0.68
Fatigues	0.18-0.4	0.23-0.45	0.62-0.77	0.8-0.9

The x' in equation (1) and (2) refers to the normalized data. The μ and σ are mean and standard deviation, respectively. The mean and standard deviations are defined by equations (3) and (4), respectively.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{4}$$

3.1.1. Sensor data

The sensor data are received from Pulse Oximeter, Sphygmomanometer, Digital Weight Machine, and Optical Scanner. The mobile application does not process the data from the Optical Scanner because of computational resource constraints. The optical sensor scans the CT images. These images are sent directly to the HAAS server for malignancy classification. The normalized sensor data without the CT images are listed in Table 3.

3.1.2. Application data

The IoMT application combines self-assessment with app-based measurement. The body temperature and insomnia are measured using the iThermonitor and the CareClinic app, respectively. The anorexia, anxiety, chest pain, constipation, and fatigue are measured through self-assessment using sliders. The normalized values of these parameters are listed in Table 4.

3.1.3. Secured communication model

The IoMT device interfaces with the healthcare application via a WiFi network. This application processes the received data before forwarding it to the access point. Given the sensitive nature of healthcare-related data, our experiment employs a 4-way handshake using the Pre-Shared Key (PSK) scheme [35]. A Pairwise Master Key (PMK) is generated through 4096 iterations, resulting in a secure 256-bit key. The chosen hash function for this system is the Password-based Key Derivation Function 2 (PBKDF2), expressed in equation (5).

$$PSK = PBKDF2(HMAC - SHA1, PW, SSID, 4096, 256) \tag{5}$$

Within equation (5), HMAC stands for Hash-based Message Authentication Code, which is responsible for generating the password hash. Here, PW denotes the Password. For this experiment, the resulting password hash is a 16-character alphanumeric string. A Password Salt (PS) is utilized to enhance security. Rather than a static PS, it's dynamically determined using the packet counter, which is then amalgamated with the Service Set Identifier (SSID).

3.1.4. IoMT to HAAS VM

The proposed IoMT device communicates with the Virtual Machine (VM) allocated for Healthcare As a Service. The Representational State Transfer (REST) API is a popular way of communicating with virtual machines on cloud servers. The REST API has been used in this experiment, which is illustrated in Fig. 3 because of the advantages it provides, including the ability to scale to handle a high volume of requests, the compatibility with a wide range of programming languages, the use of industry-standard HTTP methods and status codes, the lightweight design, the ease with which they can be cached to boost performance further, the advanced security authentication protocols, and the ability to facilitate communication between systems.

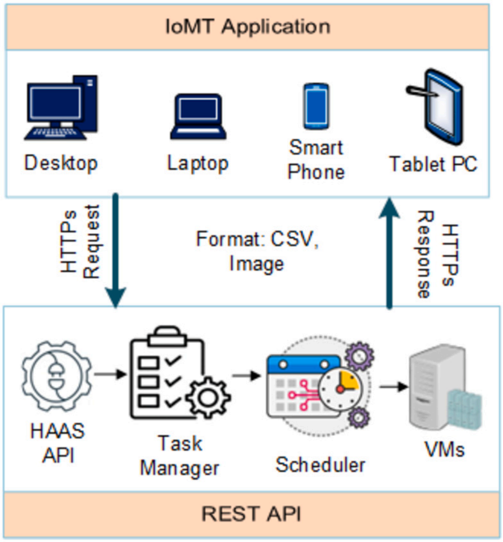

Fig. 3. The REST API communication model for CSV and image data transmission of the proposed HAAS model.

The REST API, according to Fig. 3, passes the request of the HAAS application from the end users. The request passes a Comma Separated Value (CSV) file with a set of CT images for classification and reporting. The data is received, processed, and stored in the appropriate directory by Algorithm 1. It receives the files using the Secure File Transfer Protocol (SFTP).

Algorithm 1 HAAS Application Data Protection.

```
Require: FSTP Request, Virtual Machine Initialization
Ensure: HAAS App Request
Start
$Data \leftarrow$ Decryption(SFTP Packets)
$[Image, CSV, ID] \leftarrow read(Data)$
if $ID == True$ then
    $Directory(ID) \leftarrow write(Image, Path)$
else if $ID! = True$ then
    CreateDirectory(ID)
    $Directory(ID) \leftarrow write(Image, Path)$
end if
$T_{mysql} \leftarrow SQL(CSV)$
if $ID == True$ then
    $T_{mysql} \leftarrow SQL(CSV)$
else if $ID! = True$ then
    CreateUserTable(ID)
    $T_{mysql} \leftarrow SQL(CSV)$
end if
```

The Algorithm 1 distributes the user data received through REST API over the SFTP protocol on the HAAS VM in the appropriate directory and user table. It maintains a unique User Identity (UID) to store the user data in the appropriate directory.

3.2. *Image dataset for HAASNet*

The proposed Healthcare As A Service cloud architecture uses numeric and image data to classify lung cancer from CT images. The numeric data are the lung cancer symptomatic data. The CT images are the data produced by the Computed Tomography (CT) Scanner. The proposed HAAS model uses a Convolutional Neural Network (CNN) specially designed to optimize CT images with minimal cloud resources. However, the literature review demonstrates that the machine learning model cannot perform well if the datasets are rich enough [36]. That is why data processing is essential in any machine learning-based approach [37]. There are two types of data in the proposed experiment. They are numeric and image data. The data processing methodologies utilized in this research are presented in this section.

3.2.1. *Dataset description*

The HAASNet has been trained and tested on two popular CT image datasets. These are the LIDC-IRDI and LUNGx Challenge datasets. This subsection presents the descriptions of these two datasets.

Table 5
Severity of malignancy on a scale from one to five.

Likelihood	Severity
Extremely improbable for cancer	1
Unlikely to get cancer	2
Indeterminate probability	3
Moderately symptomatic of malignancy	4
Extremely predictive of malignancy	5

LIDC-IDRI dataset: The LIDC-IDRI is the most used dataset for computer-aided, automated categorization of lung cancer nodules from CT images. It includes 1018 occurrences. In collaboration, this dataset was created by seven academic institutions and eight medical imaging companies. So, it is a credible dataset for doing research. Four expert radiologists annotate in two phases (blinded and unblinded). During the blindfolded phase, each radiologist detects the nodules independently based on their millimeter measurements. In the unblinded phase, radiologists evaluate the anonymized annotations of other radiologists in addition to their anonymized annotations. The severity of malignancy is rated on a scale from one to five, as shown in Table 5 [38–41].

The data associated with each picture in the LIDC-IDRI collection is saved as an XML file. Based on internal structure, subtlety, calcification, sphericity, margin, lobulation, spiculation, and texture, the radiologists assessed and measured the degree of malignancy. Yet, the poll indicates that radiologists base their conclusions mostly on the nodule's texture and size. The texture annotation grade is further subdivided into three categories. There are solid, semi-solid, and non-solid entities. Each image's recording resolution is 64 by 64 pixels. Each pixel is 0.7 millimeters in size. Each successive scan has a distance of 1.4 millimeters. The real range of each slice's Hounsfield Unit is between -1000 and 400. In this study, however, a standard content between 0 and 1 has been employed [42].

LUNGx challenge dataset: Using Philips Brilliance scanners, the LUNGx challenge data set has been compiled. The database contains 22,489 CT scans in Digital Imaging and Communication in Medicine (DICOM) format. Each photograph is individually recognizable using a Unique Identification (UID). This collection stores anatomy-based sequencing using the DICOM tag (0020, 0013). A single transaxial series scan covering the whole thorax is utilized. Each scan's slice thickness is 1 mm. This dataset was produced for the automated categorization of nodules as malignant or benign. There are ten calibration scans for which CSV files provide nodule location information. None of the remaining scans are labeled or calibrated. Five of these ten scans include cancer nodules, whereas the other five contain benign nodules. This dataset contains a distinct collection of testing scans. The testing scans' nodule locations are supplied in a CSV file with the dataset. This testing scan has sixty scans. These scans contain 73 nodules. Thirteen of these 73 images include two nodules. These 73 scans reveal 37 benign and 36 cancerous lesions [43].

3.2.2. Image data processing

Region of interest (ROI) segmentation: The size of the input layer is $(50 \times 50 \times 1)$. The datasets used in this experiment come with the Cartesian coordinate of the lung cancer nodules. Instead of passing the entire image, this research applies the Region of Interest (ROI)-based segmentation method where the ROI size is (50×50) pixels. The (x_0, y_0) is the top-left corner, and (x_1, y_1) is the bottom-right corner of the ROI, which is defined by equation (6).

$$I_{ROI}(x,y) = I(x+x_0, y+y_0) \quad \text{for } 0 \le x \le 49, 0 \le y \le 49 \tag{6}$$

Colorspace conversion: One of the design criteria of the proposed HAASNet is minimizing the computational complexities. That is why it has been designed to work with grayscale images only. Unlike an RGB image with three channels, a grayscale image has a single channel. The $I_{ROI}(x,y)$ is prepared from equation (6) in RGB colorspace. The $I_{ROI}(x,y)$ has been converted into $I_{ROI}(x,y)$ using equation (7) [44].

$$I_{grayscale}(x,y) = 0.2989 \times I_{ROI}(x,y,R) + 0.5870 \times I_{ROI}(x,y,G) + 0.1140 \times I_{ROI}(x,y,B) \tag{7}$$

In equation (7), $I_{ROI}(x,y,R)$, $I_{ROI}(x,y,G)$, and $I_{ROI}(x,y,B)$ denote the pixel intensities in the red, green, and blue channels of the input RGB image at coordinates (x,y), and $I_{grayscale}(x,y)$ represents the pixel intensity in the converted grayscale image at coordinates (x,y). The coefficients in the formula (0.2989, 0.5870, and 0.1140) are derived from the relative luminance contributions of the red, green, and blue channels based on the human perception of color in the RGB color space [45].

Image enhancement: Contrast-limited Adaptive Histogram Equalization (CLAHE) has been used in this research to enhance the image features [46]. The lung cancer CT images have multiple contrast variations in different areas. That is why applying a global histogram equalizer is not appropriate. The CLAHE approach divides the image into smaller and exclusive regions of size $M \times N$ pixels. Then, a histogram equalizer is calculated for each region, governed by mathematical expression (8).

$$H_i(j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \delta[I(x+k, y+l) - j] \tag{8}$$

Table 6
Number of cases, scans, nodules in the dataset.

Dataset	Cases	Scans	Nodules	Augmented Images
LIDC-IDRI	1018	5000	3875	6480
LUNGx Challenge	73	700	252	1280

Once the local histograms are calculated, the contrast limiting factors are applied to the histogram [47]. It suppresses the over-amplification of noise. A threshold, T, is used to clip the histogram. Any excess counts above T are redistributed uniformly among the other bins. The process is expressed in equation (9).

$$H_i^{\text{clipped}}(j) = \min(H_i(j), T) \tag{9}$$

In the next phase, the Cumulative Distribution Function (CDF) for each clipped histogram is calculated using equation (10). Once the CDF values are obtained, the Bilinear Interpolation (BLI) combines the CDFs of the neighboring $M \times N$ blocks expressed by equation (11). Finally, the original pixel intensities are mapped to the new intensities using equation (12).

$$CDF_i(j) = \frac{1}{MN} \sum_{k=0}^{j} H_i^{\text{clipped}}(k) \tag{10}$$

$$CDF_{\text{final}}(x, y) = \text{BLI}(CDF_i, CDF_j, CDF_k, CDF_l) \tag{11}$$

$$I'(x, y) = L \times CDF_{\text{final}}(x, y)[I(x, y)] \tag{12}$$

In this image enhancement process, $I(x, y)$ represents the pixel intensity in the original image at coordinates (x, y), $I'(x, y)$ denotes the pixel intensity in the enhanced image, $H_i(j)$ is the histogram, $H_i^{\text{clipped}}(j)$ is the contrast-limited histogram, and $CDF_i(j)$ is the cumulative distribution function for the i-th tile. The $CDF_{\text{final}}(x, y)$ is the final CDF obtained by bilinear interpolation. L represents the maximum intensity value in the image and $\delta[\cdot]$ is the Kronecker delta function.

Image augmentation: The image augmentation technique has been used to increase the number of images in the dataset [48]. The size of the dataset has a direct effect on CNN's performance. This experiment uses spatial rotation and flipping to generate the augmented images. The rotation angle $R(\theta)$ is calculated by equation (13). This angle of rotation is applied to the image using equation (14) where (x, y) is the original dimension and (x', y') is the transformed image.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \end{bmatrix} \tag{14}$$

The scaling factor to geometrically scale the images is defined by equation (15). In this equation, s_x and s_y are the scaling factors of x and y axes, respectively. These factors are applied to the image using equation (16).

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = S(s_x, s_y) \begin{bmatrix} x \\ y \end{bmatrix} \tag{16}$$

Both horizontal and vertical flipping have been applied in the research to generate augmented images. The vertical and horizontal flipping are defined by equation (17) and (18), respectively.

$$Horizontal flip: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -x \\ y \end{bmatrix} \tag{17}$$

$$Vertical flip: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ -y \end{bmatrix} \tag{18}$$

3.2.3. Dataset splitting

The LIDC-IDRI dataset contains 1018 cases (see Table 6). From these cases, 5000 images have been selected, which include 3875 nodules. After performing image augmentation, the number of images becomes 6480. The LUNGx Challenge dataset contains 73 cases. Five hundred scans have been selected from these cases, including 252 nodules. After augmentation, the number of images becomes 1280. The LUNGx challenge dataset has not been used for training. It has been used to cross-validate the performance of the proposed HAASNet trained on the LIDC-IDRI dataset. The augmented LIDC-IDRI dataset has been split into training, testing, and validation sets by maintaining a $70 : 15 : 15$ ratio. At this ratio, there are 4536 training images, 972 testing images, and 972 validation images.

Table 7
Input parameter comparison among CNNs.

Network	Width	Height	Channels	Parameters
EfficientNet	224	224	3	150528
MobileNet	224	224	3	150528
DenseNet	224	224	3	150528
ResNet	224	224	3	150528
GoogLeNet	224	224	3	150528
VGG	224	224	3	150528
AlexNet	227	227	3	154587
Proposed	50	50	1	2500

3.3. Network design and learning algorithm

After that, the HAASNet was developed through mathematical analysis of the CNN background. Instead of trying different models common in most of the research, we architected the network architecture through mathematical interpretation. However, the optimization algorithm has been chosen based on their performance through experiment [49]. Analyzing the underlying mathematical models of a CNN and modifying them is an effective way to enhance performance while reducing computational resource consumption [50]. One of the core contributions of this research is the simplicity of the network design and the reduction of the computational cost. This subsection presents the mathematical interpretations of various layers and elements of the proposed HAASNet. It also presents the exploratory analysis of the learning algorithms.

3.3.1. HAASNet architecture

The literature review presented in section 2 shows many well-architected and optimized Convolutional Neural Networks (CNN). These networks classify the lung cancer nodule with acceptable accuracy. However, most CNNs are designed and optimized to improve classification performance, ignoring the architectural complexities and computation costs. As a result, they are not suitable for cloud-based healthcare service applications. The proposed HAASNet has been developed for the Healthcare-As-A-Service model as the lung cancer nodule classifier. The HAASNet architecture maintains simplicity, ensures lower computational cost, and exhibits acceptable classification performance.

Input layer: The input layer size of the proposed HAASNet is $(50 \times 50 \times 1)$. Popular CNNs with good classification performance, including EfficientNet, MobileNet, DenseNet, ResNet, GoogLeNet, and VGG, use $(224 \times 224 \times 3)$ input layer [51]. Another popular CNN, AlexNet, uses $(227 \times 227 \times 3)$ input layer [52]. The number of input parameters for these networks is measured by equation (19) where w, h, and c represent width, height, and number of channels, respectively.

$$P = w \times h \times c \tag{19}$$

The number of input parameters of the popular CNNs has been calculated using (19) and listed in Table 7 for comparison. According to the data in Table 7 and mathematical comparison expressed by equation (20), the proposed HAASNet uses 60.44 fewer input parameters than comparing networks. It significantly reduces computational costs.

$$C = \frac{\frac{1}{n}\sum_{i=1}^{n} w_i \times h_i \times c_i}{2500} \tag{20}$$

In equation (20), c is a numeric value that represents the comparison, n is the number of comparing networks, $w_i, h_i,$and$, c_i$ are the width, height and several channels of the i^{th} network. Because of taking 60.44 times less input parameter, the proposed network works efficiently with a simpler architecture and consumes fewer computational resources. That is why the HAASNet is an ideal CNN for cloud applications.

Convolutional layers: The proposed HAASNet consists of multiple 2D convolutional layers (Conv2D). Three different sizes of filters of 32, 64, and 128 pixels have been used. However, we used the same 3×3 kernel size for every filter to minimize computational cost. We have an input of size $w \times h \times c_{in}$, where w is the width, h is the height, and c_{in} is the number of input channels. We apply a convolutional layer with a kernel size of $k \times k$, and the layer has C_{out} output channels (filters). The number of parameters (weights) in this layer is given by equation (21).

$$n_{\text{params}} = k \times k \times w \times h \times c_{\text{in}} \times c_{\text{out}} \tag{21}$$

When we use the same kernel size throughout the network, the number of parameters in each convolutional layer becomes more predictable and manageable. This can reduce the overall number of parameters in the network, reducing the computational cost during training and inference. In contrast, if we use different kernel sizes in different layers, we would need to consider the varying number of parameters in each layer, which can increase the complexity of the model and make it more difficult to optimize the overall architecture for reduced computational cost.

Weight initialization: Weight initialization impacts CNN training and model convergence [53]. Poor weight initialization can cause vanishing gradients, which make learning difficult, stall convergence, and trap the model in a suboptimal local minimum [54]. Exploding gradients come from enormous initial weights, creating training instability, oscillation, or divergence and making optimal solution convergence difficult [55]. Poor weight initialization slows convergence, requiring more training epochs and processing resources. Symmetry breaking, when initializing weights with the same values, causes neurons to learn the same features and gradients and wastes the network's capacity [56]. Appropriate weight initialization disrupts this symmetry, letting neurons learn diverse aspects. Finally, improper weight initialization can keep the model in local minima instead of the global minimum [57].

This research explores five weight initialization methods. These methods are Zero Initialization defined by equation (22), Random Initialization expressed in equation (23), Xavier/Glorot Initialization governed by equation (24), He Initialization defined by equation (25), and LeCun Initialization expressed by equation (26) [58].

$$W_{ij} = 0 \tag{22}$$

$$W_{ij} = U(-a, a) \tag{23}$$

$$W_{ij} = U\left(-\sqrt{\frac{6}{n_{in} + n_{out}}}, \sqrt{\frac{6}{n_{in} + n_{out}}}\right) \tag{24}$$

$$W_{ij} = U\left(-\sqrt{\frac{6}{n_{in}}}, \sqrt{\frac{6}{n_{in}}}\right) \tag{25}$$

$$W_{ij} = U\left(-\sqrt{\frac{3}{n_{in}}}, \sqrt{\frac{3}{n_{in}}}\right) \tag{26}$$

Here, W_{ij} represents the weight between neurons i and j, $U(a, b)$ represents the uniform distribution between a and b, n_{in} represents the number of input units, and n_{out} represents the number of output units.

Activation function: The proposed HAASNet uses the Rectified Linear Unit (ReLU) as the activation function of the hidden nodes. It is a non-linear activation function defined by equation (27) [59].

$$\mathbf{y}_i = \text{ReLU}\left(\text{BN}\left(\text{Conv}_{k\times k}\left(\mathbf{x}_{i-1}, W_i\right)\right)\right) \tag{27}$$

This equation (27) represents a common operation in a convolutional neural network (CNN) layer. The output feature map at the i-th layer is denoted by $\mathbf{y}i$. The Rectified Linear Unit (ReLU) activation function is represented by $\text{ReLU}(\cdot)$ and is defined as $\text{ReLU}(x) = \max(0, x)$, introducing non-linearity into the network. Batch Normalization (BN) is denoted by $\text{BN}(\cdot)$, a technique used to normalize the input feature map before applying the activation function, helping to improve the training process and generalization of the model. The 2D convolution operation with a kernel size of $k \times k$ is represented by $\text{Conv}k \times k(\cdot)$, responsible for learning spatial features from the input data. The input feature map at the $(i-1)$-th layer is represented by $\mathbf{x}i - 1$, and the weight matrix (convolution kernel) for the i-th layer is denoted by W_i.

Pooling layers: The proposed HAASNet architecture uses Max Pooling after each group of convolutional layers. A 2×2 size window has been used to perform the max pooling identified as MaxPooling2D, mathematically defined by equation (28).

$$\mathbf{y}_p = \text{MaxPool}_{2\times 2}\left(\mathbf{x}_p\right) \tag{28}$$

In equation (28), the $\mathbf{y}_p$ is the output of the pooling layer, $\mathbf{x}_p$ is the input to the pooling layer, and MaxPool is the max-pooling operation.

Dropout layers: The initial design of the proposed HAASNet suffers from the overfitting problem, in which the linear distance between training and validation error had been discovered. It has been discussed in detail in section 4. To prevent the network from overfitting, dropout layers were developed using equation (29).

$$\mathbf{y}_d = \text{Dropout}_r\left(\mathbf{x}_d\right) \tag{29}$$

In equation (29), the $\mathbf{y}_d$ is the output of the dropout layer, $\mathbf{x}_d$ is the input to the dropout layer, and r is the dropout rate. The dropout rate in each layer has been increased by 0.1, demonstrating effective prevention of the overfitting problem.

Dense layers: The signals from the max-pooling layer are further processed through two dense layers with 128 neurons. These neurons use the ReLU as their activation functions. The process is governed by equation (30).

$$\mathbf{y}_j = \text{ReLU}\left(\text{BN}\left(\text{Dense}\left(\mathbf{x}_{j-1}, W_j\right)\right)\right) \tag{30}$$

In equation (30), the $\mathbf{y}_j$ is the output of the j-th dense layer, $\mathbf{x}_{j-1}$ is the input to the j-th layer, W_j are the weights of the j-th layer, Dense is the fully connected layer operation, and BN is batch normalization.

3.3.2. Learning algorithm

The learning method substantially affects the performance of a Convolutional Neural Network (CNN), affecting factors such as convergence rate, model generalization, stability, noise tolerance, computational efficiency, and hyperparameter tuning [60]. The problem statement, dataset size, computational resources, and desired performance indicators must all be considered when selecting an appropriate learning algorithm. Optimal learning techniques can result in faster convergence, enhanced generalization, and enhanced performance overall. The HAASNet experiment analyzes the effect of four learning algorithms. They are Nesterov Accelerated Gradient (NAG) [61], Adaptive Gradient (AdaGrad) [62], Root Mean Square Propagation (RMSProp) [63], and Adaptive Moment Estimation (ADAM) algorithm [64].

Nesterov accelerated gradient (NAG): The proposed HAASNet has experimented with the NAG learning algorithm to improve the standard momentum by calculating the gradient at a look-ahead point. The lookahead point is calculated using equation (31). After getting the lookahead points, the gradient of the cost function concerning the lookahead weights has been measured using equation (32). The velocity vector is updated using the lookahead gradient and the momentum term defined as equation (33). Finally, the weights are updated using the updated velocity vector expressed in equation (34)

$$w_i^{\text{lookahead}} = w_i^t - \gamma v_i^t \tag{31}$$

$$\nabla_{w_i^{\text{lookahead}}} J(w^{\text{lookahead}}) \tag{32}$$

$$v_i^{t+1} = \gamma v_i^t + \eta \nabla_{w_i^{\text{lookahead}}} J(w^{\text{lookahead}}) \tag{33}$$

$$w_i^{t+1} = w_i^t - v_i^{t+1} \tag{34}$$

Adaptive gradient (AdaGrad): The AdaGrad learning algorithm has been applied in the proposed HAASNet to explore the learning performance. It adjusts the learning rate for each weight based on the previous states' data. The accumulation of the sum of the squared gradient of each weight is measured using (35). Based on this sum, the weights are updated using the adaptive learning rate, which is expressed by equation (36).

$$G_i^{t+1} = G_i^t + (\nabla_{w_i} J(w))^2 \tag{35}$$

$$w_i^{t+1} = w_i^t - \frac{\eta}{\sqrt{G_i^{t+1} + \epsilon}} \nabla_{w_i} J(w) \tag{36}$$

Here in equations (35) and (36), w_i^t is the weight, G_i^t represents the sum of squared gradients, $\nabla_{w_i} J(w)$ is the gradient of the cost, η is the global learning rate, ϵ is a small constant to prevent division by zero, and w_i^{t+1} is the updated weight of the i-th parameter at time step $t+1$.

Root mean square propagation (RMSProp): The proposed HAASNet has been studied with the effect of the RMSProp learning algorithm. It uses an exponential moving average for squared gradients calculated using the equation (37). The weights are updated using the adaptive learning rate, which is governed by equation (38)

$$E[g^2]_i^{t+1} = \rho E[g^2]_i^t + (1-\rho)(\nabla_{w_i} J(w))^2 \tag{37}$$

$$w_i^{t+1} = w_i^t - \frac{\eta}{\sqrt{E[g^2]_i^{t+1} + \epsilon}} \nabla_{w_i} J(w) \tag{38}$$

Here in equations (37) and (38), w_i^t is the weight, $E[g^2]_i^t$ is the exponential moving average, $\nabla_{w_i} J(w)$ is the gradient of the cost function, η is the global learning rate ϵ is a small constant to prevent division by zero, ρ is the decay rate which is a hyperparameter between 0 and 1 that controls the weight of the exponential moving average, and w_i^{t+1} is the updated weight of the i-th parameter at time step $t+1$.

Adaptive moment estimation (ADAM): Adaptive Moment Estimation (ADAM) is the fourth and last algorithm that this experiment explores to identify the effect on the performance of the proposed HAASNet. The ADAM combines the concept of momentum and adaptive learning rates. In this approach, the first and second moments, called the mean and uncentered variance, are calculated using equations (39) and (40), respectively.

$$m_i^{t+1} = \beta_1 m_i^t + (1-\beta_1)\nabla_{w_i} J(w) \tag{39}$$

$$m_i^{t+1} = \beta_1 m_i^t + (1-\beta_1)\nabla_{w_i} J(w) \tag{40}$$

Correcting the biases in the first and second moments is essential to get the optimal result from the ADAM optimizer. It has been done using equations (41) and (42).

$$\hat{m}_i^{t+1} = \frac{m_i^{t+1}}{1-\beta_1^{t+1}} \tag{41}$$

$$\hat{\upsilon}_i^{t+1} = \frac{\upsilon_i^{t+1}}{1-\beta_2^{t+1}} \tag{42}$$

Finally, the weights are updated using equation (43) using the adaptive learning rate and bias-corrected moments. The equation (43) combines both adaptive and momentum-based approaches.

$$w_i^{t+1} = w_i^t - \frac{\eta}{\sqrt{\hat{\upsilon}_i^{t+1}} + \epsilon}\hat{m}_i^{t+1} \tag{43}$$

In the ADAM learning algorithm development, defined by equations (39), (40), (41), (42), and (43), w_i^t represents the weight, $\nabla_{w_i} J(w)$ is the gradient of the cost function, m_i^t is the first moment, υ_i^t is the second moment, β_1 and β_2 are the exponential decay rates, η is the global learning rate, ϵ is a small constant, $\hat{m}i^{t+1}$ and $\hat{\upsilon}i^{t+1}$ are the bias-corrected first and second moments, respectively.

3.4. HAAS model architecture

Designing an architectural model to make the proposed HAAS cloud server compatible is an essential phase of this research. The HAAS model architecture has been developed using the cloud computing service model. HAAS is the abbreviation for Healthcare-As-A-Service inspired by three fundamental cloud service models. The concept of the HAAS model is borrowed from Software-As-A-Service architecture. However, it has been remodeled, and novelty has been introduced to make it compatible with the requirements we established at the beginning of the research through literature review. The proposed Healthcare-As-A-Service (HAAS) model architecture, illustrated in Fig. 4, is divided into three layers. The top layer is the presentation layer. This layer generates a User Interface (UI) for administrative and general users. The general users are the patients and the physicians. This layer depends on the application layer to function.

The application layer is hosted in a Virtual Machine (VM) that houses the HAASNet, Image Processing Application (IPA), and IoMT Data Application (IDA). These three applications, along with the configuration server, are grouped. The Servlet Loader (SVL) initiates these applications when the VM is initiated according to the configuration specified by the administrator. The Workflow Controller (WCR) maintains the communication among HAASNet, IPA, IDA, UIs, and HAAS Database Management System (HDMS). A UI Generator (UIG) module generates separate UI for the patients and physicians. The responses from the users are controlled using the Access Controller (ACR). The ACR directly communicates with the HDMS.

The HDMS is responsible for inserting, retrieving, and updating data in the HAAS database. A virtual database server manages this database. The HAAS database consists of three types of databases. They are the log database, metadata database, and app database. The log data keeps track of every instance on the HAAS server. The Metadata database contains all descriptions related to the HAAS database. The data relating to the process, user interface, the structure of the schemas, and business logic are part of the metadata database. Different types of metadata tables communicate with each other using Meta Controller (MTC). It has been observed that the data retrieval delay impacts the overall service quality of the proposed system. This weakness has been overcome using Workflow Chache (WFC) and Application Cache (APC) memories.

The proposed methodology depends on mathematical interpretation-based decision-making strategies while designing the network, training it, and processing the dataset. Furthermore, the classification process is guided by physio-symptomatic data from the IoMT devices. In addition, the HAAS system architecture has been developed according to the cloud server-compatible design. Combining everything, the proposed system exhibits good performance.

4. Experimental results and evaluation

The rigorous performance analysis through experimental results analysis and evaluation from different contexts is one of the strengths of this paper. The performance evaluation criteria have been discussed in the first subsection of this section. After that, the performance analysis results of IoMT, Cloud Computing, and HAASNet are presented.

4.1. Performance evaluation criteria

The proposed Healthcare As A Service (HAAS) model combines IoMT, Cloud Computing, and CNN. That is why it requires performance evaluation from three different perspectives. And the evaluation criteria in each perspective are different. This subsection discusses the performance evaluation criteria used in this experiment for IoMT, Cloud Computing, and Deep Learning. These criteria are listed in Table 8.

IoMT and Cloud Computing evaluation metrics are obtained from the system log. The CNN performance evaluation metrics are calculated using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values from the confusion matrix. The accuracy, precision, recall (sensitivity), and F1 Score are defined by equations (44), (45), (46), and (47), respectively.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{44}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{45}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{46}$$

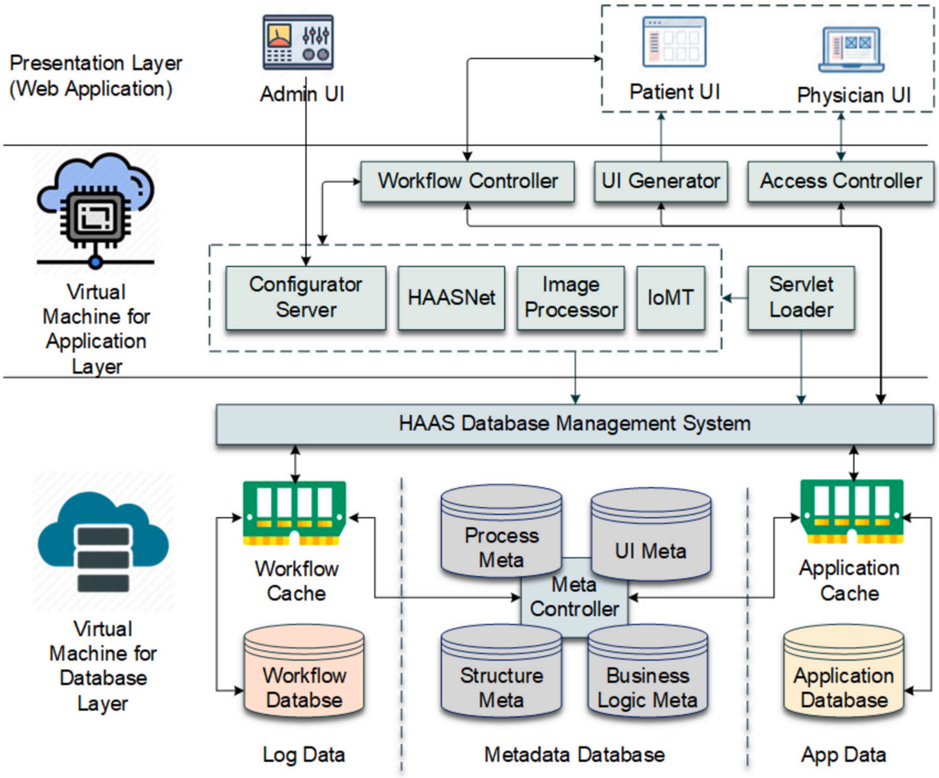

Fig. 4. The overview of the HAAS architecture with essential components.

Table 8
Performance evaluation criteria of the proposed HAAS.

IoMT	Cloud Computing	CNN
Sensor data processing delay	Server response time	Accuracy
Application data processing delay	Server throughput	Precision
Optical data processing delay	Scalability of the service	Recall
Transmission delay	Resource utilization	F1 Score
Server response delay	Server latency	ROC Curve

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{47}$$

4.2. IoMT performance analysis

The proposed diagnosis system depends on the IoMT data to provide service to the users. The IoMT communicates with the HAAS model and transmits sensor, application, and optical data. The experimenting system uses this data to diagnose the current status of lung cancer. This section presents the performance analysis of the experimenting IoMT system by examining encryption and processing delays for different types of data, including sensor, application, and optical data, along with transmission and server response time. The statistical data of the performance analysis are listed in Table 9.

The analysis has been done by observing encryption and processing delays for sensor, application, and optical data across ten instances. The data also includes each instance's transmission and server response times. The optical data have the highest encryption and processing delays, ranging from 18 to 35 seconds, followed by sensor data (1 to 4 seconds) and application data (2 to 5 seconds). Transmission times varied across instances, with the shortest being 2 seconds and the longest being 3.75 seconds. The server response times ranged from 4.25 to 7.04 seconds. The performance of the proposed IoMT for different parameters listed in Table 9 has been illustrated in Fig. 5.

The analysis, illustrated in Fig. 5, reveals that optical data requires the most processing time, possibly due to the large size and complexity. On the other hand, sensor and application data have lower encryption and processing delays, likely because they consist of smaller data packets. The variability in transmission times could be attributed to network congestion or fluctuations in the bandwidth available for data transmission. Server response times are crucial for providing timely feedback to healthcare professionals, and the observed range indicates that there could be room for improvement in optimizing server processing times.

Table 9
IoMT Performance Analysis.

	Encryption & Processing Delay			Delay in Seconds	
Instance	Sensor	App	Optical	Transmission	Server
1	4	2	25	2.42	6.89
2	2	3	20	2.00	6.75
3	4	5	28	2.25	6.50
4	3	3	21	3.33	5.21
5	3	5	28	3.00	4.25
6	2	5	18	3.00	4.25
7	3	4	29	3.00	6.89
8	3	3	24	2.92	6.75
9	1	3	28	3.75	7.04
10	3	3	35	2.92	6.82

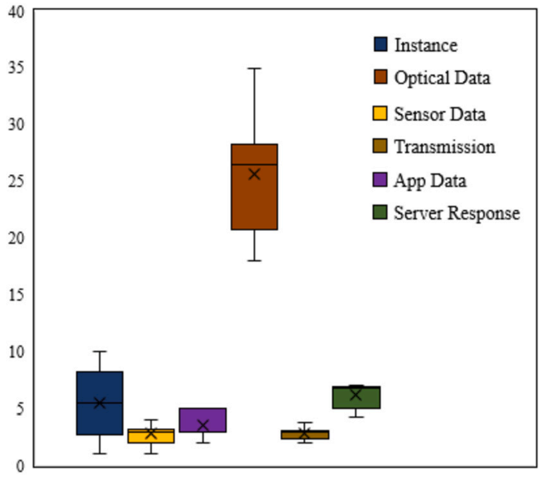

Fig. 5. IoMT performance analysis and comparison for different parameters.

Table 10
Cloud Performance Analysis.

Instance	Response Time	Throughput	Latency
1	17	4	22
2	24	3	27
3	19	3	22
4	25	2	30
5	17	4	21
6	18	3	23
7	24	3	28
8	18	3	23
9	21	3	24
10	16	4	20

4.3. Cloud computing performance analysis

The proposed HAAS is a subset of the cloud Software-As-A-Service (SAAS) model architecture. That is why the performance of the cloud section has been analyzed using the common and relevant SAAS evaluation metrics, which are listed in Table 8. The numerical evidence of the performance has been listed in Table 10. The standard deviations of response time, throughput, and latency are 3.31 seconds, 3.2 requests per unit time, and 3.16 seconds, respectively. Therefore, the performance of the experimenting cloud model is consistent for the experimental instances.

Response time, throughput & latency The Response Time, Throughput, and Latency metrics illustrated in Fig. 6 provide insights into the proposed system's efficiency. Instances 2, 4, and 7 have longer response times, while instances 1, 5, and 10 have the shortest response times and the highest throughput. The latency values show that instances 4 and 7 take the longest time to transmit data, whereas instance 10 has the lowest latency. Overall, the performance visualization demonstrates the cloud computing model's effectiveness and consistency.

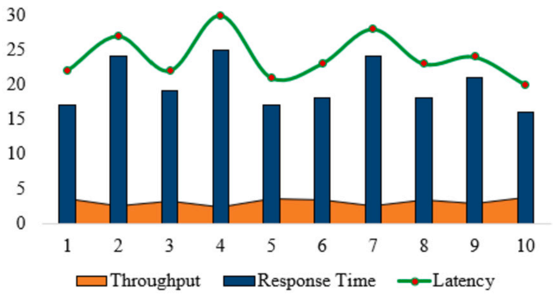

Fig. 6. The performance visualization of the proposed cloud model in terms of throughput, response time, and latency.

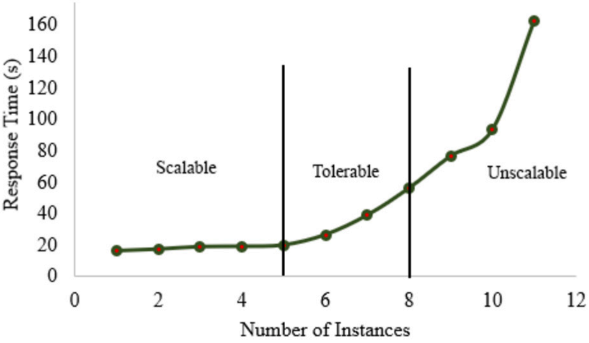

Fig. 7. Scalability analysis of the proposed system to discover scalable, tolerable, and unscalable zone.

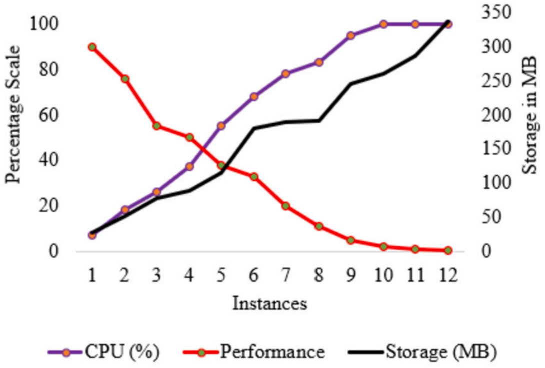

Fig. 8. CPU usage, storage demand, and performance for 12 instances.

Scalability analysis: Scalability is a crucial factor in assessing the performance of a cloud computing model. While Table 10 does not feature this metric due to varied measurement methods, the Average Response Time (ART) of the suggested HAAS model stands at 19.9 seconds. This ART serves as the benchmark for performance assessment in terms of scalability. The scalability analysis depicted in Fig. 7 demonstrates that the proposed system can efficiently handle up to five concurrent requests. It can still manage up to eight requests with acceptable performance degradation, but response times spike significantly when there are more than eight concurrent requests. Thus, the system is optimized for an average of five simultaneous requests.

Resource utilization: Scalability is intrinsically linked to the utilization of cloud resources. As the number of instances in the proposed HAAS model rises, there is a concurrent uptick in resource consumption. System performance tends to falter when resource utilization approaches its limit. Experimental data showcasing the relationship between CPU usage, storage demand, and performance across 12 instances can be seen in Fig. 8. With an increment in instances from 1 to 12, CPU usage witnesses a steep ascent from 7% to a full 100%. This surge in CPU utilization is inversely proportional to system performance, which commences at 90 for a single instance and plummets to 0.5 for 12 instances. Concurrently, storage consumption also rises, initiating at 28 MB for one instance and surging to 336 MB for twelve. This diminishing performance as the number of instances climbs is attributed to the escalating demand on CPU and storage - culminating in heightened competition among instances and an overloaded system. The inference drawn from this analysis is that as instances multiply, the system inches closer to saturation, impeding performance.

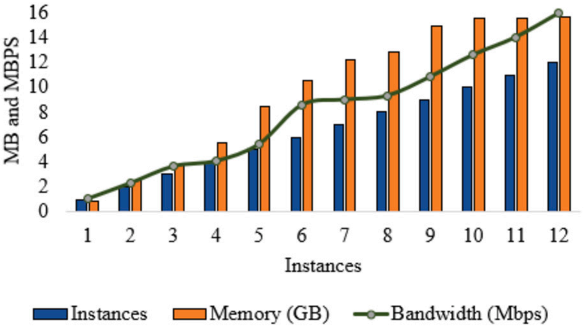

Fig. 9. Memory consumption and bandwidth requirement.

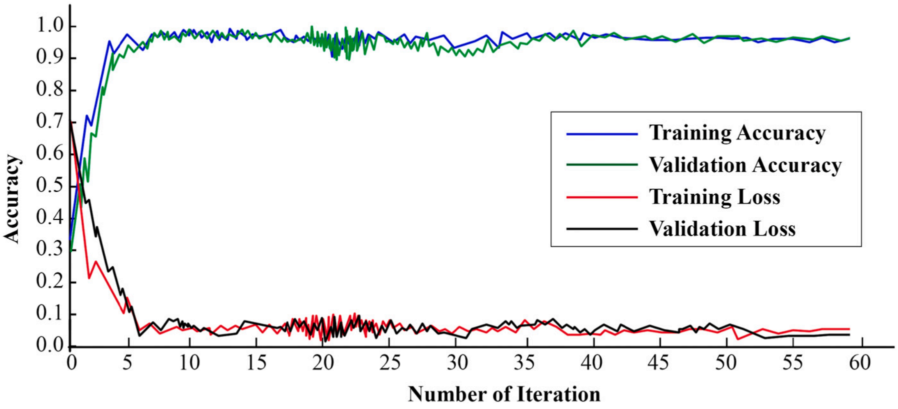

Fig. 10. The learning progress during the training phase with number of iterations and accuracy.

Fig. 9 shows the relationship between the number of instances, memory usage in gigabytes (GB), bandwidth consumption in megabits per second (Mbps), and memory consumption as a percentage. As the number of instances rises from 1 to 12, we can observe a noticeable increase in memory usage, going from 0.80 GB to 15.70 GB. This increase in memory usage is accompanied by considerable growth in memory consumption percentage, starting at 5.01% for one instance and reaching 98.10% for 12 instances. Bandwidth consumption also demonstrates an upward trend, with 1.01 Mbps for a single instance and 15.97 Mbps for 12 instances. The data indicates that the system's memory and bandwidth consumption increase as more instances are added, resulting in a higher memory consumption percentage. This higher memory consumption percentage suggests that the system's resources are being heavily utilized, potentially leading to performance issues if it approaches its maximum capacity.

4.4. HAASNet performance analysis

The HAASNet plays the most significant role in the proposed Healthcare-As-Service (HAAS) model. The overall performance of the diagnosis system depends on the accurate prediction from the HAASNet. That is why its performance has been evaluated from different contexts. This subsection presents the performance analysis of the proposed HAASNet.

4.4.1. Overall performance

The proposed HAASNet has been trained with the LIDC-IDRI dataset. Instead of training the network with all 4536 training images, the mini-batch method with a batch size of 227 has been used in this experiment. Every batch is randomly shuffled before training. With the mini-batch size 227, the network gains a validation accuracy of 96.01% after 57 epochs. The learning progress has been illustrated in Fig. 10.

Fig. 10 demonstrates a rapid increase in training accuracy till 10^{th} iterations. During this iteration, the training loss falls rapidly as well. After 10^{th} iteration, the network learns smoothly till 45^{th} epochs. After that, the learning curve flattens, indicating that the network's accuracy is no longer increasing. However, this experiment continues till 57^{th} epochs. After that, it is terminated. The training and validation accuracy curves don't show any significant difference throughout the learning progress. A similar statement is valid for training and validation loss curves. It indicates that the network is not overfitting. The experimenting HAASNet takes 127 minutes and 18 seconds to complete 57^{th} epochs by reaching 96.01% validation accuracy.

Confusion matrix analysis: After training, the HAASNet was tested with the testing dataset. There are 972 images in the training dataset. Five hundred images from these 972 images have been randomly selected for testing purposes. The performance on the testing dataset has been evaluated using the confusion matrix illustrated in Fig. 11.

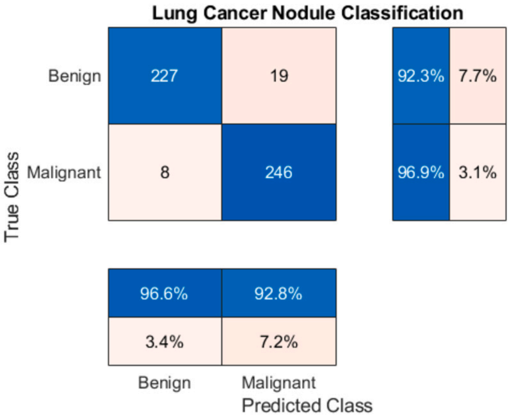

Fig. 11. Confusion Matrix Analysis on Test Dataset.

Table 11
Confusion matrix analysis on the testing dataset.

Class	Precision	Recall	F1 Score
Benign	96.66	92.3	93.98
Malignant	92.83	96.9	94.8

Table 12
k-fold cross validation on LUNGx Challenge dataset.

k	Accuracy	Precision	Recall	F1 Score
1	95.92	96.65	93.79	94.2
2	96.05	96.14	96.8	94.78
3	96.17	95.99	95.71	95.74
4	95.85	96.78	94.97	94.67
5	96.36	96.8	95.66	94.65
Average	96.07	96.47	95.39	94.81

In the confusion matrix of Fig. 11, true positive, true negative, false positive, and false negative are 227, 246, 19, and 8, respectively. The overall accuracy of the proposed HAASNet on the testing dataset is 94.60% according to equation (44). Table 11 summarizes the performance of the testing dataset.

K-fold cross validation: The accuracy of the testing dataset is 94.60%, which is impressive. However, without further validation, the generalizability of the proposed methodology is not properly justified. That is why the performance of the HAASNet has been further evaluated through extensive experiments. The performance of the proposed HAASNet was cross-validated using the LUNGx challenge dataset. This dataset comprises 252 original images and 1,028 augmented ones. From this combined set of images, 300 were randomly chosen to assess the performance of HAASNet through k-fold cross-validation with $k = 5$. The results can be found in Table 12.

The HAASNet model demonstrates consistent and high performance across all five folds in Fig. 12. The average accuracy is 96.07%, indicating that the model can correctly classify lung cancer cases in most instances. Precision, which measures the proportion of true positive cases among all predicted positive cases, averages 96.47%, suggesting that the model is highly reliable in identifying lung cancer cases. The average recall, or sensitivity, is 95.39%, reflecting the model's ability to detect true positive cases among all actual positive cases. Lastly, the F1 Score, which is the harmonic mean of precision and recall, averages 94.81%, indicating a well-balanced performance between precision and recall.

ROC curve analysis: The performance of the proposed HAASNet has been further analyzed using the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) [65], which is illustrated in Fig. 13. It represents the ability of the proposed HAASNet to discriminate malignant and benign classes. Fig. 13 shows the AUC-ROC for each k of k-fold cross-validation. The curves of each k overlap each other. As a result, variations in the average area are insignificant. It indicates the consistent and acceptable performance of the HAASNet in classifying lung cancer nodules in malignant and benign classes. The average AUC is 0.958. That means the proposed HAASNet classifies with 95.82% sensitivity.

Performance comparison The performance of the proposed HAASNet has been compared with other similar approaches, which have been listed in Table 13. S. K. Lakshmanaprabu et al. employed the Optimal Deep Neural Network (ODNN) [66] and achieved an

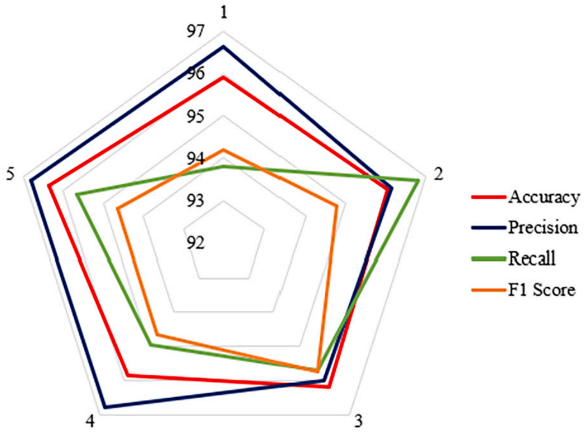

Fig. 12. Performance visualization on k-fold cross validation at $k = 5$.

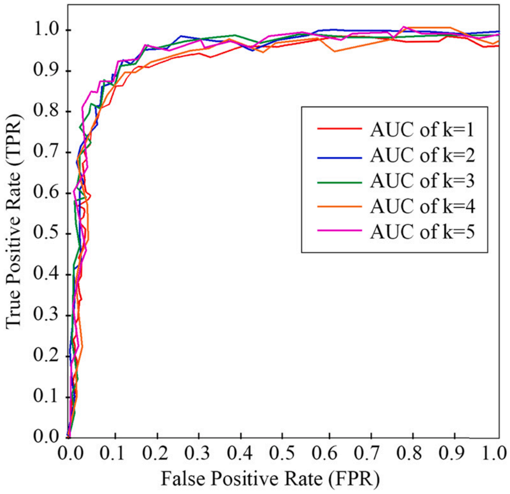

Fig. 13. Performance visualization on k-fold cross validation at $k = 5$.

Table 13
Performance comparison with other similar approaches.

Author	Method	Accuracy
S. K. Lakshmanaprabu et al. [66]	Optimal Deep Neural Network (ODNN)	94.56%
C. Liu et al. [67]	YOLOv3	95.40%
T. L. Chaunzwa et al. [68]	VGG-16	68.60%
Y Onozato et al. [69]	Multiple Models	80.40%
T. I. A. Mohamed et al. [70]	EOSA Metaheuristic Algorithm	93.21%
Proposed	**HAASNet**	**96.47%**

accuracy of 94.56%. Meanwhile, C. Liu et al. used the YOLOv3 technique [67], which slightly outperformed the former with an accuracy of 95.40%. On the other hand, T. L. Chaunzwa et al. utilized the VGG-16 approach [68], resulting in a considerably lower accuracy of 68.60%. Y. Onozato et al. integrated multiple models in their research [69], achieving an accuracy rate of 80.40%. T. I. A. Mohamed et al. applied the EOSA Metaheuristic Algorithm [70] and managed an impressive 93.21% accuracy. Notably, our proposed method, HAASNet, surpassed all the aforementioned techniques, recording the highest accuracy of 96.47%.

4.4.2. Network depth VS. performance

Optimization is an essential phase of developing practical solutions [71]. It helps get optimal operational strategies and discover the best-performing configurations [72]. The proposed HAASNet is a Convolutional Neural Network (CNN) with ten convolutional layers, two dense layers, and a final classification layer. Therefore, the network is a 13-layer-deep CNN. This optimized network depth minimizes the training and validation error, assuring consistent and reliable performance. Fig. 14 illustrates the training and validation error concerning the network layer depth. The training error keeps reducing when the network depth increases. However,

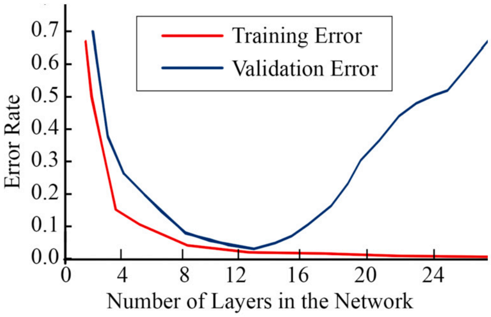

Fig. 14. Network depth optimization for better performance.

after 13^{th} layer, the validation error starts growing. The linear distance between training and validation error curves is lowest at 13^{th} layer, and after that, it starts increasing. That means the network overfits if it is more than 13 layers deep. Reducing the depth below the 13-layer would cause an underfitting effect. From the learning curve analysis presented in Fig. 14, it is evident that the network architecture of the proposed HAASNet is optimized for consistent and reliable performance.

5. Ethical implication

The proposed HAASNet has been developed and implemented to provide global accessibility to lung cancer diagnosis regardless of geographical boundaries. The underdeveloped parts of the world where proper diagnostic services are rare have been considered the primary beneficiaries of this system. However, it leads to some ethical implications discussed in this section.

5.1. Healthcare disparities

Even though the HAASNet has been developed to enable ubiquitous access to lung cancer diagnosis services, there are some barriers to accessing this service, which causes healthcare disparities. The IoMT proposed in this system requires marginal knowledge of operating IoMT devices. It also requires the knowledge of basic computing. That means even if it sheds light on eradicating the geographical boundaries for lung cancer diagnosis. It introduces new technical boundaries.

5.2. Resource allocation

The Healthcare-As-A-Service (HAAS) is accessed through IoMT, which requires electricity, internet connectivity, and IoMT devices. The unavailability of any of these resources will cause a service interruption. This paper has not explored the direction to distribute the IoMT devices unbiasedly. Without proper governance and fair distribution of resources, the HAAS model's potential cannot be fully unitized. However, fair resource allocation depends on the ethical status of the concerned authority. It is beyond the scope of the HAAS model.

The HAAS potentially reduces the lung cancer diagnosis cost and increases its availability to the mass population. However, it creates a digital discrimination. Even if the service is available, people may fail to use it because they lack digital service literacy. As a result, it may deprive a part of the target population of availing the service.

6. Limitations and future scope

The proposed HAAS is a subset of the SAAS cloud service architecture. It is a unique combination of CNN, IoMT, and Cloud Computing. This innovative approach demonstrates the potential of lowering the mortality rate of lung cancer by facilitating early diagnosis opportunities. It reduces the costs associated with the diagnosis and removes the geographical barrier to accessing lung cancer diagnosis services. However, it naturally inherits the limitations of SAAS [73], CNN [74], and IoMT [75]. Besides, there are some additional weaknesses yet to be addressed in this research. This section highlights the limitations and weaknesses of the proposed Healthcare-As-A-Service model. There are scopes of conducting more research to overcome the limitations and strengthen the weaknesses. That is why these shortcomings are considered the future scope of this research.

No usability and user experience assessment: The proposed Healthcare-As-A-Service (HAAS) model is still in the laboratory in the research and development phase. No graphical user interface has been developed yet. A command line-based interface has been developed to experiment with the model and collect data to analyze its performance. As a result, the usability and user experience assessment have not been included in this paper. However, the initiative to develop the GUI and launch the HAAS model as a full-fledged service has been taken. The usability and user experience-related analysis will be published in the subsequent paper.

Absence of compact IoMT: The proposed system has experimented with multiple sensors IoMT sensors. No compact version of IoMT has been designed. It is essential to study the usability of the proposed IoMT from the user end and prepare a compact design that includes all sensors in one device. However, it is beyond the scope of the current research to aim to commercialize the current condition of the proposed IoMT. It creates another research opportunity to study the findings from the end user's perspective, design the IoMT keeping commercial features in mind, and analyze the business prospects of the proposed HAAS.

Limited scalability: The proposed HAAS model is scalable for up to 5 diagnosis requests at a time. The service quality degrades after more than five requests at a time. However, it can still tolerate the computational load and complete the request process within an acceptable time frame for up to 8 requests. The experimental data show that this is the maximum number of requests the HAAS can handle simultaneously. This limitation portrays the scope of further research to optimize the proposed HAAS and increase the scalability.

Centralized server: Data center replication at different geographic locations improves the quality of cloud computing services [76]. However, it has not been addressed in the proposed study. It is beyond the scope of this research to replicate the HAAS model in multiple servers and analyze the performance. However, it is another research opportunity to improve the service quality of the HAAS model.

AML attack: There is no Adversarial Machine Learning (AML) attack defense mechanism integrated with HAASNet [35]. Datasets related to the healthcare sector are valuable. Moreover, the prediction from HAASNet is the key to the diagnosis of lung cancer using the HAAS model. However, successful AML attacks can hamper the overall integrity of the model. These limitations keep the window open to conduct further research in the cybersecurity domain on the HAAS model.

Cyber-physical system security: The HAAS model is dependent on IoMT data. Although the data transmitted by the IoMT devices are encrypted, no effective measurement has been taken in this model to protect the physical infrastructure of the model from Cyber-Physical System (CPS) attack [77]. It is another limitation of the HAAS model, which paves the path to explore possible solutions in subsequent research.

Any computer-based solution has limitations. The proposed HAAS mode is not immune to it. However, these limitations are pathways to improve the quality of this service model and ensure seamless automatic lung cancer diagnosis service.

7. Conclusion and discussion

The Healthcare-As-A-Service (HAAS) model streamlines the process of diagnosing lung cancer. While patients still need to visit radiologists for lung CT scans, the proposed service model automates the subsequent steps. The primary indicators of lung cancer are the presence of nodules and specific physio-symptomatic data. Traditionally, radiologists identify lung cancer nodules and prepare reports, which physicians then analyze alongside CT scans and symptom assessments to make a final diagnosis. This conventional approach requires multiple appointments with radiologists and physicians, making it a complex, multi-step process.

One of the key aspects of the Healthcare-As-A-Service model is the integration of IoMT, CNN, and Cloud Computing. This innovative approach replaces the necessity of physical inspection of the symptoms by IoMT. The IoMT integrated into the HAAS model has the capability of scanning twelve symptoms, which are crucial indicators of lung cancer. The carefully engineered device transmits these symptomatic data after encryption, which ensures data integrity and security. Only designated physicians have access to these symptomatic data to diagnose and make decisions about lung cancer. Neither patients nor physicians need to physically travel to the diagnostic center in this convenient approach, which is another novel contribution of this approach. Because it is a cloud-based service, it is accessible from anywhere worldwide. The CNN for the lung cancer classification, named HAASNet, has been designed keeping the challenges of cloud computing in mind. It has been engineered to ensure cloud scalability, resource optimization, load-balancing, and SLA maintenance. With all these unique features, remarkable capabilities, and outstanding design concepts, the HAASNet classifies lung cancer with 96.07% validation accuracy. The precision, recall, and F1-score of the classifier are 96.47%, 95.39%, and 94.81%, respectively.

Despite outstanding performance and the potential to revolutionize the lung cancer treatment process, the HAAS model suffers from several limitations. The first and most prominent limitation is the lack of performance analysis from different geographical locations. It is a common cloud computing characteristic that large-scale cloud applications can experience performance issues without data center replication in different locations. Unless it is studied thoroughly, the performance statistics regarding the geographical location is unclear. Furthermore, the system is scalable for up to eight simultaneous diagnosis requests. The experimenting cloud server suffers from performance issues when more than eight requests are simultaneously processed. Moreover, the sensor arrays used for the IoMT device used in this paper are not commercially available. That means the HAAS model is still in its early Research and Development (R&D) period. Additionally, the HAAS model is accessible through web-based services. No application for hand-held devices running on Android or iPhone Operating System (iOS) has been developed in this paper. The limitations of the methodology are not impediments to the further growth of the HAAS model as a practical lung cancer diagnosis service. The researchers of this project consider these limitations as an opportunity for further improvements. It opens multiple future directions to develop a better version of HAAS by developing a hand-held devices service model, enhancing the scalability, and conducting exploratory research on the impact on the performance in geographical locations.

The HAAS model has been developed for Computer Tomography (CT) images. It does not have the ability to classify lung cancer nodules from another image type, for example, X-ray images. This weakness paves the path to further experiments with the HAAS for X-ray images, opening a new research scope. Furthermore, it is a binary classifier that does not have the capability of classifying

Stage 1, Stage 2, Stage 3, and Stage 4 types of lung cancer. The HAASNet architecture is modifiable. Enabling it for multiclass classification is another future scope of this research. The Healthcare-As-A-Service (HAAS) model demonstrates the potential to be applied in other disease diagnoses. However, these are beyond the scope of the current phase of the research and will be explored in the future scope.

Preventing lung cancer-related deaths requires early diagnosis and appropriate treatment. Diagnostic complexities and costs often contribute to delays, but the HAAS model reduces these complexities and has the potential to lower diagnostic costs significantly. By providing a cloud-based solution, the HAAS model makes lung cancer diagnosis services more accessible to users worldwide. As a web-based service, anyone can access it from anywhere and receive a lung cancer diagnosis, ultimately contributing to a reduction in lung cancer mortality rates.

CRediT authorship contribution statement

Nuruzzaman Faruqui: Conceptualization, Methodology, Writing – original draft. **Mohammad Abu Yousuf:** Investigation, Supervision. **Faris A. Kateb:** Project administration, Validation. **Md. Abdul Hamid:** Resources, Writing – review & editing. **Muhammad Mostafa Monowar:** Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] K.C. Gogebakan, J. Lange, C.G. Slatore, R. Etzioni, Modeling the impact of novel systemic treatments on lung cancer screening benefits, Cancer 129 (2) (2023) 226–234.

[2] H. Zhang, N. Razmjooy, Optimal Elman neural network based on improved Gorilla Troops Optimizer for short-term electricity price prediction, J. Electr. Eng. Technol. (2023) 1–15.

[3] S. Huang, J. Yang, N. Shen, Q. Xu, Q. Zhao, Artificial intelligence in lung cancer diagnosis and prognosis: current application and future perspective, in: Seminars in Cancer Biology, Elsevier, 2023.

[4] J. Alonso, L. Orue-Echevarria, V. Casola, A.I. Torre, M. Huarte, E. Osaba, J.L. Lobo, Understanding the challenges and novel architectural models of multi-cloud native applications–a systematic literature review, J. Cloud Comput. 12 (1) (2023) 1–34.

[5] M.M. Warrier, L. Abraham, A review on early diagnosis of lung cancer from CT images using deep learning, in: Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences: PCCDS 2022, Springer, 2023, pp. 653–670.

[6] H. Sugawara, Y. Yatabe, H. Watanabe, H. Akai, O. Abe, S.-i. Watanabe, M. Kusumoto, Radiological precursor lesions of lung squamous cell carcinoma: early progression patterns and divergent volume doubling time between hilar and peripheral zones, Lung Cancer 176 (2023) 31–37.

[7] W. Sun, K. Zhang, S.-K. Chen, X. Zhang, H. Liang, Software as a service: an integration perspective, in: Service-Oriented Computing–ICSOC 2007: Fifth International Conference, Proceedings 5, Vienna, Austria, September 17–20, 2007, Springer, 2007, pp. 558–569.

[8] N. Pramod, A.K. Muppalla, K. Srinivasa, Limitations and challenges in cloud-based applications development, in: Software Engineering Frameworks for the Cloud Computing Paradigm, 2013, pp. 55–75.

[9] G. Blinowski, A. Ojdowska, A. Przybyłek, Monolithic vs. microservice architecture: a performance and scalability evaluation, IEEE Access 10 (2022) 20357–20374.

[10] S.H. Hosseini, R. Monsefi, S. Shadroo, Deep learning applications for lung cancer diagnosis: a systematic review, Multimed. Tools Appl. (2023) 1–31.

[11] G. Vijaya, Deep learning-based computer-aided diagnosis system, in: Application of Deep Learning Methods in Healthcare and Medical Science, Apple Academic Press, 2022, pp. 23–48.

[12] M. Cellina, M. Cè, G. Irmici, V. Ascenti, N. Khenkina, M. Toto-Brocchi, C. Martinenghi, S. Papa, G. Carrafiello, Artificial intelligence in lung cancer imaging: unfolding the future, Diagnostics 12 (11) (2022) 2644.

[13] N. Faruqui, M.A. Yousuf, M. Whaiduzzaman, A. Azad, A. Barros, M.A. Moni, LungNet: a hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data, Comput. Biol. Med. 139 (2021) 104961.

[14] O. Singh, K.L. Kashyap, K.K. Singh, Mesh-free technique for enhancement of the lung CT image, Biomed. Signal Process. Control 81 (2023) 104452.

[15] U. Demiroğlu, B. Şenol, M. Yildirim, Y. Eroğlu, Classification of computerized tomography images to diagnose non-small cell lung cancer using a hybrid model, Multimed. Tools Appl. (2023) 1–22.

[16] M. Chen, A comparative study of transfer learning based models for lung cancer histopathology classification, Highlights Sci. Eng. Technol. 39 (2023) 26–34.

[17] C. Venkatesh, N. Sai Prasanna, Y. Sudeepa, P. Sushma, Detection and classification of lung cancer using optimized two-channel CNN technique, in: Advances in Cognitive Science and Communications: Selected Articles from the 5th International Conference on Communications and Cyber-Physical Engineering (ICCCE 2022), Springer, Hyderabad, India, 2023, pp. 305–317.

[18] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[19] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 210–214.

[20] A.R. Bushara, R.S. Vinod Kumar, S.S. Kumar, LCD-capsule network for the detection and classification of lung cancer on computed tomography images, Multimed. Tools Appl. (2023) 1–20.

[21] H. Mkindu, L. Wu, Y. Zhao, Lung nodule detection of CT images based on combining 3D-CNN and squeeze-and-excitation networks, Multimed. Tools Appl. (2023) 1–14.

[22] D. Kawahara, N. Imano, R. Nishioka, Y. Nagata, Image masking using convolutional networks improves performance classification of radiation pneumonitis for non-small cell lung cancer, Phys. Eng. Sci. Med. (2023) 1–6.

[23] K. Barbouchi, D. El Hamdi, I. Elouedi, T.B. Aïcha, A.K. Echi, I. Slim, A transformer-based deep neural network for detection and classification of lung cancer via PET/CT images, Int. J. Imaging Syst. Technol. (2023).

[24] N. Maleki, S.T.A. Niaki, An intelligent algorithm for lung cancer diagnosis using extracted features from computerized tomography images, Healthc. Anal. 3 (2023) 100150.

[25] W. Liu, X. Liu, X. Luo, M. Wang, G. Han, X. Zhao, Z. Zhu, A pyramid input augmented multi-scale CNN for GGO detection in 3D lung CT images, Pattern Recognit. 136 (2023) 109261.

[26] C. Shankara, S. Hariprasad, D. Latha, Detection of lung cancer using convolution neural network, SN Comput. Sci. 4 (3) (2023) 225.

[27] E.A. Siddiqui, V. Chaurasia, M. Shandilya, Detection and classification of lung cancer computed tomography images using a novel improved deep belief network with Gabor filters, Chemom. Intell. Lab. Syst. (2023) 104763.

[28] M.M. Hossain, M.M. Hasan, M.A. Rahim, M.M. Rahman, M.A. Yousuf, S. Al-Ashhab, H.F. Akhdar, S.A. Alyami, A. Azad, M.A. Moni, Particle swarm optimized fuzzy CNN with quantitative feature fusion for ultrasound image quality identification, IEEE J. Transl. Eng. Health Med. 10 (2022) 1–12.

[29] T.C. Hiremath, K.S. Rekha, Optimization enabled deep learning method in container-based architecture of hybrid cloud for portability and interoperability-based application migration, J. Exp. Theor. Artif. Intell. (2022) 1–18.

[30] S.K. Kishore, G. Vasukidevi, E.P.C. Prasad, T.R. Patnala, V.P. Reddy, P.B. Chanda, A real-time machine learning based cloud computing architecture for smart manufacturing, in: 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), IEEE, 2022, pp. 562–565.

[31] M. Sachdeva, A.K.S. Kushwaha, et al., The power of deep learning for intelligent tumor classification systems: a review, Comput. Electr. Eng. 106 (2023) 108586.

[32] D. Bahmanyar, N. Razmjooy, S. Mirjalili, Multi-objective scheduling of IoT-enabled smart homes for energy management based on arithmetic optimization algorithm: a node-RED and NodeMCU module-based technique, Knowl.-Based Syst. 247 (2022) 108762.

[33] J. Indumathi, A. Shankar, M.R. Ghalib, J. Gitanjali, Q. Hua, Z. Wen, X. Qi, Block chain based internet of medical things for uninterrupted, ubiquitous, user-friendly, unflappable, unblemished, unlimited health care services (BC IoMT U6 HCS), IEEE Access 8 (2020) 216856–216872.

[34] D. Swift, K. Cresswell, R. Johnson, S. Stilianoudakis, X. Wei, A review of normalization and differential abundance methods for microbiome counts data, Wiley Interdiscip. Rev.: Comput. Stat. 15 (1) (2023) e1586.

[35] S. Trivedi, T.A. Tran, N. Faruqui, M.M. Hassan, An exploratory analysis of effect of adversarial machine learning attack on IoT-enabled industrial control systems, in: 2023 International Conference on Smart Computing and Application (ICSCA), IEEE, 2023, pp. 1–8.

[36] L.P.O. Paula, N. Faruqui, I. Mahmud, M. Whaiduzzaman, E.C. Hawkinson, S. Trivedi, A novel front door security (FDS) algorithm using GoogleNet-BiLSTM hybridization, IEEE Access 11 (2023) 19122–19134.

[37] S. Trivedi, N. Patel, N. Faruqui, NDNN based U-Net: an innovative 3D brain tumor segmentation method, in: 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, 2022, pp. 0538–0546.

[38] E. Dandıl, A computer-aided pipeline for automatic lung cancer classification on computed tomography scans, J. Healthc. Eng. 2018 (2018).

[39] C. Jacobs, E.M. van Rikxoort, K. Murphy, M. Prokop, C.M. Schaefer-Prokop, B. van Ginneken, Computer-aided detection of pulmonary nodules: a comparative study using the public LIDC/IDRI database, Eur. Radiol. 26 (7) (2016) 2139–2147.

[40] S.G. Armato III, G. McLennan, L. Bidaut, M.F. McNitt-Gray, C.R. Meyer, A.P. Reeves, B. Zhao, D.R. Aberle, C.I. Henschke, E.A. Hoffman, et al., The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans, Med. Phys. 38 (2) (2011) 915–931.

[41] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, et al., The cancer imaging archive (TCIA): maintaining and operating a public information repository, J. Digit. Imag. 26 (2013) 1045–1057.

[42] S. Armato, G. McLennan, M. McNitt-Gray, C. Meyer, A. Reeves, L. Bidaut, B. Zhao, B. Croft, L. Clarke, We-b-201b-02: the lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed public database of CT scans for lung nodule analysis, Med. Phys. 37 (6) (2010) 3416–3417.

[43] J.S. Kirby, S.G. Armato, K. Drukker, F. Li, L. Hadjiiski, G.D. Tourassi, L.P. Clarke, R.M. Engelmann, M.L. Giger, G. Redmond, et al., LUNGx challenge for computerized lung nodule classification, J. Med. Imag. 3 (4) (2016) 044506.

[44] N. Faruqui, Open Source Computer Vision for Beginners: Learn OpenCV Using C++ in Fastest Possible Way, 2017.

[45] N.F. Aurna, M.A. Yousuf, K.A. Taher, A. Azad, M.A. Moni, A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models, Comput. Biol. Med. 146 (2022) 105539.

[46] K.U. Ahamed, M. Islam, A. Uddin, A. Akhter, B.K. Paul, M.A. Yousuf, S. Uddin, J.M. Quinn, M.A. Moni, A deep learning approach using effective preprocessing techniques to detect COVID-19 from chest CT-scan and X-ray images, Comput. Biol. Med. 139 (2021) 105014.

[47] T.R. Mim, M. Amatullah, S. Afreen, M.A. Yousuf, S. Uddin, S.A. Alyami, K.F. Hasan, M.A. Moni, GRU-INC: an inception-attention based approach using GRU for human activity recognition, Expert Syst. Appl. 216 (2023) 119419.

[48] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.

[49] M. Ramezani, D. Bahmanyar, N. Razmjooy, A new optimal energy management strategy based on improved multi-objective antlion optimization algorithm: applications in smart home, SN Appl. Sci. 2 (2020) 1–17.

[50] N. Faruqui, M.A. Yousuf, M. Whaiduzzaman, A. Azad, S.A. Alyami, P. Liò, M.A. Kabir, M.A. Moni, SafetyMed: a novel IoMT intrusion detection system using CNN-LSTM hybridization, Electronics 12 (17) (2023) 3541.

[51] Y. Yang, L. Zhang, M. Du, J. Bo, H. Liu, L. Ren, X. Li, M.J. Deen, A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions, Comput. Biol. Med. 139 (2021) 104887.

[52] I. Singh, G. Goyal, A. Chandel, AlexNet architecture based convolutional neural network for toxic comments classification, J. King Saud Univ, Comput. Inf. Sci. 34 (9) (2022) 7547–7558.

[53] S. Achar, N. Faruqui, M. Whaiduzzaman, A. Awajan, M. Alazab, Cyber-physical system security based on human activity recognition through IoT cloud computing, Electronics 12 (8) (2023).

[54] M. Valipour, H. Khoshkam, S.M. Bateni, C. Jun, S.S. Band, Hybrid machine learning and deep learning models for multi-step-ahead daily reference evapotranspiration forecasting in different climate regions across the contiguous United States, Agric. Water Manag. 283 (2023) 108311.

[55] C.L. Galimberti, L. Furieri, L. Xu, G. Ferrari-Trecate, Hamiltonian deep neural networks guaranteeing non-vanishing gradients by design, IEEE Trans. Autom. Control (2023).

[56] Q. Ni, X. Kang, A novel decomposition-based multi-objective evolutionary algorithm with dual-population and adaptive weight strategy, Axioms 12 (2) (2023) 100.

[57] M. Zambra, A. Testolin, M. Zorzi, A developmental approach for training deep belief networks, Cogn. Comput. 15 (1) (2023) 103–120.

[58] W. Boulila, M. Driss, E. Alshanqiti, M. Al-Sarem, F. Saeed, M. Krichen, Weight initialization techniques for deep learning algorithms in remote sensing: recent trends and future perspectives, in: Advances on Smart and Soft Computing: Proceedings of ICACIn 2021, 2022, pp. 477–484.

[59] S.-H. Wang, Y. Chen, Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique, Multimed. Tools Appl. 79 (2020) 15117–15133.

[60] S. Raziani, M. Azimbagirad, Deep CNN hyperparameter optimization algorithms for sensor-based human activity recognition, Neurosci. Inform. 2 (3) (2022) 100078.

[61] A. Botev, G. Lever, D. Barber, Nesterov's accelerated gradient and momentum as approximations to regularised update descent, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 1899–1903.

[62] A. Lydia, S. Francis, AdaGrad - an optimizer for stochastic gradient descent, Int. J. Inf. Comput. Sci. 6 (5) (2019) 566–568.

[63] K.P. Rakshitha, N. Naveen, Op-RMSprop (optimized-root mean square propagation) classification for prediction of polycystic ovary syndrome (PCOS) using hybrid machine learning technique, Int. J. Adv. Comput. Sci. Appl. 13 (6) (2022).

[64] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.

[65] D.L. Streiner, J. Cairney, What's under the ROC? An introduction to receiver operating characteristics curves, Can. J. Psychiatry 52 (2) (2007) 121–128.

[66] S. Lakshmanaprabu, S.N. Mohanty, K. Shankar, N. Arunkumar, G. Ramirez, Optimal deep learning model for classification of lung cancer on CT images, Future Gener. Comput. Syst. 92 (2019) 374–382.

[67] C. Liu, S.-C. Hu, C. Wang, K. Lafata, F.-F. Yin, Automatic detection of pulmonary nodules on CT images with YOLOv3: development and evaluation using simulated and patient data, Quant. Imaging Med. Surg. 10 (10) (2020) 1917.

[68] T.L. Chaunzwa, A. Hosny, Y. Xu, A. Shafer, N. Diao, M. Lanuti, D.C. Christiani, R.H. Mak, H.J. Aerts, Deep learning classification of lung cancer histology using CT images, Sci. Rep. 11 (1) (2021) 5471.

[69] Y. Onozato, T. Iwata, Y. Uematsu, D. Shimizu, T. Yamamoto, Y. Matsui, K. Ogawa, J. Kuyama, Y. Sakairi, E. Kawakami, et al., Predicting pathological highly invasive lung cancer from preoperative [18f]FDG PET/CT with multiple machine learning models, Eur. J. Nucl. Med. Mol. Imaging 50 (3) (2023) 715–726.

[70] T.I. Mohamed, O.N. Oyelade, A.E. Ezugwu, Automatic detection and classification of lung cancer CT scans based on deep learning and Ebola optimization search algorithm, PLoS ONE 18 (8) (2023) 0285796.

[71] X. Fei, R. Xuejun, N. Razmjooy, Optimal configuration and energy management for combined solar chimney, solid oxide electrolysis, and fuel cell: a case study in Iran, Energy Sources Part B: Recovery Util. Environ. Eff. (2019) 1–21.

[72] G. Zhang, C. Xiao, N. Razmjooy, Optimal operational strategy of hybrid PV/wind renewable energy system using homer: a case study, Int. J. Ambient Energy 43 (1) (2022) 3953–3966.

[73] S.M.A. Bokhari, F. Azam, et al., Limitations of service oriented architecture and its combination with cloud computing, Bahria Univ. J. Inf. Commun. Technol. 8 (1) (2015).

[74] H. Hosseini, B. Xiao, M. Jaiswal, R. Poovendran, On the limitation of convolutional neural networks in recognizing negative images, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2017, pp. 352–358.

[75] Y.B. Zikria, M.K. Afzal, S.W. Kim, Internet of multimedia things (IoMT): opportunities, challenges and solutions, Sensors 20 (8) (2020) 2334.

[76] Y. Liu, Y. Ping, L. Zhang, L. Wang, X. Xu, Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning, Robot. Comput.-Integr. Manuf. 80 (2023) 102454.

[77] S. Achar, N. Faruqui, M. Whaiduzzaman, A. Awajan, M. Alazab, Cyber-physical system security based on human activity recognition through IoT cloud computing, Electronics 12 (8) (2023) 1892.