Ingénierie des Systèmes d'Information
Vol. 30, No. 5, May, 2025, pp. 1177-1188
Journal homepage: http://iieta.org/journals/isi

A Custom Federated Weighted Learning Model for Melanoma Detection Using Deep Neural Networks

Ashwini Jewalikar[1,2*], Rais Abdul Hamid Khan[1], Deepak T. Mane[3]

[1] Department of CSE, Sandip University, Nasik 422001, India
[2] Pune Institute of Computer Technology, SPPU, Pune 411043, India
[3] Department of Computer Engineering, VIT, Pune 411037, India

Corresponding Author Email: ashwinijewalikar@gmail.com

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.300506

Received: 24 February 2025
Revised: 23 April 2025
Accepted: 3 May 2025
Available online: 31 May 2025

Keywords:
skin cancer detection, federated learning, distributed learning, data privacy, privacy-focused image analysis, federated averaging approaches, deep learning models

ABSTRACT

A timely and accurate diagnosis of skin cancer is crucial, as it is a potentially life-threatening condition. However, implementing traditional machine learning algorithms in healthcare presents substantial challenges, particularly in maintaining data privacy and effectively managing distributed, non-IID datasets. To address these issues, we propose a Custom Federated Weighted Learning (CFWL) model for melanoma detection, leveraging federated learning (FL) to ensure privacy preservation while enhancing the accuracy and convergence of the central model. The proposed approach incorporates two advanced deep learning architectures, CNN and VGG16, to improve feature extraction and classification performance. Our model is evaluated on a skin cancer dataset under non-IID conditions and compared with existing FL techniques, including FedAvg, FedProx, FedAdagrad, FedAdam, and FedYogi. Experimental results demonstrate that our CFWL model with VGG model outperforms these baseline approaches in terms of classification accuracy, loss, and model robustness. This work highlights the potential of FL, augmented by advanced deep learning methods, to deliver scalable, privacy-preserving solutions for melanoma detection and other critical healthcare applications.

1. INTRODUCTION

Skin cancer is the most prevalent type of cancer globally. Initial diagnosis typically involves clinical screenings, followed by confirmatory procedures such as biopsies, histological tests, and dermoscopy [1]. Skin cancer develops when the normal growth of skin cells is disrupted, leading to DNA mutations that result in malignant cell growth. While ultraviolet (UV) radiation is the primary cause of skin cancer, other contributing factors include fair skin, exposure to radiation and harmful chemicals, severe skin injuries or burns, weakened immunity, aging, and smoking. Early diagnosis is critical, and researchers are increasingly leveraging artificial intelligence-based techniques to achieve this goal. Skin cancers can be broadly classified into melanoma and non-melanoma types, with the latter being generally less aggressive and more treatable. These cancers, though less severe than melanoma, still require timely diagnosis and treatment to prevent complications [2].

Significant advancements have been achieved in the use of AI for identifying disease patterns from medical imaging [3]. In dermatology, AI-based tools and applications are being developed to assess the severity of conditions such as psoriasis and to perform specialized dermatological tasks, such as classifying skin lesions as melanoma or nonmelanoma skin cancer. These tools utilize sophisticated algorithms capable of self-learning and improving their accuracy over time [4].

The integration of federated learning (FL), deep learning, and transfer learning technologies [5] offers substantial benefits to both patients and dermatologists by enhancing the prediction and diagnosis of suspicious skin lesions. In this work, we explored various federated averaging techniques and deep learning algorithms, benchmarked public datasets, and examined for melanoma classification. This work provides a comprehensive resource on the application of deep and FL techniques for diagnosing malignant melanoma and non-melanoma skin cancers, aiming to advance research and clinical applications in the field.

1.1 FL and its working

A distributed machine learning approach called FL enables several devices or edge nodes to work together to train models without exchanging local data. By storing data locally and sending only model updates to a central server, this method solves privacy and security issues with data. Data from multiple sources is combined on a single server for training in a standard centralized machine learning arrangement, which presents privacy concerns, particularly in delicate industries like healthcare and finance [4]. By allowing each participating device to independently train a model on its own data and then transmitting just the trained model parameters—like gradients or weights—to the central server, FL gets around this problem. To enhance a global model, the server gathers these updates,

frequently by averaging them. The global model is returned to the devices for additional improvement as part of this iterative process. Figure 1 illustrates the working mechanism of FL within a healthcare system.

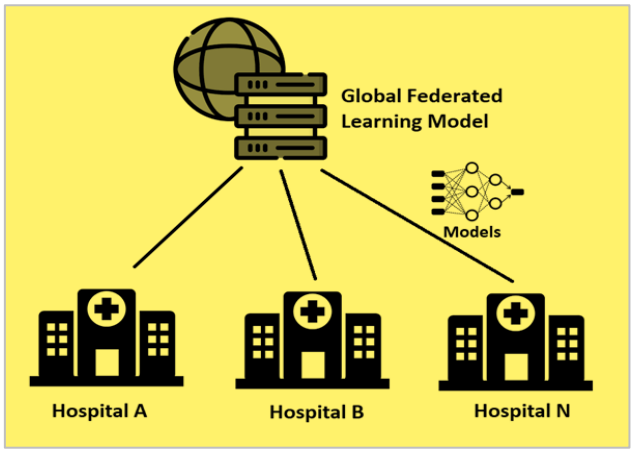

Figure 1. Working of FL within healthcare system

FL 's main benefits go beyond privacy; they also include lower latency and data transfer. Furthermore, as every client (device) may have different data distributions, FL is appropriate for handling extremely diverse data. But FL also has to deal with issues like heterogeneous models, connectivity limitations, and disparate device data quality. Furthermore, FL-participating devices frequently have limited memory and power, necessitating the use of effective model updating techniques. In order to overcome these obstacles and guarantee that FL keeps providing privacy-preserving, effective, and scalable solutions across industries, it is necessary to create strong aggregation techniques, increase communication efficiency, and integrate strategies for managing non-iid (non-independent and identically distributed) data.

1.2 FL in healthcare

FL, which permits cooperative machine learning model training across several healthcare facilities while protecting data privacy, is quickly becoming a game-changing strategy in the healthcare industry. Sensitive by nature, healthcare data includes private health information that is subject to stringent privacy laws such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). With the abundance of digital health data—from wearable technology to medical imaging and electronic health records (EHRs).

Each medical facility (such as hospitals, clinics, and research centers) keeps its own local patient data and uses it to train a model locally in a FL configuration for healthcare. The institution does not share the raw data with a central server; instead, it shares the learnt parameters, like gradients or model weights. By combining these updates, the central server improves a global model that takes advantage of the various data sources spread throughout different institutions. Due to regional variations in patient demographics, disease incidence, and treatment practices, the global model can achieve high predictive accuracy and generalization capacity by repeating this procedure. This is particularly crucial in the healthcare industry.

Implementing FL in healthcare faces challenges such as non-IID data distribution across institutions, high communication costs for large medical datasets, and the need for privacy-preserving techniques like secure multiparty computation and differential privacy, which increase complexity. Despite these challenges, FL holds promise for developing secure, collaborative models that enhance clinical outcomes, enabling powerful diagnostic and predictive tools to improve healthcare and patient care.

1.3 FL averaging techniques

Aggregation techniques are extremely important in FL, as they are responsible for combining model updates from many client devices that are located in different locations in order to develop a robust global model. These aggregation strategies are meant to accommodate varied data distributions, client unpredictability, and bandwidth limits while simultaneously ensuring that the final model is accurate and safe. Both of these goals are accomplished simultaneously. The following is a list of some of the most important aggregation techniques that are typically utilized in FL. Various federated aggregation techniques are presented in Figure 2 and elaborated below.

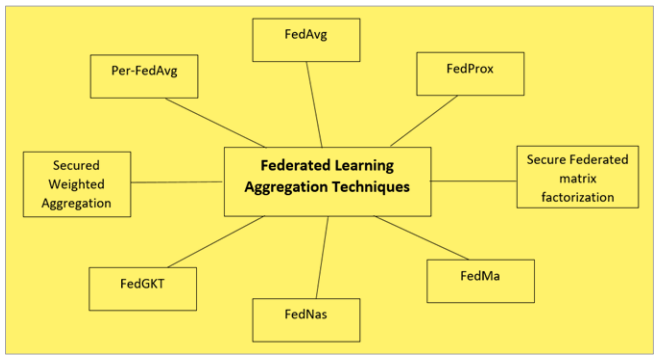

Figure 2. FL aggregation approach

1.3.1 FedAvg

Clients use Federated Averaging (FedAvg), training locally and sending model updates to a central server, which averages them to create a global model. It's efficient and effective when client data is similar but may struggle with non-iid data. It's widely used in homogeneous settings for its simplicity and performance.

1.3.2 FedSGD

Federated Stochastic Gradient Descent (FedSGD) modifies SGD for FL, sending gradient updates instead of model parameters. It reduces local training but is communication-intensive and less suitable for bandwidth-limited applications. It's fast but not scalable for large, decentralized networks.

1.3.3 Adaptive federated optimization

Techniques like FedAdam, FedYogi, and FedAdagrad adjust learning rates to handle data heterogeneity, stabilizing training and speeding up convergence, especially with non-iid data. However, they are computationally intensive and require more resources on both client and server.

The challenges of privacy and communication efficiency drive the need for innovative approaches in healthcare, particularly for skin cancer prediction. In this paper, we propose a custom FL averaging technique integrated with deep learning for melanoma detection, aiming to improve both communication efficiency and prediction accuracy while

addressing data privacy concerns. Our approach introduces a decentralized, privacy-aware model tailored for skin cancer prediction in healthcare settings.

Key contributions include:

(1) An in-depth review of FL-based AI applications in healthcare, covering foundational concepts of FL, essential AI principles, and FL averaging techniques are discussed.
(2) Introduced a novel approach leveraging CNNs and VGG16 at client nodes, integrated with a customized FL averaging strategy for global model aggregation.
(3) Conducted a comprehensive evaluation on a skin cancer dataset, showcasing that our proposed approach outperforms baseline FL methods by delivering faster convergence, improved prediction accuracy, and enhanced communication efficiency.

The paper is organized as follows: Section 2 covers prior FL reviews; Section 3 introduces the proposed methodology of FL averaging and algorithms used for classification of melanoma disease; Section 4 provides experimental setup, performance parameters, result analysis and comparative analysis of proposed results with similar systems; Section 5 discusses conclusion and future scope and references are mention at end of article.

2. LITERATURE REVIEW

The literature review compares various techniques, including machine learning, deep learning, and transfer learning algorithms, applied to healthcare datasets within a FL environment. It also examines different FL frameworks and federated averaging methods, analysing their effectiveness in addressing challenges such as data privacy, communication efficiency, and model convergence. This comprehensive analysis highlights the strengths and limitations of these approaches in healthcare applications.

2.1 Machine learning techniques

Non-melanoma skin cancers (NMSCs) have been extensively studied in recent research, with a particular focus on their biology and clinical features [2]. This study emphasized the importance of early detection and personalized treatment approaches, while also detailing the epidemiology, risk factors, and clinical manifestations of these tumors. Available treatment options, including both surgical and non-surgical methods, have been thoroughly examined, along with emerging therapies that target specific molecular alterations in NMSCs.

Recent research has examined the challenges faced by traditional machine learning algorithms and explored the integration of FL into privacy-aware healthcare systems for skin cancer prediction [6]. The study provides a comparative analysis of the performance of various machine learning and FL algorithms in detecting skin lesions, along with an evaluation of different datasets used for skin cancer prediction. Mobile-based artificial intelligence approaches show promise for improving melanoma identification in skin cancer diagnosis, enabling rapid and accurate detection through mobile devices [7]. The suggested system provides a viable method for early melanoma diagnosis in resource-constrained environments by combining machine learning models with image processing methods.

The artificial bee colony (ABC) optimization method has been adapted for FL feature selection to enhance the accuracy and efficiency of heart disease diagnosis in clinical settings, combining the strengths of both FL techniques and ABC optimization [8].

FL has been comprehensively reviewed with detailed examination of its core technologies, communication protocols, and diverse application domains [9]. The growing interest in FL which has surfaced as a viable paradigm for data privacy-preserving collaborative machine learning across decentralized devices is the subject of this survey. A comparative analysis between FL and traditional machine learning approaches was conducted using COVID-19 chest X-ray datasets, evaluating the impact of key parameters including activation functions, optimizers, learning rates, training iterations, and dataset sizes on model accuracy and loss metrics [10]. Results showed that FL outperformed traditional models in accuracy and loss, particularly with SGD optimizers and softmax activation, though it required more computational time.

A novel framework combining secure multiparty computation and differential privacy has been developed to optimize the accuracy-privacy trade-off in collaborative machine learning systems [11, 12]. By integrating these strategies, the suggested approach preserves privacy without jeopardizing a certain degree of confidence by reducing the noise level as the number of participants increases. The study [12] also presented an innovative use of FL to safeguard privacy in healthcare data. This project investigates the use of FL to improve data privacy and facilitate cross-institutional collaborative machine learning. This strategy addresses privacy issues associated with traditional centralised techniques by maintaining patient data localised and only exchanging model changes, guaranteeing the confidentiality of sensitive information.

A groundbreaking framework utilizing federated electronic health records (EHRs) has been developed for predictive modeling through FL approaches [13]. In order to protect patient privacy, this work tackles the difficulty of collaborative predictive modelling across dispersed healthcare systems. The authors provide a method for training prediction models on decentralised EHR data sources without revealing sensitive patient information by utilising FL techniques.

2.2 Deep learning techniques

Significant advancements in automated non-melanoma skin cancer (NMSC) detection have been achieved through the development of Multi-Site Cross-Organ Calibrated Deep Learning (MuSCID), a calibrated deep learning framework designed to maintain consistent diagnostic accuracy across diverse clinical settings while addressing data variability from multiple healthcare institutions [14]. The MuSClD model improves its generalisation ability by utilising a large dataset that includes a variety of pictures of NMSC from different organs.

Deep residual networks have been successfully applied to enhance melanoma diagnostic accuracy by leveraging their powerful feature extraction capabilities [15], while simultaneously addressing common challenges in dermoscopic image analysis including lesion variability and imaging artifacts through an optimized network architecture. Hybrid fully convolutional networks (FCNs) incorporating deep features offer a novel solution for improving both skin

lesion segmentation and melanoma detection accuracy [16]. The suggested technique delivers robust melanoma detection and improves segmentation performance by combining FCNs with deep features taken from pre-trained convolutional neural networks (CNNs). Optimized deep learning features significantly advance melanoma diagnosis capabilities [17]. The objective of the research is to improve melanoma diagnosis accuracy by the optimisation of deep learning features derived from CNNs. By utilising sophisticated methods for feature extraction and selection, the authors show enhanced ability to distinguish between benign lesions and malignant melanomas.

Neural network depth, learning process design, and high-quality training data critically impact model performance [18] By employing deep learning architectures and large datasets, the scientists hope to improve the precision and dependability of melanoma diagnosis. In order to enhance screening performance, the study looks into a number of design options, such as data augmentation and model optimisation strategies.

Fine-tuning strategies and source task selection significantly affect transfer learning performance for melanoma detection [19]. Additionally, they evaluate the performance gains by using deeper and more complex models. Their results show that deeper models, pretrained on ImageNet, perform far better when optimised for particular applications. Using two different skin-lesion datasets, the study assesses these models and shows that deeper, more refined models significantly increase prediction accuracy.

A hybrid approach integrating support vector machines, deep learning, and sparse coding techniques demonstrates enhanced performance for melanoma identification in dermoscopic images [20]. The combination of natural image feature transfer and unsupervised learning eliminates the necessity for annotated data in the target task, which is a major benefit. Using a technique akin to that of clinical specialists, the system is able to compare dermoscopic visual patterns with observations from the real world.

2.3 Transfer learning techniques

The VGG16 CNN architecture effectively detects and mitigates data poisoning attacks in deep learning systems [21]. This method allows for collaborative model training while protecting sensitive medical data, and it is implemented inside a FL architecture that involves 10 healthcare institutions. To effectively diagnose skin cancer, the FL technique makes use of VGG16's powerful feature extraction capabilities. Using strict criteria and outlier detection algorithms to identify and assess suspicious model alterations, the study presented a complete approach to FL data poisoning threat identification.

A comprehensive taxonomy classifies both malignant and non-malignant skin cancers while reviewing state-of-the-art federated and transfer learning algorithms for malignant lesion detection [22].

FedPacket introduces a FL framework for mobile packet classification that enhances both privacy protection and network performance [23]. FedPacket maintains excellent classification accuracy while protecting user privacy by allowing decentralised training across several devices without exchanging raw data. The authors carry out thorough analyses, showing that FedPacket performs better in terms of efficacy and privacy preservation than conventional centralised methods. By providing a solid method for packet categorization that respects privacy, this study makes a substantial contribution to the field of mobile computing.

FL demonstrates transformative potential in healthcare by enabling privacy-preserving collaborative research across institutions, with distinct applications ranging from neuro-oncology to comparative oncology. In brain lesion analysis—including traumatic injuries, gliomas, and stroke—FL improves predictive accuracy while maintaining data security [2]. The framework similarly enhances diagnostic precision in broader medical applications through secure distributed model training [4]. Complementary translational research further illustrates FL's value in comparative oncology, where canine melanoma models provide insights into human tumor behavior and treatment responses [8]. These collective advances highlight FL's dual capacity to accelerate medical innovation while addressing critical data privacy constraints in healthcare research.

The literature review highlights the transformative role of FL in healthcare, focusing on its ability to enable collaborative data analysis while preserving patient privacy. These works also address challenges such as communication overhead, data heterogeneity, and privacy concerns, proposing future research directions. Together, the review underscores FL's potential to revolutionize healthcare by balancing privacy, collaboration, and innovation.

3. RELATED WORK

In this section, we focus on various federated averaging techniques, including FedAvg, FedProx, FedYogi, FedAdagrad, and FedAdam. Each of these methods is explained in detail, highlighting their unique approaches to model aggregation and their impact on communication efficiency, convergence, and overall performance in FL environments.

3.1 Federated averaging (FedAvg)

With the introduction of FL, the most fundamental aggregating technique is known as Federated Averaging (FedAvg). This technique was developed to enable decentralized model training without the need to share raw data. In order to enhance a global model, the fundamental concept is to take the average of model updates from a number of different clients (for example, devices or institutions). In FedAvg, every client N is responsible for updating its local model by executing several epochs of stochastic gradient descent (SGD) on its local data, which is represented by Dk. At the same time, a loss function Lk (w) is applied, which is dependent on the distribution of the client's data.

Each client computes a weight update Δw_k locally as:

$$\Delta w_k = w_k^t - \eta \nabla L_k(w_k^t)$$

where, w_k^t wkt is the model parameters for client k at round t, and η is the learning rate. After multiple local updates, the central server aggregates these by taking a weighted average based on the number of data samples n_k at each client, updating the global model as follows:

$$w^{t+1} = \frac{\sum_{k=1}^{K} n_k \cdot w_k^{t+1}}{w_{k=1}^{K} n_k}$$

The updated global model is denoted by the symbol w^{t+1},

while the total number of customers is denoted by the symbol n. In contexts where the data from all of the clients is consistent, FedAvg is a straightforward, efficient, and effective solution. On the other hand, FedAvg may have difficulty generalizing successfully among clients when data distributions are very heterogeneous (non-iid). This is because the averaging of gradients, which may represent competing updates from different data distributions, might make it difficult for FedAvg to generalize well.

3.2 Federated proximal (FedProx)

FedProx is an extension of FedAvg that was developed to better manage the heterogeneous data distributions that occur between customers. It does this by incorporating a proximal term into the loss function, which helps to minimize the divergence between the local model of each client and the global model. This helps to address problems that occur when the data distributions of clients are significantly different from one another. By punishing departures from the global model, this term essentially regularizes the updates that each client makes, so ensuring that stability is maintained during the aggregation process.

The local optimization objective for each client k in FedProx is modified as follows:

$$L_k(w) = L_k(w) + \frac{\mu}{2} ||w - w^t||^2$$

where, w is the current model on the client, w^t is the global model at round t, and μ is a regularization parameter that controls the influence of the proximal term. The proximal term $\frac{\mu}{2} ||w - w^t||^2$ encourages the local model w to stay close to the global model wt, thereby mitigating the effects of client-specific data distributions. FedProx performs the aggregation step similarly to FedAvg, using the weighted average of updates from each client. By adjusting μ, FedProx can balance model convergence and stability, especially in non-iid settings, making it a powerful tool for FL with high client variability.

3.3 Federated adagrad (FedAdagrad)

FedAdagrad is a variation of FedAvg that utilizes adaptive learning rates with the use of the Adagrad algorithm for optimization. Learning rates in such an algorithm depend on update frequency and update size. It proves useful in federated settings with unbalanced distributions of data at the clients, with FedAdagrad decreasing learning rates for updated parameters, with a bias towards stabilizing convergence. This makes FedAdagrad particularly valuable in these kinds of contexts. In FedAdagrad, each client k updates its model using a local gradient gk and an adaptive learning rate ηk, which depends on the accumulated squared gradients:

$$w^{t+1} = w^t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_k$$

where, $G_t = \sum_{i=1}^{t} g_i^2$ is the sum of squared gradients from previous rounds up to t, and ϵ is a small constant to prevent division by zero. The adaptive term $\frac{\eta}{\sqrt{G_t+\epsilon}}$ reduces the learning rate for frequently updated parameters, leading to more stable updates. After local training, each client's model updates are aggregated using the weighted average method, similar to FedAvg. FedAdagrad enhances FedAvg by enabling smoother convergence, especially for heterogeneous data, as it naturally adapts to the learning rates based on parameter-specific update history.

3.4 Federated adam (FedAdam)

By merging the concepts of momentum and adjustable learning rates, FedAdam is able to apply the Adam optimization technique to the federated environment. Accelerating convergence and stabilizing updates are two benefits that result from the use of exponential moving averages in Adam, which smoothes gradients. In non-iid federated contexts, FedAdam is effective because it combines the adaptive updates of Adagrad with momentum, which helps eliminate oscillations in gradient updates. This allows FedAdam to maximize its effectiveness. FedAdam involves two main update terms for each parameter: the first moment (mean of gradients) and the second moment (variance of gradients):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

where, m_t and v_t are the first and second moment estimates at time t, β_1 and β_2 are decay rates, and g_t is the gradient at round t. The model update is then given by:

$$w^{t+1} = w^t - \frac{\eta}{\sqrt{v_t+\epsilon}} m_t$$

A weighted average method is utilized by the central server in order to compile the updates that are received from each client. FedAdam is able to deliver smoother convergence in addition to decreasing fluctuations in updates and making it resistant against different client data distributions. This is accomplished by the integration of momentum and adaptive learning rates.

3.5 Federated yogi (FedYogi)

FedYogi is an adaption of the Yogi optimizer. It adjusts learning rates based on accumulated gradients in a manner that is comparable to FedAdam, but it modifies the variance updating rule. Due to the fact that FedYogi takes a novel method to managing the second moment of gradients, it is especially stable in situations that contain a great deal of heterogeneous data. Yogi adjusts learning rates in a more conservative manner based on the gradient history, as opposed to Adam, who constantly increases or decreases them. This allows Yogi to prevent severe learning rate variations in conditions that are not iid. FedYogi's update rule for the second moment term vt is slightly different from Adam's:

$$v_t = v_{t-1} - (1 - \beta_2) \cdot sign(v_{t-1} - g_t^2) \cdot g_t^2$$

By scaling the variance term according to the direction of the gradient, this adjustment prevents the phenomenon of over-accumulation in the variance term. The following formula is then used to compute the update for each parameter w:

$$w^{t+1} = w^t - \frac{\eta}{\sqrt{v_t+\epsilon}} m_t$$

where, m_t is the first moment estimate and η is the learning rate. FedYogi is advantageous in FL settings with high client variability because it stabilizes learning rates over time without drastically reducing them, which can be crucial for convergence in non-iid data environments.

4. PROPOSED METHODOLOGY

The proposed methodology begins with loading the malenoma dataset, followed by a comprehensive pre-processing step to prepare the data for FL. Pre-processing involves resizing all images to a uniform dimension of 112×112112, converting them into tensors, and normalizing the image data to ensure uniformity across the dataset. The dataset is then split into training and testing subsets using an 80%-20% ratio. The training data is distributed among a predefined number of clients, simulating the FL environment. Multiple federated averaging techniques, including FedAvg, FedProx, FedAdagrad, FedAdam, and FedYogi, are implemented to evaluate the performance of these approaches. Additionally, a hybrid averaging technique combining FedAvg and FedProx is proposed to address challenges like data heterogeneity and communication efficiency. Then deep learning models such as CNN and VGG16 are utilized for classification of malignant and non-malignant images. Finally, the models are evaluated through performance metrics, including accuracy and loss, to analyze the effectiveness of the applied techniques. This approach systematically combines traditional and advanced FL strategies to improve model performance in distributed data settings. The detail system architecture diagram is shown in Figure 3, and detail steps are explained below.

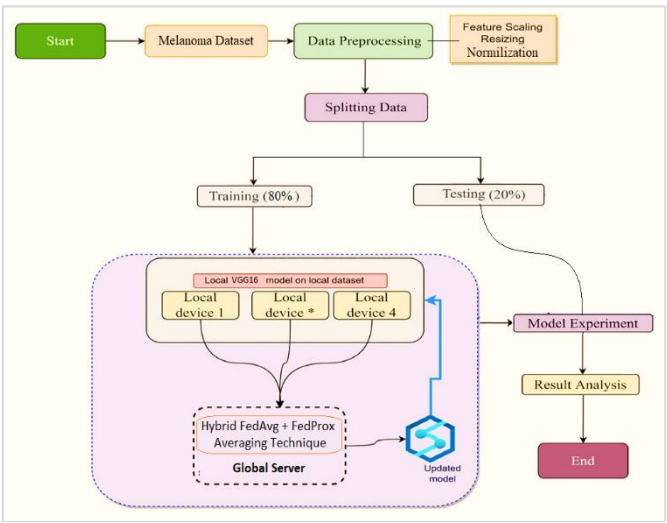

Figure 3. Proposed FL-based melanoma classification architecture diagram

4.1 Input dataset

The ISIC 2018 dataset is a widely recognized resource for skin lesion analysis, specifically aimed at advancing melanoma detection. The dataset supports three primary tasks: Lesion Segmentation, Lesion Attribute Detection, and Disease Classification [24]. These tasks are designed to enhance the accuracy and reliability of automated systems in diagnosing skin diseases, particularly melanoma. The dataset comprises 10,015 dermoscopic images representing seven distinct categories of skin diseases. For the purposes of this study, we focus on two key classes: Melanoma and Benign lesions. The third task, Disease Classification, is particularly relevant to our work, as it seeks to improve the automated prediction of disease categories in dermoscopic images. By leveraging this dataset, we aim to contribute to the development of more robust and accurate classification models for early and precise melanoma detection. The image sample of dataset are shown in Figure 4.

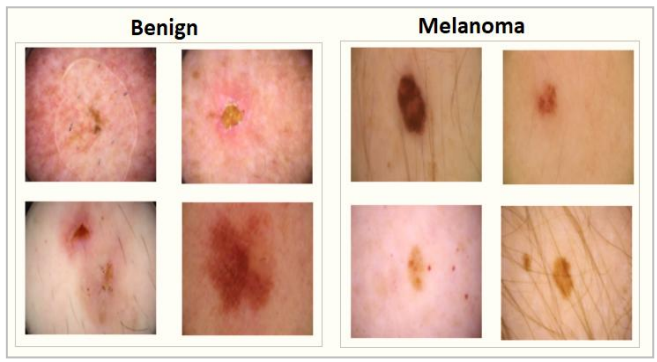

Figure 4. Melanoma dataset sample

4.2 Dataset pre-processing

Pre-processing is a crucial step to ensure that the melanoma images are correctly formatted, standardized, and ready for input into a deep learning model. The following steps describe the pre-processing pipeline used for melanoma image analysis:

4.2.1 Image resizing (112×112 pixels)

To standardize the input data and ensure compatibility with deep learning models, the first step involves resizing the original images to a fixed dimension of 112×112 pixels. This step is necessary because images in the dataset can vary in size, and deep learning models require inputs of a consistent shape. The resizing process uses interpolation methods, such as bilinear interpolation, to preserve the aspect ratio and prevent distortion, ensuring that important image features remain intact. The choice of a 112x112 resolution is a compromise between retaining sufficient image detail and reducing the computational complexity, allowing the model to process the images efficiently.

4.2.2 Convert to tensor

Once the images have been resized, they need to be converted into a format suitable for deep learning models. In this step, the images are transformed from their original format (such as a NumPy array or a PIL image) into a PyTorch tensor using the 'ToTensor' function. This conversion changes the structure of the image from a 2D array (height x width) with 3 color channels (RGB) to a 3D tensor with the shape 'channels, height, width'. Additionally, the pixel values, which originally range from 0 to 255, are scaled to a range of 0.0 to 1.0. This scaling is done automatically during the conversion to tensor, ensuring that the neural network can process the pixel values in a standardized range, which aids in more efficient learning.

4.2.3 Image normalization

Normalization adjusts the pixel values to have a mean of zero and a standard deviation of one. This is typically achieved by subtracting the dataset's mean and dividing by its standard

deviation for each color channel (R, G, B). This step helps to center the data around zero, ensuring that all input features (i.e., pixel values) are on a similar scale. By performing normalization, the model is able to train more efficiently, as it prevents certain features from dominating the learning process due to larger numerical ranges.

4.2.4 Select number of clients

Once the dataset has been pre-processed the next step in a FL workflow is to determine how many clients will participate in the training. This process is important because, in FL, training occurs across multiple devices or nodes (clients), with each client using its own local data for training without sending raw data to a central server. Instead, the clients send model updates (such as weights or gradients) to the server, which then aggregates these updates to improve the global model.

4.3 Perform train-test split (80%-20%)

Before starting the FL process, it is essential to split the data into two subsets: one for training and one for testing. This ensures that the model is trained on one portion of the data and validated on a separate, unseen portion to evaluate its performance.

4.3.1 80% for training

80% of the available data is used for training the model. This portion is distributed across the selected clients in the FL setup, where each client trains the model on its local data. The local updates are then sent to the central server for aggregation.

4.3.2 20% for testing

20% of the data is reserved for testing the model. This data is not used during training but is essential for evaluating the final performance of the trained model after the FL process.

The purpose of this split is to ensure that the model generalizes well on unseen data and does not overfit to the training data.

4.4 Distribute data into clients

Once the data has been split into training and testing sets, the training data needs to be distributed among the clients participating in the FL process. This step is important because FL allows clients to work on local data without directly sharing it, thus ensuring data privacy.

4.4.1 Data distribution

The training data is distributed across a set of clients, where each client holds a portion of the data. Each client trains the model independently on its local dataset and shares model updates (such as gradients or weights) with the central server.

4.5 Apply deep learning model

The next step is to apply a deep learning model to the FL setup. Here, CNNs and VGG16 models are commonly used for image classification tasks, such as melanoma detection.

- CNN Architecture: Convolutional Neural Networks are widely used for image classification tasks. They consist of multiple layers of convolutions, pooling, and fully connected layers to learn hierarchical features from images. CNNs are particularly suited for image processing due to their ability to capture spatial hierarchies of features in images.
- Federated Training: Each client in FL will use its share of the data to train its local CNN, then communicate the model modifications to the central server.
- VGG16 Architecture: VGG16 is a deep convolutional neural network model that consists of 16 layers with very small convolution filters (3×3) and uses max-pooling layers to downsample the image. It has proven to be highly effective in various image classification tasks, making it a suitable model for FL tasks like melanoma detection.
- FL Setup: Just like CNN, VGG16 will be trained on each client's local data, with model updates shared back to the central server. VGG16's deeper architecture can capture more complex features in the images, which is useful for detecting intricate patterns like those found in melanoma lesions.

4.6 Select federated averaging techniques

FL typically uses techniques to aggregate model updates from the clients and update the global model. The Federated Averaging (FedAvg) algorithm is the most commonly used technique for this purpose, but other variants like FedProx, FedAdagrad, FedAdam, and FedYogi can be explored to improve training performance, especially in cases of non-IID (non-independent and identically distributed) data.

4.6.1 FedAvg (Federated averaging)

FedAvg is the most common FL algorithm. It involves averaging the model weights sent by each client to update the global model. Each client performs local training for several epochs, and then the local models are averaged to form a global model. The process is repeated over multiple rounds.

4.6.2 FedProx (Federated proximal)

FedProx is a modification of FedAvg designed to address challenges when clients' data distributions are heterogeneous (non-IID data). It introduces a proximal term in the optimization process to mitigate the effect of highly skewed or non-IID data.

4.6.3 FedAdagrad (Federated adagrad)

FedAdagrad adapts the learning rate for each parameter based on the past gradients, which can help improve training in environments where there are sparse updates or highly variable data. It adjusts learning rates locally for each client, enhancing efficiency.

4.6.4 FedAdam (Federated adam)

FedAdam is based on the Adam optimizer, which adjusts the learning rate using both first and second moments of the gradients. It is particularly useful for models that have large amounts of data or require more sophisticated updates to improve convergence.

4.6.5 FedYogi (Federated yogi)

FedYogi is an extension of the Adam optimizer and is designed to handle the challenges of non-convex optimization, which often occurs in deep learning models. It adjusts the learning rate dynamically for each parameter and is robust to noisy data.

4.7 Proposed federated averaging technique (FedAvg + FedProx)

The proposed federated averaging technique (CWFL) combines the strengths of FedAvg and FedProx. The idea behind this hybrid approach is to leverage the simplicity and effectiveness of FedAvg while incorporating the robustness of FedProx to handle non-IID data.

In this combined approach, the basic FedAvg procedure is augmented with the proximal term from FedProx. This ensures that the updates from clients are more aligned with the global model, helping mitigate issues related to data heterogeneity while maintaining the efficiency of FedAvg.

4.7.1 Performance analysis

Once the FL model has been trained, the next step is to evaluate its performance using various metrics. Performance parameters such as accuracy, loss, accuracy and loss curve are used.

4.7.2 Algorithm: Hybrid averaging approach algorithm (FedAvg + FedProx)

This algorithm combines FedAvg and FedProx to handle heterogeneity in client data and ensure stable convergence while aggregating model parameters.

Input:

- N: Number of clients
- w_t^{global}: Global model parameters at round t
- w_t^k: Local model parameters for client k
- D_k: Size of the dataset for client k
- μ: Proximal constant (hyperparameter controlling regularization)
- D_k: Local dataset for client k
- K: Number of communication rounds
- E: Number of local epochs (each client trains on its data for E epochs)
- η: Learning rate for local training

Output:

- Updated global model parameters w_{t+1}^{global}

Algorithm:

1. Initialize:

Set global model parameters w_0^{global} (initialization can be random or pre-trained).

For each round $t = 0,1,2, \ldots K-1$

a. **Client selection**: Select a subset of clients $S_t \subseteq \{1,2, \ldots, N\}$ for this round.

b. **Local Model Updates** (on each selected client): For each client in $k \in S_t$, perform the following:

2. Train locally:

Perform local training on client k data D_k for E epochs using the local optimizer:

$$w_t^k = Train\ (w_t^k, D_k, \eta)$$

3. Compute local update:

Compute the difference between the local model and the global model parameters:

$$\Delta w_t^k = w_t^k - w_t^{global}$$

4. Apply proximal regularization (FedProx):

Apply the proximal term regularization to the local update:

$$w_t^k = w_t^{global} + (1-\mu)\Delta w_t^k$$

The term μ helps in regularizing the local update, encouraging the local models to stay close to the global model parameters.

5. Server aggregation (FedAvg + FedProx):

Aggregate the updated local models w_t^k from all selected clients S_t to compute the global model parameters:

$$w_{t+1}^{global} = \frac{1}{|S_t|}\sum_{k \epsilon S_t} w_t^k$$

The aggregation is done via simple averaging, but the local updates have been regularized through the proximal term.

6. Return:

After Krounds, the final global model w_K^{global} parameters are returned.

5. RESULT ANALYSIS

In this section, we outline the dataset and evaluation metrics utilized in this study, providing a detailed explanation of their relevance and application. We also present and analyze the experimental results of the proposed architecture, discussing its performance, key insights, and implications for the research objectives.

5.1 Experimental setup

For the experiment, the development environment was configured using a Jupyter notebook, which provided an interactive platform for coding and experimentation, having additionally, 113 GB of Google Drive storage, 13 GB of RAM and 15 GB of GPU RAM. The flower framework and FL models was implemented, trained, and validated using the TensorFlow backend alongside the Keras 2.4.3 framework, both of which were instrumental in streamlining the development and evaluation processes. This setup ensured efficient handling of the computational demands associated with training and validating the model.

5.2 Performance analysis

In evaluating the performance of classification models, various metrics are employed to assess overall effectiveness. In this study, we have used accuracy, precision, recall, F1-score, loss, and the confusion matrix to evaluate the models. Below are the performance parameters formulas;

$$Accuracy = \frac{Number\ of\ Correctly\ Classified\ Sample}{Total\ Number\ of\ Sample}$$

$$Precision = \frac{Number\ of\ True\ Positive\ Sample}{Number\ of\ TP + Number\ of\ FP}$$

$$Recall = \frac{Number\ of\ TP}{Number\ of\ TP + Number\ of\ FN}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where,

$$TP = True\ Positive\ Sample$$
$$TN = True\ Negitive\ Sample$$
$$FP = False\ Positive\ Sample$$
$$FN = False\ Negitive\ Sample$$

In the result section, we compared various averaging techniques applied to both CNN and VGG16 models. To ensure the reliability of the results, each model was independently trained and evaluated five times using different random initializations. The average test accuracy and final loss across the five runs were computed and compared against baseline models to assess performance improvements. The hyper parameters used for performing the experiment with various averaging techniques are listed in Table 1. This table includes both the FL strategy parameters and the additional parameters related to the TensorFlow model architecture, optimizer, and layers. It provides a comprehensive overview of the complete set of hyper parameters used in the simulation and model training.

Table 1. Hyper parameters

Hyper Parameters	Value	Description
Number of Clients	3	Total number of clients participating in the simulation.
Batch Size	16	Batch size for training the model on each client.
Image Size	(224, 224)	Input size for images, as required by CNN and VGG16.
Layers	Conv2D, MaxPooling2D, Flatten, Dense	Convolutional layers with ReLU activation, pooling, and dense layers for classification.
Optimizer	adam	Adam optimizer used for training the model.
Loss Function	binary_crossentropy	Binary crossentropy used as the loss function for binary classification.
Filters	32 / 64 / 128	Number of filters in the convolutional layer.
Kernel Size	(3,3)	Kernel size of the convolutional layer.
Pool Size	(2,2)	Pooling size of the max-pooling layer.
Activation	'relu' / Sigmoid	ReLU activation function for the dense layer and Sigmoid for output layer.
Epoch	10	Number of iterations for which models are run.

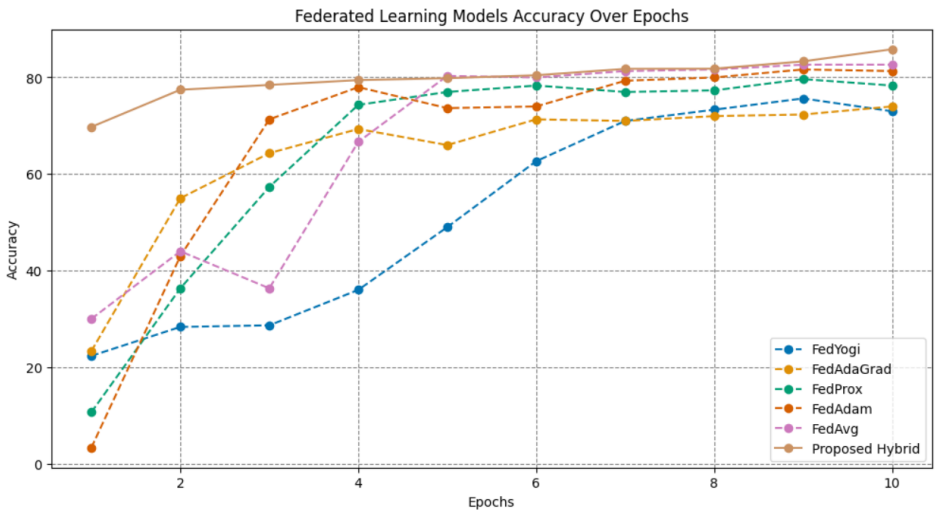

Figure 5. Accuracy comparison of averaging technique using CNN model

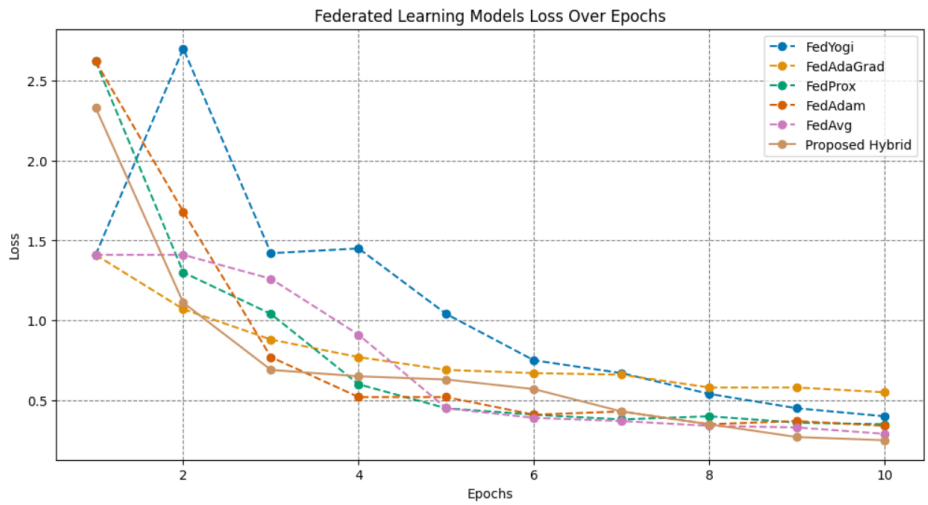

Figure 6. Loss comparison of averaging technique using CNN model

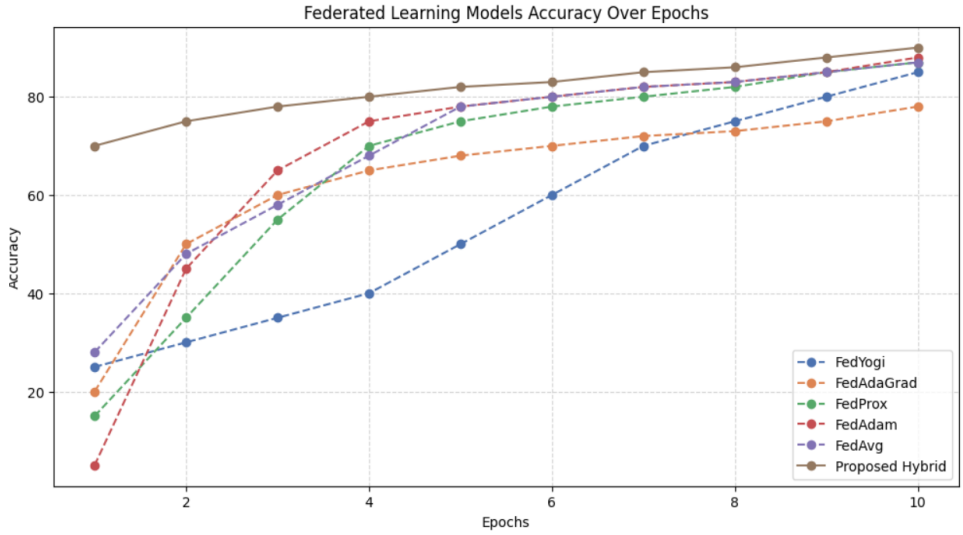

Figure 7. Accuracy comparison of averaging technique using VGG16 model

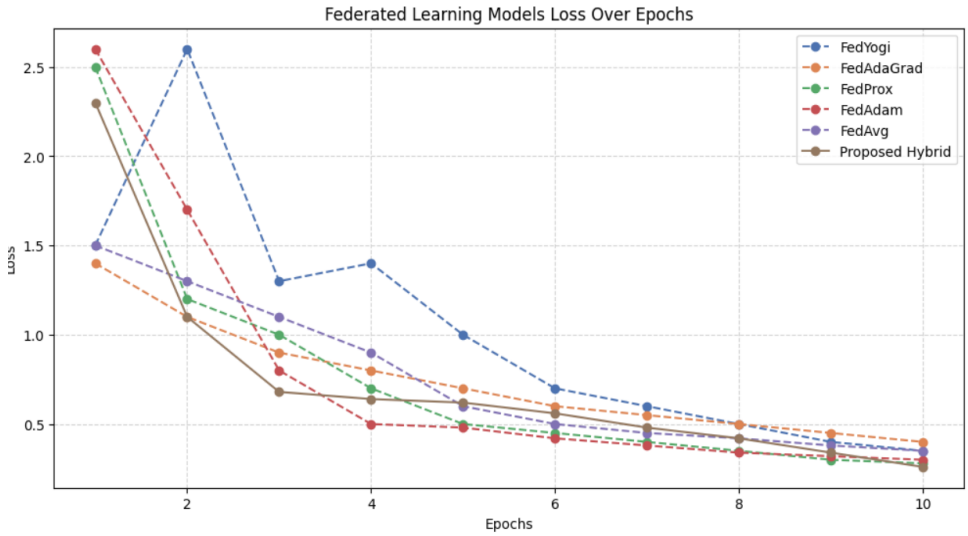

Figure 8. Loss comparison of averaging technique using VGG16 model

The accuracy and loss curves for the CNN and VGG16 models across various averaging techniques are shown in Figures 5-8. Figures 5 and 6 show the accuracy and loss curve of averaging techniques using CNN model, while Figures 7 and 8 show the accuracy and loss curve of averaging techniques using VGG16 model.

Figure 9 provides a comparative analysis of our experimental results. The VGG16 model with the proposed hybrid approach, combining FedAVG (Federated Averaging) and FedProx (Federated Proximal), outperformed all other models in terms of global accuracy. Specifically, this hybrid model achieved an impressive 90% accuracy, significantly higher than the other techniques tested. Additionally, the loss associated with this model was considerably low, reaching approximately 0.25, indicating better generalization and less overfitting. This suggests that the hybrid approach not only enhanced the model's ability to learn from decentralized data but also helped improve its convergence, resulting in a more robust and accurate performance compared to traditional averaging methods.

These results demonstrate the effectiveness of combining FedAvg and FedProx in FL settings, particularly when deployed with a powerful architecture like VGG16. The results of the proposed model demonstrate that strong performance can be achieved by utilizing the proposed architecture for melanoma image classification datasets. This indicates that the model is capable of generalizing well to a wide range of FL classification problems. Figure 10 shows the Confusion Matrix of CFWL with VGG16 model.

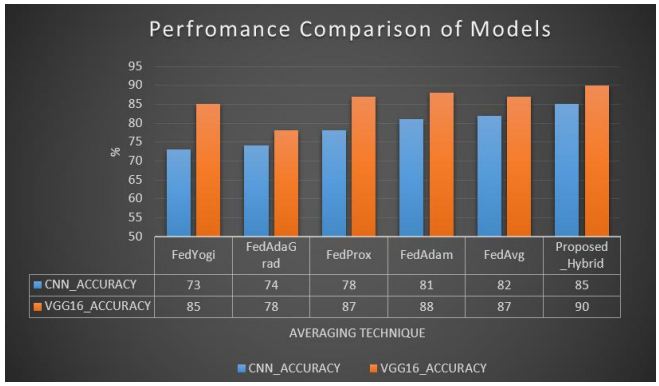

Figure 9. Comparative analysis of models

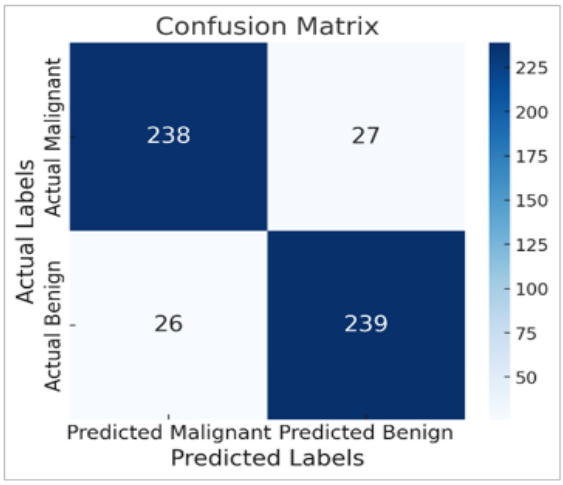

Figure 10. Confusion matrix

6. CONCLUSION

This study underscores the critical importance of timely and accurate skin cancer diagnosis, a potentially life-threatening condition, while addressing the significant challenges posed by traditional machine learning approaches in healthcare, such as data privacy concerns and the management of distributed, non-IID datasets. To overcome these challenges, we proposed the CFWL model for melanoma detection, which leverages FL to ensure privacy preservation while enhancing the accuracy and convergence of the central model. By integrating advanced deep learning architectures, specifically CNN and VGG16, our model demonstrated superior performance in feature extraction and classification tasks under non-IID conditions.

Experimental results revealed that our (CFWL) model, particularly when paired with the VGG architecture, outperformed existing FL averaging techniques such as FedAvg, FedProx, FedAdagrad, FedAdam, and FedYogi in terms of classification accuracy, loss reduction, and overall robustness. Notably, the model achieved exceptional accuracy on the ISIC dataset, correctly classifying 238 out of 265 malignant images and 239 out of 265 benign images. Furthermore, FL experiments demonstrated an impressive global accuracy of up to 90.0% in multi-client scenarios, highlighting the model's scalability and effectiveness in privacy-preserving environments.

This work makes a significant contribution to the fields of medical imaging and AI-driven healthcare by showcasing the potential of FL combined with advanced deep learning techniques to deliver scalable, privacy-preserving solutions for melanoma detection. The findings pave the way for future research and practical implementations, enabling the adoption of AI technologies in a manner that prioritizes both accuracy and patient privacy.

In future work, we aim to address several real-world challenges commonly encountered in FL environments. These include handling client dropout, where devices may intermittently disconnect during training, and managing communication costs, which can become significant in bandwidth-constrained settings. Enhancements such as robust aggregation methods to tolerate partial client participation, model compression techniques to reduce communication overhead, and adaptive client selection strategies will be explored. Incorporating these improvements will make the proposed hybrid FedAvg/FedProx approach more practical and effective for deployment in real-world scenarios.

REFERENCES

[1] Naeem, A., Anees, T., Fiza, M., Naqvi, R.A., Lee, S.W. (2022). SCDNet: A deep learning-based framework for the multiclassification of skin cancer using dermoscopy images. Sensors, 22(15): 5652. https://doi.org/10.3390/s22155652

[2] Cives, M., Mannavola, F., Lospalluti, L., Sergi, M.C., Cazzato, G., Filoni, E., Tucci, M. (2020). Non-melanoma skin cancers: Biological and clinical features. International Journal of Molecular Sciences, 21(15): 5394. https://doi.org/10.3390/ijms21155394

[3] Afza, F., Sharif, M., Khan, M.A., Tariq, U., Yong, H.S., Cha, J. (2022). Multiclass skin lesion classification using hybrid deep features selection and extreme learning machine. Sensors, 22(3): 799. https://doi.org/10.3390/s22030799

[4] Naeem, A., Farooq, M.S., Khelifi, A., Abid, A. (2020). Malignant melanoma classification using deep learning: Datasets, performance measurements, challenges and opportunities. IEEE Access, 8: 110575-110597. https://doi.org/10.1109/ACCESS.2020.3001507

[5] Huang, H.Y., Hsiao, Y.P., Mukundan, A., Tsao, Y.M., Chang, W.Y., Wang, H.C. (2023). Classification of skin cancer using novel hyperspectral imaging engineering via YOLOv5. Journal of Clinical Medicine, 12(3): 1134. https://doi.org/10.3390/jcm12031134

[6] Yaqoob, M.M., Alsulami, M., Khan, M.A., Alsadie, D., Saudagar, A.K.J., AlKhathami, M., Khattak, U.F. (2023). Symmetry in privacy-based healthcare: A review of skin cancer detection and classification using federated learning. Symmetry, 15(7): 1369. https://doi.org/10.3390/sym15071369

[7] Rivera, D., Grijalva, F., Acurio, B.A.A., Álvarez, R. (2019). Towards a mobile and fast melanoma detection system. In 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1-6. https://doi.org/10.1109/LA-CCI47412.2019.9037058

[8] Yaqoob, M.M., Nazir, M., Yousafzai, A., Khan, M.A., Shaikh, A.A., Algarni, A.D., Elmannai, H. (2022). Modified artificial bee colony based feature optimized federated learning for heart disease diagnosis in healthcare. Applied Sciences, 12(23): 12080. https://doi.org/10.3390/app122312080

[9] Aledhari, M., Razzak, R., Parizi, R.M., Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access, 8: 140699-140725. https://doi.org/10.1109/ACCESS.2020.3013541

[10] Abdul Salam, M., Taha, S., Ramadan, M. (2021). COVID-19 detection using federated machine learning. PloS ONE, 16(6): e0252573. https://doi.org/10.1371/journal.pone.0252573

[11] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 1-11. https://doi.org/10.1145/3338501.3357370

[12] Süzen, A.A., Şimşek, M.A. (2020). A novel approach to

machine learning application to protection privacy data in healthcare: Federated learning. Namık Kemal Tıp Dergisi, 8(1): 22-30. https://doi.org/10.37696/nkmj.660762

[13] Brisimi, T.S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I.C., Shi, W. (2018). Federated learning of predictive models from federated electronic health records. International Journal of Medical Informatics, 112: 59-67. https://doi.org/10.1016/j.ijmedinf.2018.01.007

[14] Zhou, Y., Koyuncu, C., Lu, C., Grobholz, R., Katz, I., Madabhushi, A., Janowczyk, A. (2023). Multi-site cross-organ calibrated deep learning (MuSClD): Automated diagnosis of non-melanoma skin cancer. Medical Image Analysis, 84: 102702. https://doi.org/10.1016/j.media.2022.102702

[15] Yu, L., Chen, H., Dou, Q., Qin, J., Heng, P.A. (2016). Automated melanoma recognition in dermoscopy images via very deep residual networks. IEEE Transactions on Medical Imaging, 36(4): 994-1004. https://doi.org/10.1109/TMI.2016.2642839

[16] Jayapriya, K., Jacob, I.J. (2020). Hybrid fully convolutional networks-based skin lesion segmentation and melanoma detection using deep feature. International Journal of Imaging Systems and Technology, 30(2): 348-357. https://doi.org/10.1002/ima.22377

[17] Majtner, T., Yildirim-Yayilgan, S., Hardeberg, J.Y. (2019). Optimised deep learning features for improved melanoma detection. Multimedia Tools and Applications, 78: 11883-11903. https://doi.org/10.1007/s11042-018-6734-6

[18] Valle, E., Fornaciali, M., Menegola, A., Tavares, J., Bittencourt, F.V., Li, L.T., Avila, S. (2017). Data, depth, and design: Learning reliable models for melanoma screening. arXiv preprint arXiv:1711.00441, 1(1): 770-778.

[19] Menegola, A., Fornaciali, M., Pires, R., Bittencourt, F. V., Avila, S., Valle, E. (2017). Knowledge transfer for melanoma screening with deep learning. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pp. 297-300. https://doi.org/10.1109/ISBI.2017.7950523

[20] Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A., Smith, J.R. (2015). Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images. In International Workshop on Machine Learning in Medical Imaging, pp. 118-126. https://doi.org/10.1007/978-3-319-24888-2_15

[21] Omran, A.H., Mohammed, S.Y., Aljanabi, M. (2023). Detecting data poisoning attacks in federated learning for healthcare applications using deep learning. Iraqi Journal for Computer Science and Mathematics, 4(4): 225-237. https://doi.org/10.52866/ ijcsm.2023.04.04.018

[22] Riaz, S., Naeem, A., Malik, H., Naqvi, R.A., Loh, W.K. (2023). Federated and transfer learning methods for the classification of Melanoma and Nonmelanoma skin cancers: A prospective study. Sensors, 23(20): 8457. https://doi.org/10.3390/s23208457

[23] Bakopoulou, E., Tillman, B., Markopoulou, A. (2021). Fedpacket: A federated learning approach to mobile packet classification. IEEE Transactions on Mobile Computing, 21(10): 3609-3628. https://doi.org/10.1109/TMC.2021.3058627

[24] Ali, R., Hardie, R.C., De Silva, M.S., Kebede, T.M. (2019). Skin lesion segmentation and classification for ISIC 2018 by combining deep CNN and handcrafted features. arXiv preprint arXiv:1908.05730. https://doi.org/10.48550/arXiv.1908.05730