A FEDERATED LEARNING AIDED SYSTEM FOR CLASSIFYING CERVICAL CANCER USING PAP-SMEAR IMAGES

RAMIZA RUMAISA ALIYA
NAZIA SHEHNAZ JOYNAB
A.S.M. RAKIBUL HASAN

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

MARCH 2023

A FEDERATED LEARNING AIDED SYSTEM FOR CLASSIFYING CERVICAL CANCER USING PAP-SMEAR IMAGES

B.Sc. ENGINEERING THESIS

By

RAMIZA RUMAISA ALIYA
NAZIA SHEHNAZ JOYNAB
A.S.M. RAKIBUL HASAN

Approved as to style and content by the Examiners in March 2023:

____

Lt Col Muhammad Nazrul Islam, PhD Associate Professor of Computer Science and Engineering MIST, Dhaka, Bangladesh	Supervisor
Nafiz Imtiaz Khan Lecturer of Computer Science and Engineering MIST, Dhaka, Bangladesh	Co-Supervisor
Brigadier General Md Mahfuzul Karim Majumder, ndc, psc, te Head of the Computer Science and Engineering MIST, Dhaka, Bangladesh, Dhaka, Bangladesh	Head of The Department

Department of Computer Science and Engineering MIST, Dhaka

CANDIDATE'S DECLARATION

A FEDERATED LEARNING AIDED SYSTEM FOR CLASSIFYING CERVICAL CANCER USING PAP-SMEAR IMAGES

DECLARATION

We hereby declare that the study reported in this thesis entitled as above is our own original work and has not been submitted before anywhere for any degree or other purposes. Further we certify that the intellectual content of this thesis is the product of our own work and that all the assistance received in preparing this thesis and sources have been acknowledged and cited in the reference section.

RAMIZA RUMAISA ALIYA
Student No. 201914015

NAZIA SHEHNAZ JOYNAB
Student No. 201914032

A.S.M. RAKIBUL HASAN
Student No. 201914056

Department of Computer Science and Engineering MIST, Dhaka

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF NOTATIONS

FL	Federated Learning
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
CNN	Convolutional Neural Network
HGB	Histogram Gradient Boosting
$XGBoost$	Extreme Gradient Boosting

ABSTRACT

A FEDERATED LEARNING AIDED SYSTEM FOR CLASSIFYING CERVICAL CANCER USING PAP-SMEAR IMAGES

Cervical cancer is one of the most common causes of mortality in women worldwide, and there is a lack of effective screening programs in low-income countries for the detection and treatment of precancerous conditions. Classification of pap-smear test cervical cell images is crucial as it gives essential information for the diagnosis of malignant or precancerous lesions and thus helps in providing a proper diagnosis. Most of the existing methods require accumulating pap-smear test images of all patients in a centralized location for classification purposes. However, this procedure may hamper the privacy of patient data and creates data ownership issues. In this study, a novel convolutional neural network based federated learning system is introduced for achieving both the objectives of accurate image classification and data privacy. In the proposed FL system, the updates of the locally trained models get aggregated with an initially untrained global model in order to increase its performance. In traditional ML-based systems, the more the train data, the more efficiently the model performs, but in the proposed system, clients can participate remotely to train a robust model even with the disadvantage of possessing a small dataset. The proposed CNN-based FL architecture showed the test precision, recall, f1-score and accuracy of 89.58%, 88.47%, 87.83% and 88.46% respectively. Thus multiple hospitals across different countries can use the proposed system to train their local models with their private dataset without sharing it centrally, which eventually helps to build the central model of FL architecture with diverse datasets.

সারসংক্ষেপ

সার্ভিকাল ক্যান্সারের প্যাপ স্মিয়ার ইমেজ শ্রেণীবদ্ধ করার জন্য একটি ফেডারেটেড লার্নিং এইডেড সিস্টেম

জরায়ু মুখের ক্যান্সার বিশ্বব্যাপী মহিলাদের মৃত্যুর সবচেয়ে সাধারণ কারণগুলির মধ্যে একটি, যেখানে কম আয়ের দেশগুলিতে প্রাক-ক্যান্সারজনিত অবস্থার সনাক্তকরণ এবং চিকিৎসার জন্য কার্যকর স্ক্রিনিং প্রোগ্রামের অভাব রয়েছে। প্যাপ-স্মিয়ার টেস্ট সার্ভিকাল কোষের চিত্রগুলির শ্রেণীবিভাগ অত্যন্ত গুরুত্বপূর্ণ কারণ এটি ম্যালিগন্যান্ট বা প্রাক-ক্যানসারাস ক্ষত নির্ণয়ের জন্য প্রয়োজনীয় তথ্য দেয় এবং এইভাবে একটি সঠিক রোগ নির্ণয় প্রদানে সহায়তা করে। বিদ্যমান পদ্ধতিগুলির বেশিরভাগের জন্য শ্রেণীবিভাগের উদ্দেশ্যে একটি কেন্দ্রীভূত স্থানে সমস্ত রোগীর প্যাপ-স্মিয়ার পরীক্ষার চিত্র সংগ্রহ করা প্রয়োজন। এই পদ্ধতিটি রোগীর ডেটার গোপনীয়তাকে বাধাগ্রস্ত করতে পারে এবং ডেটা মালিকানার সমস্যা তৈরি করতে পারে। এই গবেষণায়, নির্ভুল ইমেজ শ্রেণীবিভাগ এবং ডেটা গোপনীয়তার উভয় উদ্দেশ্য অর্জনের জন্য একটি কনভোলিউশনাল নিউরাল নেটওয়ার্ক-ভিত্তিক ফেডারেটেড লার্নিং সিস্টেম চালু করা হয়েছে। প্রস্তাবিত এফএল সিস্টেমে, স্থানীয়ভাবে প্রশিক্ষিত মডেলগুলির আপডেটগুলি একটি প্রাথমিকভাবে অপ্রশিক্ষিত বৈশ্বিক মডেলের সাথে একত্রিত করা হয় যাতে এর কার্যকারিতা বাড়ানো যায়। প্রথাগত মেশিন লার্নিং সিস্টেমে, ট্রেনের ডেটা যত বেশি হবে, মডেলটি তত বেশি দক্ষতার সাথে সঞ্চালন করবে, কিন্তু প্রস্তাবিত সিস্টেমে, ক্লায়েন্টরা একটি শক্তিশালী মডেল প্রশিক্ষণের জন্য দূর থেকে অংশগ্রহণ করতে পারে এমনকি একটি ছোট ডেটাসেট থাকার অসুবিধা সত্ত্বেও। প্রস্তাবিত কনভোলিউশনাল নিউরাল নেটওয়ার্ক ফেডারেটেড লার্নিং আর্কিটেকচারে যথাক্রমে ৮৯.৫৮%, ৮৮.৮৭%, ৮৭.৮৩% এবং ৮৮.৪৬% পরীক্ষার নির্ভুলতা, প্রত্যাহার, এফ-১ স্কোর এবং যথার্থতা দেখানো হয়েছে। এইভাবে বিভিন্ন দেশের একাধিক হাসপাতাল তাদের স্থানীয় মডেলগুলিকে কেন্দ্রীয়ভাবে ভাগ না করে তাদের ব্যক্তিগত ডেটাসেটের সাথে প্রশিক্ষণের জন্য প্রস্তাবিত সিস্টেমটি ব্যবহার করতে পারে, যা শেষ পর্যন্ত বিভিন্ন ডেটাসেটের সাথে ফেডারেটেড লার্নিং আর্কিটেকচারের কেন্দ্রীয় মডেল তৈরি করতে সহায়তা করে।

ACKNOWLEDGEMENTS

We are thankful to Almighty Allah for His blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, Lt Col Muhammad Nazrul Islam, PhD, Supervisor, Associate Professor, Department of Computer Science Engineering, and co-supervisor Nafiz Imtiaz Khan, Lecturer, Department of Computer Science and Engineering, Military Institute of Science and Technology for their constant supervision, affectionate guidance and great encouragement and motivation. Their keen interest on the topic and valuable advices throughout the study was of great help in completing thesis. We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

CHAPTER 1

INTRODUCTION

This chapter puts forth a discussion on the background of the research, problem statement, objectives of the thesis, methodological overview, scope of the thesis and organization of the remaining chapters. It starts with describing the background of the research to give an insight into the problem statement and to demonstrate the objectives of the thesis. Then further details are introduced through the overview, scope and organization of the thesis.

1.1 Research Background

Cervical cancer is a type of cancer that develops in the cells of the lower portion of the uterus which connects to the cervix. The human papillomavirus (HPV), a sexually transmitted infection, is responsible for more than 95% cases of cervical cancer. It may also be caused due to smoking, a weak immune system, birth control pills, having many sexual partners, etc.

In 2020, there was a global estimation of 604,127 new cases and 341,831 deaths due to cervical cancer (Sung et al., 2021). With 569,847 new cases every year, cervical cancer has become the fourth most frequent disease afflicting women globally, after breast, colorectal, and lung cancers (Pimple & Mishra, 2022). In the context of Bangladesh, there are 58.9 million women, aged 15 years or older, are at risk of acquiring cervical cancer (Bruni L, 2021). Each year 4971 women die from cervical cancer while 8268 women are diagnosed with it (estimations for 2020) (Bruni L, 2021). Therefore the second most common malignancy in Bangladeshi women is cervical cancer (Bruni L, 2021).

Although no symptoms are visible during the early stages, some symptoms like vaginal bleeding, pelvic pain etc. are noticed later. Death due to cervical cancer can be reduced if effective screening strategies are implemented. Regular pap smear screening tests must be used to monitor women for early identification of cervical cancer so that effective treatment can be provided to the patients. For a proper diagnosis and the identification of malignant or precancerous lesions, it is crucial to classify cervical squamous cells according to their cytomorphology in Pap smear images (Plissiti et al., 2018a).

Existing machine learning approaches to classify cervical cell images combine two or more datasets in order to increase the model's performance. But such publicly available datasets are few and far between. In order to harness the benefit of data privacy with the development of an effective deep learning model, researchers have applied FL for brain tumor segmentation and breast cancer histopathological image classification. But there is no research conducted for developing a differential privacy-enhanced FL system for cervical cell classification.

1.2 Problem Statements

In most of the existing methods, various machine learning and deep learning techniques have been introduced which require accumulating the pap-smear test images from different data resources in a centralized location for classification purposes. The more training data, the higher the model's accuracy. In the medical imaging domain, acquiring sufficient data is a significant challenge. Though this challenge could be addressed through collaboration between multiple institutions, sharing medical data in a centralized location faces various legal, privacy, technical, and data-ownership challenges. And also the institutions in possession of low amount of data can't achieve a satisfying performance from models trained with the existing methods.

1.3 Thesis Objectives

The primary goals that we want to achieve with this research are as follows:

1. To propose a novel Deep Learning architecture for the classification of cervical cancer cell images considering small datasets.
2. To integrate the best performed Deep Learning model with the FL architecture by evaluating them continuously using hyper-parameter tuning.
3. To develop a user-friendly and interactive web app for applying the proposed Federated Learning architecture in a real-world environment.

1.4 Methodological Overview

In order to meet the desired objectives, a literature review was conducted to gather knowledge from the previous research done on the topic. The study helps to gain better knowledge on the scope and uses of federated learning and to get a proper insight as to how the classification of Cervical Cancer can be done. It helps to know about the procedure of merging the data from different hospitals using federated learning to get an improved classification, without sharing of private data. Keeping this in mind, Federated Learning architecture was applied to a novel CNN model for the classification of cancer cells.

1.5 Scope of the Thesis

The study particularly focuses on the development of a web application, through which clients from different places can add their data, which would eventually aid in getting an accurate classification of the cancer cells. Here the main target is to update the global model periodically with the help of a variety of data collected from different institutions. The collaborating institutions can benefit from this system in several ways.

1. Federated Learning allows individual hospitals to address the challenge of possessing small dataset by collaborating with multiple non-affiliated hospitals.

2. Federated Learning utilizes diverse datasets of numerous collaborators for building a robust deep learning model.

3. Federated Learning helps to classify the pap smear images of cervical cancer by aggregating locally trained model weights from different hospitals, referred to as "clients" to a centralized location model, known as the "server" model, without needing the clients to share their personal data.

1.6 Organization of Thesis

The remaining chapters are organized as such: The theoretical background and the related work have been discussed in chapter 2. Firstly, a detailed discussion has been put forth about Cervical Cancer. Then the existing methodologies have been discussed. Finally, a critical appraisal is given. In chapter 3, a detailed description was provided of fifteen different traditional ML algorithms, Deep Learning and the process of Federated Learning. Federated Learning training algorithms: FedSGD and FedAvg were also discussed. The methodology which was carried out in this research is presented in Chapter 4. The process of system development has been discussed in chapter 5. This chapter is described using 3 subsections. The first subsection is about the acquisition of the dataset, the second subsection presents how the classical machine learning models have been developed and the results obtained from those models were analyzed. The final subsection describes how the architecture of federated learning is built from data partitioning, augmentation, foundation model development to prototype development for applying FL in a real-world environment. Thus the main purpose of this whole chapter is to give a brief discussion about the overall design of the system including the development of the base model. The ending chapter states the limitations and future works of thesis.

CHAPTER 2

BACKGROUND THEORY AND RELATED WORK

This chapter is focused on the introduction of cervical cancer as well as previous research methodologies of detecting and classifying cervical cancer.

2.1 Cervical Cancer

Cervical cancer occurs within the cells of the cervix. The cervix is the slender end of the uterus that forms a connection between the uterus and the birth canal. Before the appearance of cancer in the cervix, the cells of the cervix undergo such a change, in which abnormal cells are found to appear in the cervical tissue. These abnormal cells may turn to cancer cells over time, if not destroyed or removed. The cancer cells then start to grow and spread deeper into the cervix, as well as the surrounding areas. The two main types of cervical cancer are Squamous cell carcinoma and Adenocarcinoma. To detect the type and to know how far cervical cancer has spread, cell images have been divided into 5 categories here. The categories considered are Dyskeratotic, Koilocytotic, Metaplastic, Parabasal, and Superficial-Intermediate. The Dyskeratotic cells are the type of squamous cells that underwent premature abnormal keratinization within individual or clusters of cells. The presence of dyskeratocytes in cervical smears may be predictive of either a simultaneous HPV infection or an infection the Koilocytotic cells are mostly the mature squamous cells, and Metaplastic cells are small or large parabasal cells with prominent cellular borders which may contain large intracellular vacuole, Parabasal cells are the immature squamous cells which are the smallest epithelial cells, Superficial-Intermediate cells show morphological changes which indicate more severe lesions. This classification would help to know better about the severity and thus offer a better treatment accordingly.

2.2 Related Works for Cervical Cancer Classification

For the purpose of detecting cervical cancer, researchers employed three distinct classifiers: Softmax regression (SR), Support vector machine (SVM), and GentleBoost ensemble of decision trees (GEDT) (Rehman, Ali, Taj, Sajid, & Karimov, 2020). Over a convolutional neural network, they suggested using these three to create an autonomous cervical cancer detection system. They came to the conclusion that, when compared to the other strategies stated, the proposed system appeared to perform better and produced the maximum performance.

For the first time, federated learning was presented in a study on the modality of cardiovascular magnetic resonance (CMR), with four centers derived from subsets of the MM and ACDC datasets, focusing on the diagnosis of hypertrophic cardiomyopathy (HCM) (Linardos, Kushibar, Walsh, Gkontra, & Lekadir, 2022). They modified a 3D-CNN network that had previously been trained on action recognition and investigated two approaches to incorporating shape prior information into the model, as well as four different data augmentation setups, systematically analyzing their impact on the various collaborative learning options. Despite the small sample size (180 subjects from four centers), they demonstrated that privacy-preserving federated learning achieves promising results that are competitive with traditional centralized learning. They also discovered that federatively trained models are more robust and less susceptible to domain shift effects.

Federated learning enabled deep learning model was used on multimodal brain scans (Sheller, Reina, Edwards, Martin, & Bakas, 2019). The quantitative results show that federated semantic segmentation models (Dice=0.852) perform similarly to models trained by sharing data (Dice=0.862). The comparison was shown among federated learning and two other collaborative learning methods and it concluded that these methods fall short of the performance of federated learning.

A Federated learning-based cancer diagnosis model was proposed where six first-level impact indicators were identified, as well as historical case data from cancer patients (Ma et al., 2022). In the federated learning framework combined with the convolutional neural network, various physical examination indicators of patients were used as input. An auxiliary diagnostic model was built using patients' recurrence time and location, and comparison algorithms included linear regression, support vector regression, Bayesian regression, gradient ascending the tree, and multilayer perceptrons neural network. CNN's federated prediction model based on improved accuracy under the condition of joint modeling and simulation on the five types of cancer data accuracy reached more than 90.

In a study conducted by Pati et al. (Pati et al., 2022), The most extensive real-world FL effort to develop an accurate and generalizable ML model for detecting glioblastoma subcompartment boundaries was described in a research. Notably, the study's collaborators' extensive global footprint yields the largest dataset ever reported in the literature assessing this rare disease. FL provided unprecedented access to the most common and fatal adult brain tumor dataset, as well as meaningful ML training to ensure model generalizability across out-of-sample data. Because FL enabled large and diverse data, the final consensus model outperformed the public initial model against both the collaborators' local validation data and the entire out-of-sample data.

In another study, the feasibility of using differential-privacy techniques was investigated to protect patient data in a federated learning setup (W. Li et al., 2019). They developed and tested practical federated learning systems for brain tumor segmentation on the BraTS dataset. The experimental results revealed that there is a tradeoff between model performance and privacy protection costs.

How data dispersion affects FL performance was outcome of a research (Adnan, Kalra, Cresswell, Taylor, & Tizhoosh, 2022). The two parts of their proposed system, bag preparation and Multiple-Instance Learning, were local to each client (MIL). The authors explored the possibility of learning from distributed medical data via differentially private federated learning. They mainly showed how FL might be utilized in clinical contexts to guarantee data privacy while also ensuring minimal performance reduction.

One study gave an outline of how federated artificial intelligence can be used in medical imaging applications while maintaining security and privacy (Kaissis, Makowski, Rückert, & Braren, 2020). They talked about how AI has changed the area of medicine, what is needed for the best privacy preservation, and the privacy and security concerns with medical imaging.

The study of Ghoneim, Muhammad, & Hossain showed the development of a cervical cancer categorization and detection method based on CNN (Ghoneim, Muhammad, & Hossain, 2020). Three CNN models and an ELM-based classifier were studied. The shallow CNN model was trained and tested using the 5-fold cross-validation method on the Herlev dataset from the database. They concluded by demonstrating how the ELM-based classifier produced a greater accuracy than all the other methods.

The use of hybrid pipelines for the detection and classification of aberrant regions in liquid-based cytology (LBC) pictures using a combination of deep learning (DL) and traditional machine learning (ML) techniques was demonstrated in another study (Silva, Sampaio, Teixeira, & Vasconcelos, 2021). They made use of a personal database containing 1920 x 2560 pixel photos. They demonstrated how to inspect cervical samples using a RetinaNet model for the detection of aberrant regions.

A pre-trained CNN architecture was used along with a support vector machine for the detection of Cervical Cancer (Taha, Dias, & Werghi, 2017). For the pre-trained architecture, the AlexNet was used to extract the desired features and showed how it performs better with better recall, precision, specificity and accuracy scores than other compared techniques.

In another research, the Herlev and SIPaKMeD datasets were integrated for the purpose of detecting cervical cancer. They demonstrated their ability to successfully analyze multi-layer cervical cells and developed a binary and multi-class classification pipeline to identify cancer in Pap smear images (Bhatt, Ganatra, & Kotecha, 2021).

A focused study included reviews of the various cervical cancer diagnostic techniques (Shanthi, Hareesha, & Kudva, 2022). They called attention to the flaws and limitations of the analytical techniques and procedures. They made the observation that subpar preprocessing and segmentation resulted in subpar classification outcomes.

The main discoveries made from the study include:

1. Except some recent works, very few such instances of incorporation of Federated Learning in the classification of cervical cancer cells are available.
2. Most previous attempts of classification are methods which require data sharing which eventually hampers privacy.
3. Most existing systems do not show concern for the development of a personalized Convolutional Neural Network model for the purpose of classification.

2.3 Critical Summary

In order to protect patient data privacy and produce reliable findings even with the drawback of limited data, federated learning is necessary in a variety of illness prediction and classification systems. This technique has already been used in the diagnosis of hypertrophic cardiomyopathy, the segmentation of brain tumors, the prediction of breast cancer, and many other things. A decentralized machine learning method called federated learning enables several parties to train a machine learning model without disclosing their personal information. This method has a great deal of potential in the healthcare industry, where data privacy is a major problem.

Without actually sending the data to a central server, federated learning can be utilized in the healthcare industry to train models on private patient information. This can support patient privacy protection while yet allowing for medical research and individualized care. Healthcare fields including disease prediction, medication research, and clinical decision-making can all benefit from federated learning.

The creation of a model to forecast diabetic retinopathy, the main cause of adult blindness, is one instance of federated learning in the healthcare industry. Without releasing the data directly, researchers trained the model using data from many hospitals using a federated learning approach. This method produced a very accurate model while preserving data privacy.

Ultimately, federated learning has the potential to transform healthcare by facilitating more individualized medical research and treatment while protecting patient privacy. The standardization of data across many hospitals and ensuring the stability and dependability of the federated learning models are two obstacles that must yet be overcome.

CHAPTER 3

THEORETICAL BACKGROUND

The theoretical background defines a variety of subjects, including traditional machine learning methods, deep learning methods, federated learning concepts for training the model to classify the cervical cell images and other relevant concepts.

3.1 Traditional Machine Learning Algorithms

3.1.1 LightGBM

Light Gradient Boosting Machine or, LightGBM is a highly efficient Gradient boosting decision tree, which provides increased accuracy and efficiency. It is a gradient boosting framework that utilizes tree based learning algorithms. It has the advantages like capability of handling larger scale data, better accuracy, decreased memory usage, etc. LightGBM splits the tree leaf-wise in order to ensure lower loss.

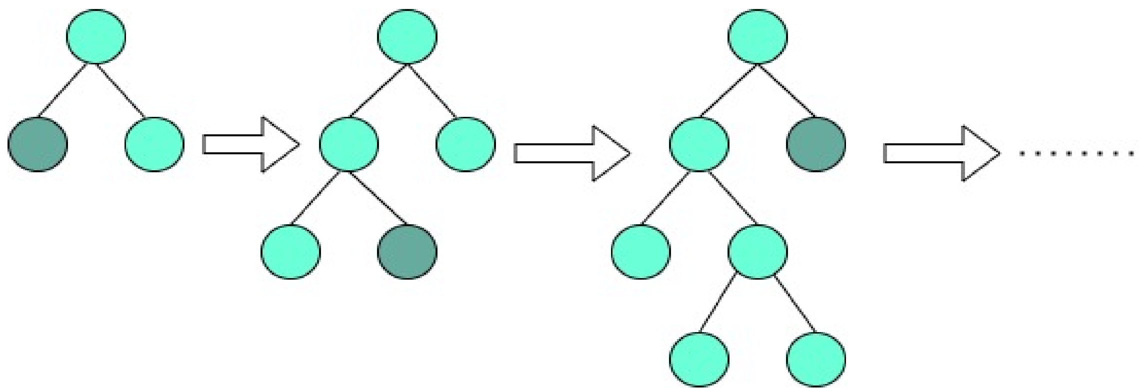


Figure 3.1: Leaf-wise tree growth used in LightGBM

The LightGBM algorithm utilizes two novel techniques called Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) which allow the algorithm to run faster while maintaining a high level of accuracy (Ke et al., 2017).

3.1.2 Histogram Gradient Boosting With LightGBM

In order to speed up the Gradient boosting decision tree (GDBT) process,using histogram proves to be an important technique (Tyree, Weinberger, Agrawal, & Paykin, 2011). To ensure better quality in less inference time, an advanced base learner called piecewise linear tree is utilized.

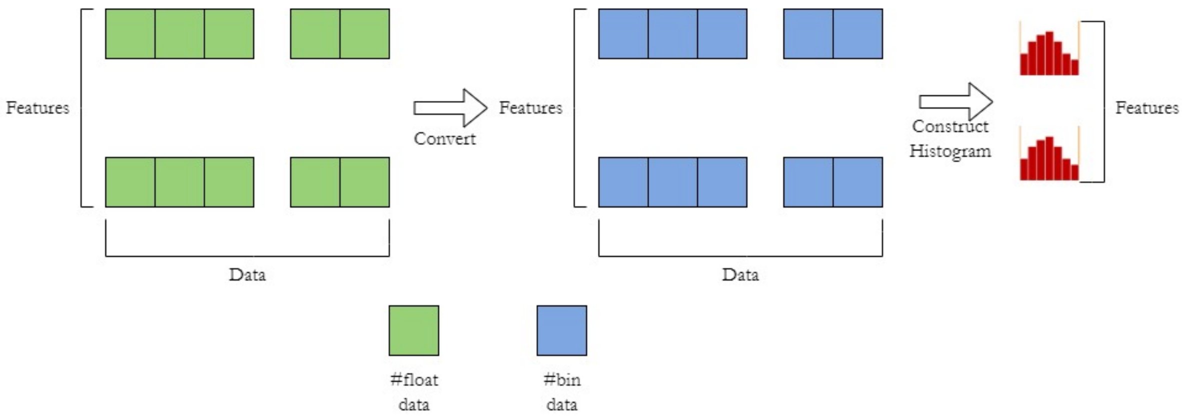


Figure 3.2: Histogram Algorithm of LightGBM

3.1.3 Histogram Gradient Boosting

Histogram Boosting was released by Scikit-Learn, which was named as HistGradientBoostingClassifier. Its operation is similar to that of LightGBM, but offers a much faster speed than gradient boosting. It provides built in support for the missing values, and is used for faster training of decision trees.

3.1.4 Xtra Trees

Extremely randomized trees or extra trees is an ensemble supervised machine learning method. The extra trees algorithm works by creating multiple decision trees, where the sampling for each tree is done randomly. Thus the dataset taken for each tree is made with unique samples. To output the classification result, it aggregates the results of the multiple de-correlated decision trees, which are collected in a ”forest”. Thus it holds similarity to the Random Forest Classifier, but differs with respect to the construction of the Decision Trees in the forest.

3.1.5 SVM

Support Vector Machine (SVM) is a machine learning method which helps to largely overcome the curse of dimensionality and other issues faced while applying the traditional learning methods. SVM takes the optimal separating hyperplane as the decision surface, which is able to classify two classes of data while maximizing the distance between the points (H. Li, Chung, & Wang, 2015). Thus all of the data points on one side of the hyperplane will represent one category and the data points on the other side of the hyperplane will represent a different category. Support vector machines are a set of supervised learning methods, that can be used to detect cancerous cells, on the basis of millions of cells.

3.1.6 SVM Grid Search

SVM has some hyper-parameters, and the optimal one can be found by creating a grid of hyper-parameters and just try all the possible combinations. This approach taken is termed as Grid Search. In the Grid Search method, all the possible combinations of hyperparameters will pass one by one into the model and check each model's score. As a result, it gives us a set of hyperparameters which give the best possible score. Grid Search calculates the error for various hyperparameter values, and eventually aids in choosing the best values.

3.1.7 Gradient Boosting

Gradient boosting is a machine learning technique that is used in both regression and classification tasks. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. It is termed as a sequential ensemble learning technique, as the performance of the model is found to improve over iterations. In this case a group of weak prediction models, like regression decision trees, are modeled by adding new learners in a sequential manner. It can give prediction results based on the decision nodes, and can provide an improved accuracy when viewed as an ensemble (Chakrabarty, Kundu, Dandapat, Sarkar, & Kole, 2019).

3.1.8 XGBOOST

Extreme Gradient Boosting or XGBoost also uses a gradient-boosting based approach in order to ensure optimization in the tree split against a predefined loss function. It is an ensemble learning method which combines the predictions of multiple weak models to produce a stronger prediction. XGBoost has the advantage that it can deal with any missing data, unlike other ML models that require additional processing on the training data. It does not require an immense set of pre-processing operations and thus minimizes the communication costs significantly and also reduces the leakage of private data (Ong, Zhou, Baracaldo, & Ludwig, 2020).

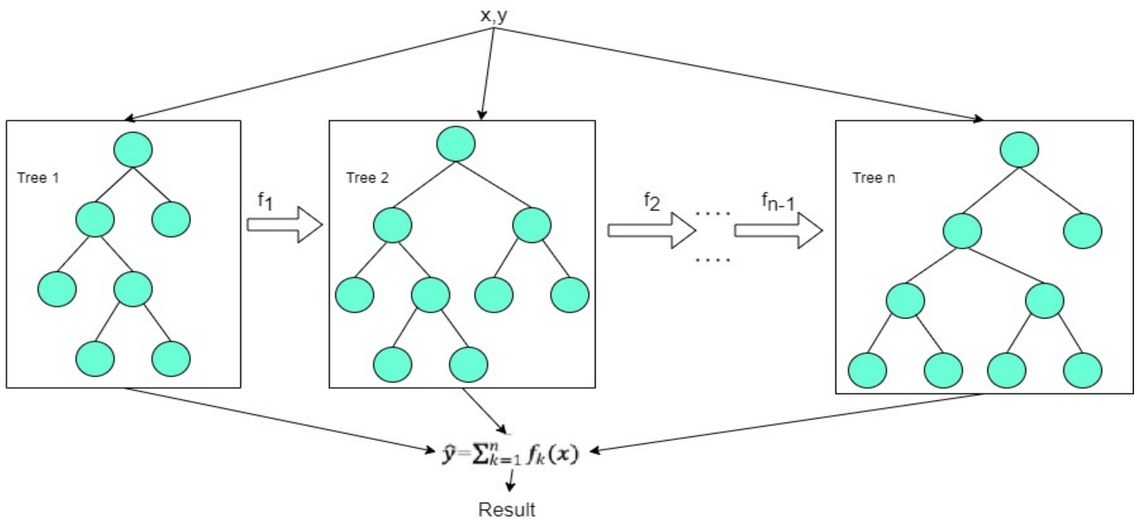


Figure 3.3: A General Architecture of XGBoost

XGBoost has a wide range of hyperparameters that can be adjusted to optimize performance, making it highly customizable and also provides feature importances, which helps in making important predictions.

3.1.9 MLP

A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN). The multilayer perceptron can be trained to approximate virtually any measurable function, as it does not make any prior assumption regarding the data distribution. The multilayer perceptron can model the non-linear functions, and can also be trained so that whenever presented with any unseen data, it can accurately generalise the information (Gardner & Dorling, 1998). The multilayer perceptron is a system of interconnected neurons, consisting of input and output layers, and one or more hidden layers.

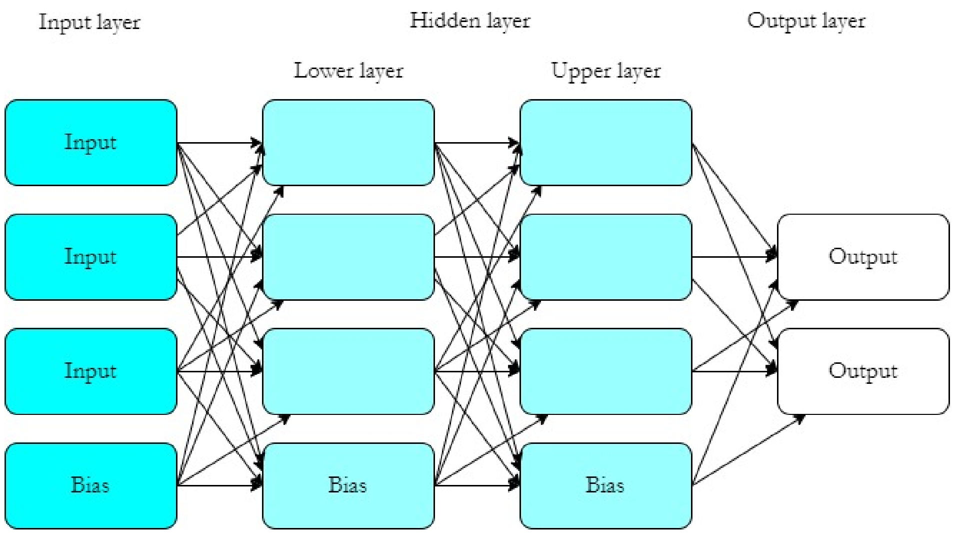


Figure 3.4: The multilayer perceptron

3.1.10 Histogram Gradient Boosting With XGBoost

Histograms can be applied to the Extreme Gradient Boosting, or XGBoost for the continuous input variables. It eventually helps to provide a highly optimized implementation of gradient boosting.

3.1.11 KNN

K-Nearest Neighbor or KNN Algorithm is a non-parametric, supervised learning classifier, which makes use of approximation to make classifications about the grouping of an individual data point. It can be used for both regression and classification, but it is mostly utilized in classification algorithms. The average of the k nearest neighbors is considered to make a prediction about the classification of a new data point. For this, the K-NN algorithm assumes the similarity between the new data and all the available data and places the new data into the category which is most similar to the available categories.

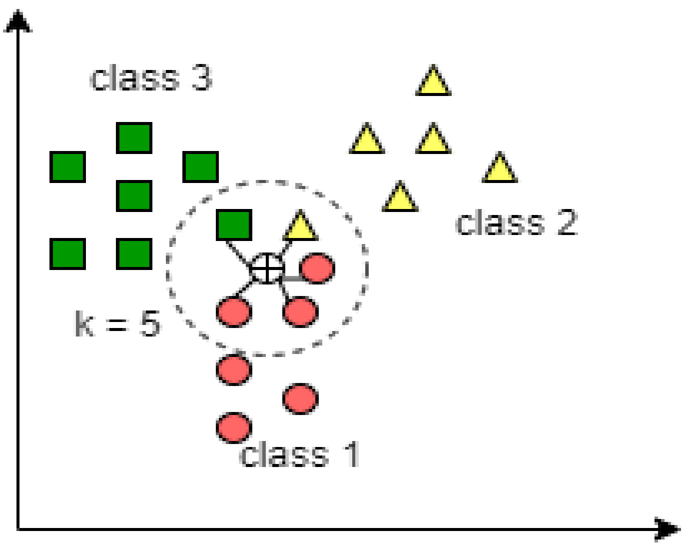


Figure 3.5: K Nearest Neighbours

KNN is a non-parametric algorithm, as it does not make any assumption regarding the underlying data. It is sometimes termed as a lazy-learner as it does not learn anything from the training set immediately. It merely stores the dataset and performs an action on it at the time of classification.

3.1.12 Decision Tree

A decision tree is a decision support tool that uses a tree-like model, made up of decisions and their possible consequences. An instance is classified by starting at the root node of the tree, then it gradually moves down until it reaches the leaf. The internal nodes denote a test on an attribute. Drawn from left to right, the contents of the leaf node represent the outcome of the decisions. The decision tree follows a non-parametric approach, which means that it is distribution-free and does not depend on the assumptions of probability distribution. It can operate on any high-dimensional data with exceptional accuracy.

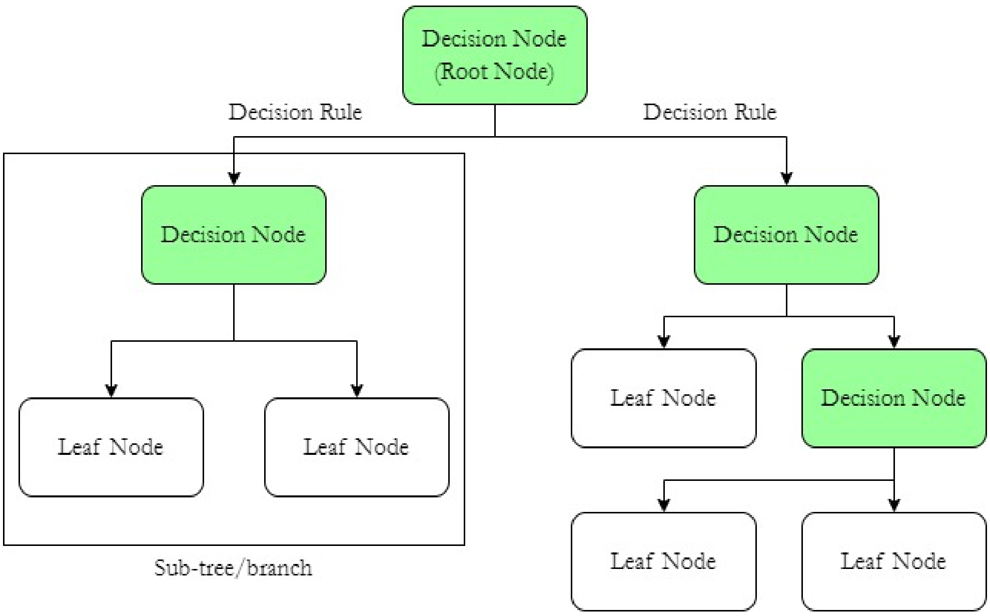


Figure 3.6: Decision Tree Classifier

3.1.13 AdaBoost

AdaBoost or Adaptive Boosting is a method in machine learning, which is used as an ensemble learning method. A strong model can be built by grouping multiple weak classifiers where each one gets to progressively learn from the others' wrongly classified objects. The multiple weaker models are independently trained and their predictions get combined to make the overall prediction.

Boosting is a machine learning technique used for both regression and classification problems. It produces a prediction model which holds characteristics which represent an ensemble of weak prediction models, typically decision trees. It builds the model in a gradual manner, like other boosting methods do. It generalizes the models by allowing optimization of an arbitrary differentiable loss function ((Pandey & Prabhakar, 2016)).

3.1.14 Random Forest

are ensemble learning techniques for classification, regression, and other problems that work by building a large number of decision trees during the training phase The class most trees choose in a classification problem is the output of the random forest.

3.1.15 Gaussian Naive Bayes

It is an extension of the Naive Bayes Algorithm. Gaussian Naive Bayes is a classification technique, which is used in Machine Learning (ML) based on the probabilistic approach and Gaussian distribution. This algorithm does not need much time for training and gives results which are greatly reliable.

3.2 Deep Learning

Artificial neural networks, a class of algorithms inspired by the structure and operation of the brain, are the focus of the machine learning discipline known as deep learning. Some of the data pre-processing that is generally involved with machine learning is eliminated with deep learning. These algorithms can handle text and visual data that is unstructured and automate feature extraction, reducing the need for human specialists.

The process of recognizing photographs and classifying them into one of several predetermined different categories is known as image recognition (or image classification). As a result, image recognition software and apps can identify the objects in a photo and differentiate one from another. Computer vision is the branch of study that aims to give machines this capability. Deep learning models can be used by researchers to solve computer vision problems.

Deep learning systems need access to enormous amounts of training data and processing power in order to attain an acceptable degree of accuracy. Until the age of big data and cloud computing, neither of these resources was readily available to programmers. Deep learning is able to produce precise predictive models from enormous amounts of unlabeled, unstructured data because it is capable of producing complicated statistical models directly from its own iterative output.

3.2.1 CNN

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, give importance (learnable weights and biases) to various aspects in the image, and be able to distinguish one from the other. In comparison to other classification methods, a ConvNet requires significantly less pre-processing. ConvNets have the capacity to learn these filters and properties, whereas in basic techniques filters are hand-engineered.

The structure of a ConvNet was influenced by the way the Visual Cortex is organized and is similar to the connectivity pattern of neurons in the human brain.

3.2.2 Layers of CNN

Convolutional Neural Networks (CNNs) are constructed from layers that perform different operations on the input data. The primary layers used to construct a CNN are as follows:

Convolutional layer

To perform the convolution process, a convolutional layer applies a number of learnable filters or kernels to the input image or feature map. A set of feature maps that represent different aspects of the input picture or feature map make up a convolutional layer's output.

Let K be the kernel detector and I be the image used as the input. The output feature map O can then be created by convolving the input image with the kernel and using an activation function, similar to what was done in 3.1:

$$O(i,j) = f(\sum\sum I(m,n) * K(i-m, j-n) \tag{3.1}$$

where f denotes the activation function, (i,j) denotes a location in the output feature map, and (m,n) denotes the location of the input image pixel that the kernel is now aligned with. The whole feature map can be retrieved by swiping the kernel over the full input image.

Activation Layer

An activation layer applies a non-linear activation function to the output of a fully connected layer or a convolutional layer.

Let X be the input (which is often the output of a convolutional or fully connected layer) and f be the activation function for the activation layer. Each element of the input X is subjected to the activation function by the activation layer, which results in the output Y. As seen below, the output Y is calculated:

$$Y(i,j,c) = f(X(i,j,c) \tag{3.2}$$

where i, j, and c are the input and output tensors' spatial and channel coordinates. Sigmoid, tanh, and ReLU (Rectified Linear Unit) activation functions are frequently used in CNNs. The precise requirements of the work and the type of data being processed dictate the activation function to be used.

Pooling Layer

A pooling layer reduces the spatial size of feature maps while retaining the most crucial data. The two most used pooling operations are average pooling, which chooses the average value from each patch of a feature map to produce a smaller map, and max pooling, which returns the maximum value inside a pooling window. The following mathematical diagram illustrates how pooling works: Let P represent the pooling size, S represent the stride, and F represent the input feature map. The output feature map O can then be created by pooling the input feature map with a pooling window of size P and a stride of S:

$$O(i,j) = Pooling(F(iS:iS+P, jS:jS+P) \tag{3.3}$$

where Pooling is a pooling function that produces a single value from values in the pooling window.

Batch Normalization Layer

The layer of batch normalization enables the network's layers to learn more independently. The output of the earlier layers is normalized using it. In normalization, the activations scale the input layer. Learning becomes more effective when batch normalization is utilized, and it can also be used as regularization to prevent model overfitting. To standardize the input or the output, the layer is added to the sequential model. It can be applied in a number of places between the model's layers. It is frequently positioned immediately after the convolution and pooling layers and after specifying the sequential model.

Dropout

The regularization method used to stop overfitting in the model is called dropouts. A certain percentage of the network's neurons are switched at random with the addition of dropouts. The incoming and outgoing connections to the neurons are also turned off when they are turned off. To help the model learn more effectively, this is done. Dropouts are typically advised against using them after convolution layers; instead, they should be utilized after the network's dense layers. It is always advisable to turn off the neurons just to 50%. There is a possibility that the model leaning and the forecasts would be poor if we turned off more than 50%.

Flatten Layer

A flattened layer transforms a 2D or 3D feature map into a 1D vector that can be used as input to fully connected layer. Let F be the input feature map of size H x W x C, where H is the height, W is the width, and C is the number of channels. Concatenating the rows of the feature map will then yield the flattened vector F':

$$F' = [F(1,1,1), F(1,2,1), ..., F(1,W,1), F(2,1,1), ..., F(H,W,1), F(1,1,2), ..., F(H,W,C)] \tag{3.4}$$

where the value of the feature map at position (i,j) in channel c is represented by F(i,j,c). The 2D or 3D feature map is flattened into a 1D vector using the equation above, which may then be utilized as input to a fully connected layer or other kinds of layers in the neural network.

Fully Connected Layer

The flattened feature vector serves as the fully connected layer's input, and a group of class scores serves as its output. Neurons in the layer that is entirely interconnected are coupled to each and every neuron in the layer below it. Generally, the output of the fully connected layer is fed into a softmax layer, which converts class scores into a probability distribution across classes.

3.2.3 Activation Function

An activation function is a mathematical function that deep neural networks employ to introduce nonlinearity into a layer's output. The use of activation functions is used to achieve this. It is used to change a layer's output into a more complex representation, allowing the network to learn complex and non-linear relationships between the input and the output. The ReLU activation function is applied following each convolutional layer in any CNN. As a result, the layer's output is effectively made non-linear, allowing the network to learn intricate details from the input images.

The activation functions Sigmoid, Tanh (Hyperbolic Tangent), and Leaky ReLU are frequently employed in deep learning. Based on the features of the data and the nature of the problem being addressed, the activation function is chosen.

ReLU (Rectified Linear Unit):

Deep neural networks frequently use the ReLU activation function since research has shown that it enhances the training of deep models. Any negative input is translated to 0, while positive inputs are left unaffected. The ReLU function is described as:

$$f(x) = max(0, x) \tag{3.5}$$

The activation function's input in this case is 'x'. The ReLU function's output is almost always non-negative. The ReLU function's output is equal to the input when the input is positive. The output of the ReLU function is zero when the input is negative. ReLU is frequently used as the activation function in a neural network's hidden layers, while sigmoid is frequently utilized as the activation function in the output layer for binary classification tasks.

Sigmoid:

For binary classification tasks, the sigmoid activation function is frequently used. All real-valued input is converted to a value between 0 and 1, which can be thought of as a probability. The sigmoid function is described as:

$$f(x) = \frac{1}{(1+e^{-x})} \tag{3.6}$$

In this case, the activation function's input is 'x'. The sigmoid function's output has an inflection point at x = 0, and it always ranges between 0 and 1. The sigmoid function's output gets closer to one when the input is positive. The output of the sigmoid function gets closer to zero when the input is negative.

3.3 Federated Learning

3.3.1 Federated Learning Mechanism

Federated Learning is a collaborative machine learning technique where multiple edge devices(clients) participate remotely to train a common, robust machine learning model (server) without exchanging/sharing their local data samples with the centralized location or server which helps organizations to make better decisions with AI, addressing critical issues of data privacy, security, access rights and access to heterogeneous data.

3.3.2 Traditional ML Vs. Federated Learning

Traditional machine-learning approaches require a centralized location i.e. a single server to aggregate user data and carry out the complete training process, whereas federated learning allows continuous training on edge devices while ensuring no sensitive information exits the device. In the classical machine learning model, it is quite common to assume that the data are independent and identically distributed. On the contrary, federated learning assumes the data to be non-i.i.d because in real-world circumstances, various users have different datatypes and the expected and actual number of actors varies.

3.3.3 Data Partitioning

There are 3 types of data partitioning systems in federated learning: horizontal data partitioning, vertical data partitioning and federated transfer learning system. The description of the these 3 types is given below:

1. Horizontal Data Partitioning : In horizontal partition, each edge device has overlapping features with different observations. For instance, if multiple hospitals in various countries collect data on breast cancer patients but have little to no overlapping of patients, that distribution is horizontal partitioning (Pfitzner, Steckhan, & Arnrich, 2021a).

2. Vertical Data Partitioning : In the vertical partition, each device has different features with overlapping observations. For example, if a hospital suggests patients to a specific surgeon, both the hospital and the surgeon collect different kinds of data but will have many common patients (Pfitzner et al., 2021a).

3. Federated Transfer Learning : If there are a few similar samples with few similar features, but also samples and features do not overlap, then federated transfer learning can be applied in this situation (Pfitzner et al., 2021a).

3.3.4 Foundation ML model

Depending on real-life scenarios, limitations of available datasets and problem statements, different kinds of ML models are decided to be used as the foundation model of federated learning architecture. Neural networks, decision trees, and even linear models are used on basis of use cases. This base model serves as a global model which is initially distributed as an untrained or pre-trained model from a central server to the local clients. The clients then collaboratively train the global model by continuously improving it in every communication round between the central server and the local clients.

3.3.5 Communication Architecture

There are two types of FL communication architecture: centralized and decentralized. Both types of architecture work similarly; the distinction between them is in client-server communication.

1. Centralized Federated Learning : In a centralized federated learning system, a single central server is used, so there is only one possible point of failure (Korkmaz et al., 2020). In a centralized federated learning environment, a central server is utilized to manage all the participating nodes and orchestrate the various steps of the algorithms. The server is in charge of selecting the nodes at the start of the training process and aggregating the received model parameter updates. The server can end up being the system's bottleneck because each of the chosen nodes must communicate updates to the server (Pfitzner, Steckhan, & Arnrich, 2021b).

2. Decentralized Federated Learning : Decentralized federated learning does not rely on a single central server to provide updates, in contrast to centralized federated learning (Korkmaz et al., 2020). In this architecture, the nodes can collaborate among themselves to produce the global model. As the model updates are solely transferred between linked nodes without the coordination of a central server, this configuration avoids single-point failures (*Federated learning*, 2023).

3.3.6 Scale of Federation

Federated learning can be divided into two categories based on the participating clients and the model training scale: cross-device FL and cross-silo FL (Huang, Huang, & Liu, 2022).

1. Cross-silo: Cross-silo FL, where clients are organizations or companies and the client number is typically low (e.g., within a hundred) (Huang et al., 2022).

2. Cross-device: Cross-device, where clients are typically mobile devices and the client number can reach millions (Huang et al., 2022).

3.3.7 One Federated Round

The step where each participant completes training their local model and sends their model weights to the server so the server can combine their global model with the recently updated parameters of local models is referred to as a single communication round between the server and participant clients.

3.3.8 Participants of the network

There are two types of participants in a single federated round :

- Client devices: These are the local devices that participate in the federated learning process by sending the updated weights of their locally trained models to the server.
- Server: The server uses the federated learning process by aggregating the model weights received from the client devices and sending the averaged weights back to the local devices for further training and to improve their accuracy.

3.3.9 Process

In a single federated round, the following steps are completed one after another:

1. The client devices obtain the initialized global model from the cloud server.

2. They train the model using the local datasets and generate the most recent local model update (model parameters).

3. Then they send the updated weights of their locally trained model to a central server.

4. The cloud server collects various local update parameters and aggregates the model weights received from all of the client devices by averaging them together.

5. The averaged weights are then sent back to the client devices, which use them to train their local models further.

6. This process is repeated until the models on the client devices reach satisfactory accuracy.

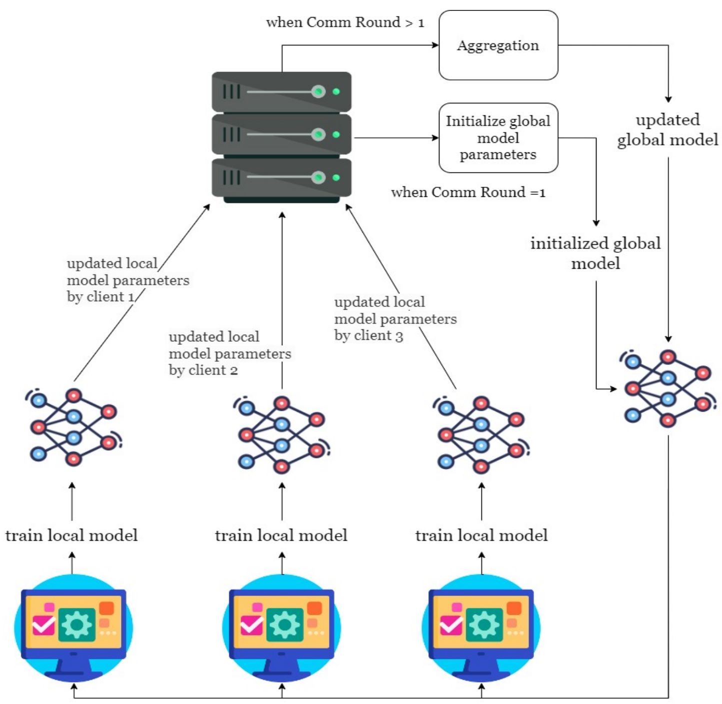


Figure 3.7: Flowchart of federated learning

3.4 Federated Learning training algorithm

3.4.1 FedSGD (Federated Stochastic Gradient Descent)

Stochastic Gradient Descent (SGD) can be applied to as a base to the Federated Learning training algorithm. A single batch gradient calculation is done on each round of communication. Although this approach is computationally efficient, it requires a large number of training rounds in order to produce good models. To apply this approach in the federated optimization problem, C-fraction of clients are selected on each round and the gradient of the loss is computed over all the data of the clients. Thus C is being used to control the global batch size, which when set to C=1, corresponds to full-batch (non-stochastic) gradient descent. This baseline algorithm is termed as FederatedSGD (or FedSGD).
The terms associated with the FedSGD computation are as follows:
w_t = Model weights in communication round #t
w_t^k = Model weights in communication round #t on client k
C = Fraction of clients performing computations in each round
E = Number of training passes each client makes over its local dataset on each round
B = The local minibatch size used for the client updates
η = The learning rate
Φ_k = Set of data points on client k
η_k = Number of data points on client k
f_i = Loss l(x_i, y_i, z_i) i.e., loss on example (x_i, y_i) with model parameters w

3.4.2 FedAvg (Federated Averaging)

One approach of FedSGD is that, the client computes the gradient, updates the model and sends it to the server. Now, if the model is updated multiple times before being sent to the server for aggregation, then the method is called FederatedAVG (or FedAVG). Here, the computation is kept in following parameters:
C = Fraction of clients participating in that round

E = No. of training passes each client makes over its local dataset each round

B = Local minibatch size used for client updates

K = number of clients indexed by k ;

The pseudo code for FedAVG algorithm is given below:

Algorithm 1: Federated Learning Algorithm

Server executes:
initialize w_0
for each round $t = 1, 2, \ldots$ do
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow random\ set\ of$ m $clients$
  for each client $k \in S_t$ in parallel do
    $w_{t+1}^k \leftarrow$ ClientUpdate (k, w_t)
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate (k, w) : // Run on client k
  $\mathscr{B} \leftarrow ($ split $\mathscr{P}_k$ into batches of size $B)$
  for each local epoch i from 1 to E do
    for batch $b \in \mathscr{B}$ do
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

CHAPTER 4

METHODOLOGY

This chapter discusses about the methodology that has been used to carry out the thesis work. The research is carried out using two approaches: Classical Machine Learning approach and Federated Learning approach.

4.1 Identification of Research Gaps

Defining the system's goals and reading certain literature reviews are the first steps of the applied research methodology. The primary goals of the thesis are presented and the dataset of pap smear test cervical cell images is acquired. Then two approaches were followed for classification of those images.

4.2 Classical ML approach

In this first approach, fifteen different Machine Learning models have been applied to train the model to determine which would provide an overall increased accuracy. Subsequently, the models are compared with respect to the accuracy, precision, recall and f1-score of the training set and the accuracy, precision and recall of the test set.

4.3 Federated Learning Approach

In the second approach, various CNN models were developed and integrated with FL architecture. Hyperparameter tuning was done to figure out the best possible architecture. After evaluating the models, the best model was deployed through streamlit and a user interface was designed in order to apply the proposed system in a real-world environment.

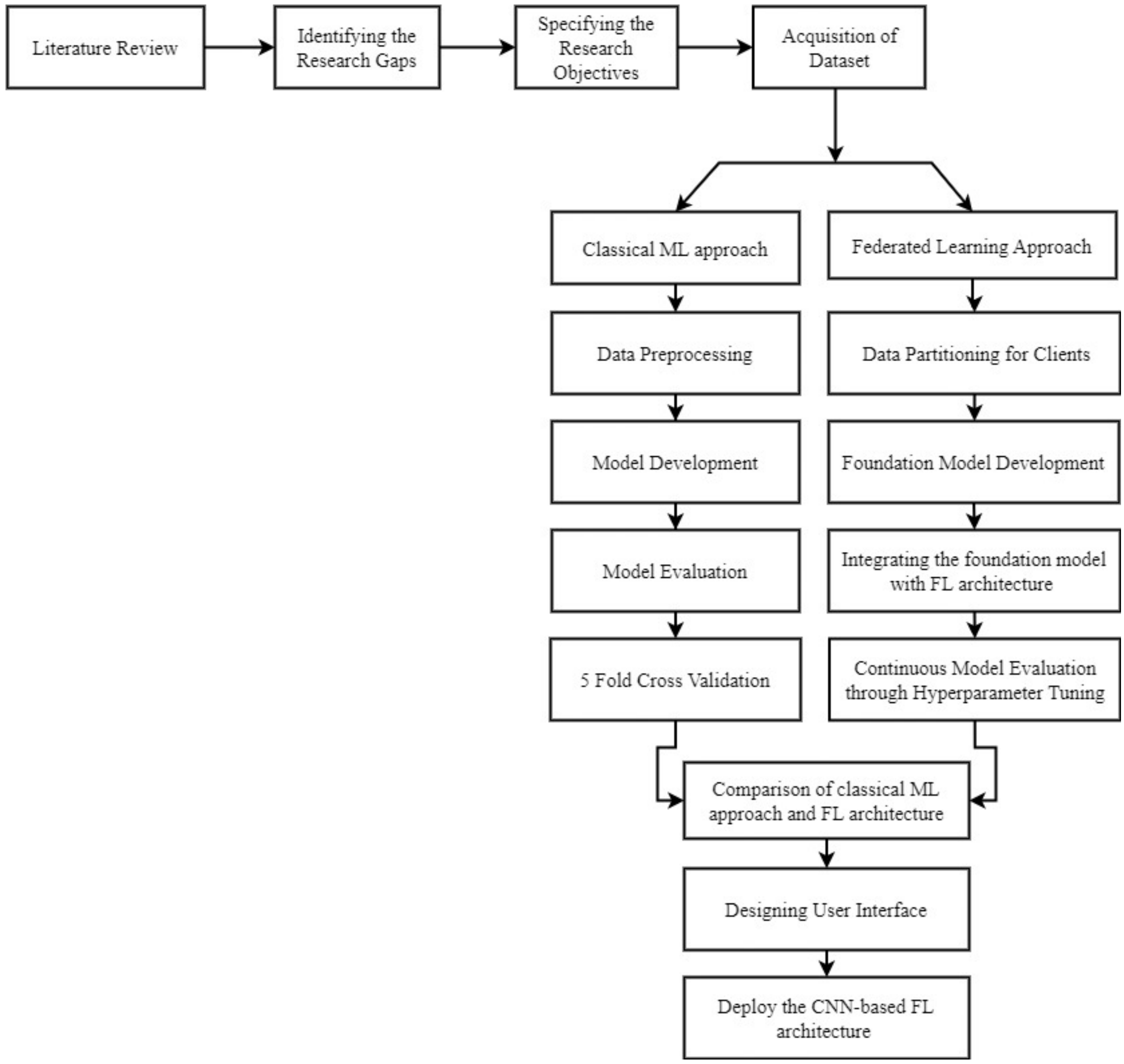


Figure 4.1: Overview of Methodology

An overview of methodology is shown in Fig 4.1. By following this methodology, a conclusion was reached that FL performs significantly better than classical ML models.

CHAPTER 5

SYSTEM DEVELOPMENT

This chapter discusses the process of the development of the whole system. There were two approaches to this research. One approach included classical machine learning algorithms to predict the classes of cervical cells. Another approach was CNN-based federated learning technique in order to achieve the same objectives.

5.1 Data Collection

The dataset used for the development of classic ML and base model of FL architecture has been taken from Kaggle namely SipakMed Database (Plissiti et al., 2018b), which is the largest dataset available. The database contains images of isolated cells, taken manually from the cluster cell images of Pap smear slides. The dataset contains a total of 4049 images, each of size 66X66. The cell images are divided into five categories in which there are 813 images of Dyskeratotic cells, 825 images of Koilocytotic cells, 793 images of Metaplastic cells, 787 images of Parabasal cells, and lastly 831 images of Superficial-Intermediate cells.

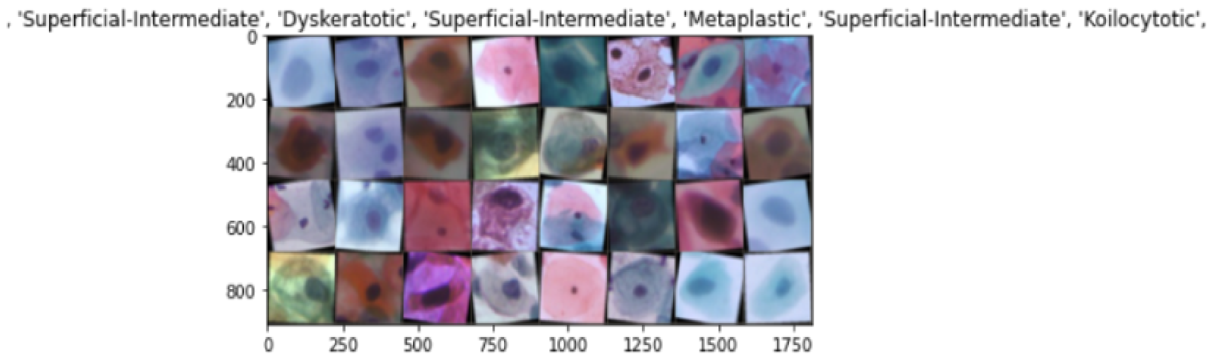


Figure 5.1: Visualization of the SIPAKMED dataset

5.2 Developing the Classical ML models

5.2.1 Data Pre-processing

a. Gray-scale Conversion: Gray-scale conversion of images facilitates the simplification of algorithms and also removes the difficulties associated with computational requirements. Grayscale compression reduces an image to its most basic pixel.

b. Flattening: In order to feed information from a 1-D array into a classification model, a technique known as flattening is utilized to transform multidimensional arrays into 1-D arrays.The multidimensional image array was flattened before processing the data to the ml models because 1-D arrays use less memory and multi-dimensional arrays require more memory. Flattening aids in minimizing memory usage as well as speeding up model training.

c. Resizing the images: We resized the images to the size of 28x28 pixels and created a dataframe of 784 pixels for each images.

d. Rescaling the images: Each image was re-scaled from pixel range to the range of 0 to 255.

e. Train-Test Splitting: The dataset was divided into train and test images with a test ratio of 20%.

f. Handling Imbalanced Data: After train-test splitting, Synthetic Minority Oversampling Technique, or SMOTE was used for handling imbalanced data. Figure 5.2 shows the effect of the process. Before the 5 cells were found to have variations in total numbers of data, but after the application of SMOTE the amount of data in each class gets equally distributed, and thus aiding in removing the imbalance of data.

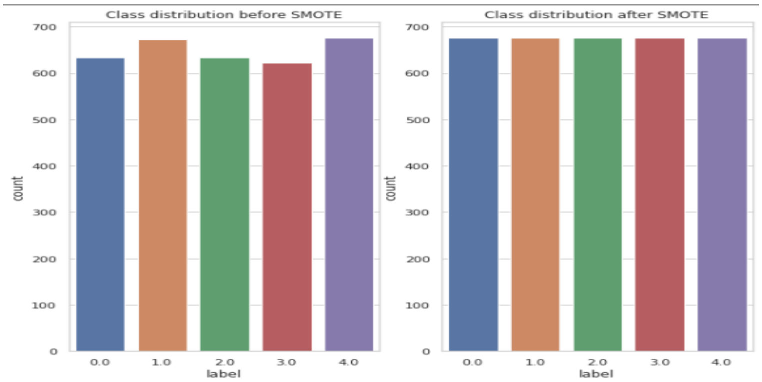
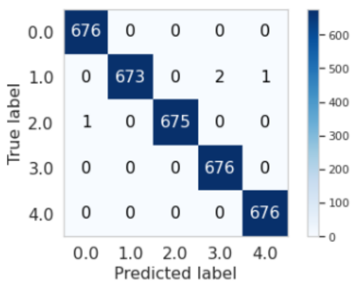


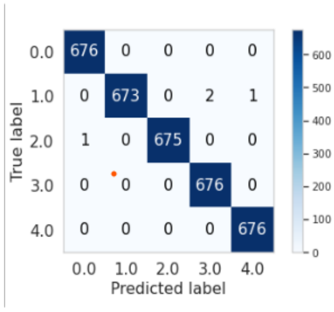
Figure 5.2: Class distribution before and after applying SMOTE

5.2.2 Result Analysis of the ML Models

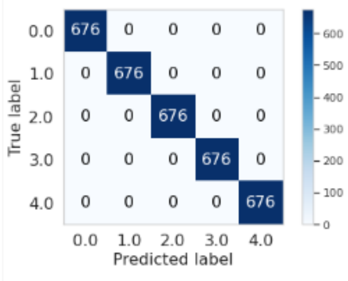
A confusion matrix is the table used to describe the performance of a classification model or classifier on a set of test data for which the true values are known. A confusion matrix is used to measure the performance of a classifier in depth. Figure-5.3 shows the confusion matrices for train data and Figure-5.4 shows the confusion matrices for test data. Here the top 5 models are as follows: LightGBM, HGB+LightGBM, HGB, Extratrees and SVM. The lightgbm model is found to perform better than the other models.



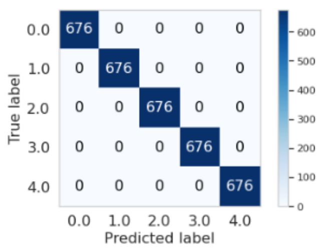
(a)



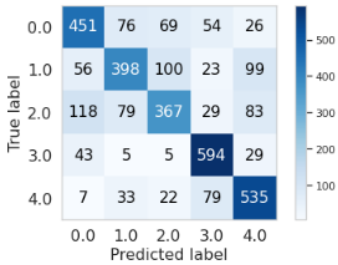
(b)



(c)

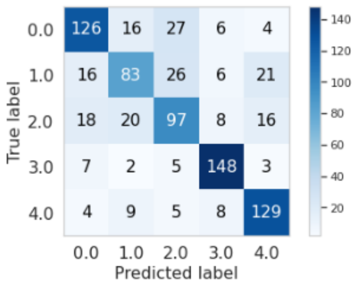


(d)

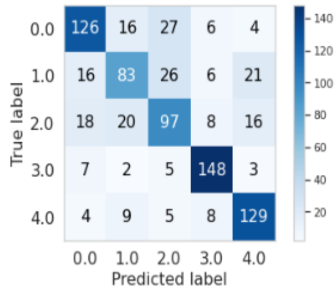


(e)

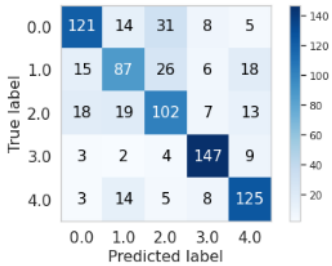
Figure 5.3: Confusion Matrix of true train vs predicted train for (a) LightGBM, (b) HGB + LightGBM, (c) HGB, (d) Extratrees and (e) SVM



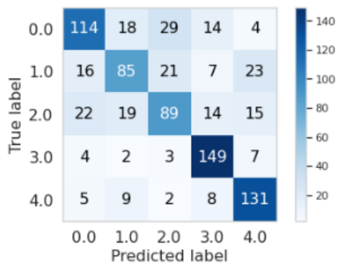
(a)



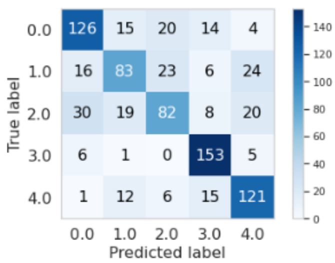
(b)



(c)



(d)



(e)

Figure 5.4: Confusion Matrix of true test vs predicted test for (a) LightGBM, (b) HGB + LightGBM, (c) HGB, (d) Extratrees and (e) SVM

Evaluation Metrics

Table-5.1 shows the evaluation matrics of train data, where the LightGBM and HGB+LightGBM are found to provide same accuracy. Similarly, HGB, Xtra Trees and KNN, Decision trees are also found to perform similar. In the table-5.2 the evaluation metrics of the test data are sorted according to the test accuracy.

Table 5.1: Evaluation Metrics of Train Data

Model	Accuracy	Precision	Recall	F1_score
LightGBM	0.999	0.999	0.999	0.999
HGB+LGBM	0.999	0.999	0.999	0.999
HGB	1	1	1	1
Xtra Trees	1	1	1	1
SVM	0.694	0.69	0.694	0.689
SVM_Grid_Search	0.872	0.875	0.872	0.872
Gradient Boosting	0.846	0.847	0.846	0.845
XGBOOST	0.77	0.775	0.77	0.768
MLP	0.705	0.716	0.705	0.704
HGB + XGBoost	0.768	0.771	0.768	0.766
KNN	1	1	1	1
Decision Tree	1	1	1	1
AdaBoost	0.481	0.496	0.481	0.483
Random Forest	0.477	0.488	0.477	0.449
Gaussian Naive Bayes	0.408	0.449	0.408	0.361

Table 5.2: Evaluation Metrics of Test Data

Model	Accuracy	Precision	Recall	F1_score
LightGBM	0.72	0.714	0.718	0.714
HGB+LGBM	0.72	0.714	0.718	0.714
HGB	0.719	0.715	0.717	0.715
Xtra Trees	0.701	0.694	0.701	0.694
SVM	0.698	0.689	0.695	0.688
SVM_Grid_Search	0.695	0.685	0.692	0.687
Gradient Boosting	0.686	0.678	0.685	0.678
XGBOOST	0.683	0.678	0.681	0.674
MLP	0.673	0.681	0.67	0.668
HGB + XGBoost	0.669	0.662	0.667	0.66
KNN	0.562	0.567	0.555	0.539
Decision Tree	0.535	0.527	0.531	0.529
AdaBoost	0.485	0.503	0.482	0.486
Random Forest	0.456	0.46	0.448	0.424
Gaussian Naive Bayes	0.435	0.456	0.423	0.369

Cross Validation Results

Cross Validation evaluates the model using different chunks of the data set as the validation set. In K-fold cross validation, 'K' represents the number of groups that a given dataset is to be split into. The dataset is divided into k parts, and in different iterations, one part is used for testing the model, while the other remaining parts are used for training purposes. Then after k-iterations, an average of all the individual scores is done to obtain the final score. In this study, 5-fold cross validation is applied on the entire dataset.

In table 5.3, 5-fold cross validation results are shown for training set and in table 5.4, 5-fold cross validation results are shown for validation set. Here the top 6 models which performs best on training data are: Histogram Gradient Boosting (HGB) + LightGBM, LightGBM, Histogram Gradient Boosting, Extreme Gradient Boosting (XGBoost), Histogram Gradient Boosting + Extreme Gradient Boosting and Extra trees.

Table 5.3: Cross Validation Results for Training Set

Model_name	Mean Training Accuracy (in percentage)	Mean Training Precision	Mean Training Recall	Mean Training F1 Score
HGB + LightGBM	100	1	1	1
LightGBM	100	1	1	1
HGB	100	1	1	1
XGBoost	100	1	1	1
HGB + XGBoost	100	1	1	1
Extra Trees	100	1	1	1
Gradient Boosting	84.90985253	0.849366961	0.849433242	0.848145364
SVM_Grid_Search	86.81156926	0.870367728	0.868133276	0.86773494
SVM	69.93081236	0.696427677	0.699306189	0.693327781
MLP	68.77627221	0.696460777	0.688059482	0.683649547
KNN	100	1	1	1
Decision Tree	100	1	1	1
Random Forest	47.24004132	0.478736029	0.47345261	0.447405383
Adaboost	47.97471785	0.496924946	0.480725118	0.483934274
Naive Bayes	41.37440111	0.448321855	0.412054824	0.363859397

Table 5.4: Cross Validation Results for Validation Set

Model_name	Mean Validation Accuracy (in percentage)	Mean Validation Precision	Mean Validation Recall	Mean Validation F1 Score
HGB + LightGBM	70.56133925	0.701983809	0.706121962	0.702796063
LightGBM	70.56133925	0.701983809	0.706121962	0.702796063
HGB	69.72107006	0.693044681	0.697875695	0.694534567
XGBoost	69.3016527	0.688479078	0.693655077	0.689358876
HGB + XGBoost	69.05428131	0.685542518	0.691234928	0.686148781
Extra Trees	68.9553938	0.684843567	0.690223384	0.684319871
Gradient Boosting	66.56002686	0.661458162	0.6663828	0.661037635
SVM_Grid_Search	66.21407316	0.657991898	0.66290078	0.65762497
SVM	63.79355705	0.631791996	0.638140273	0.629743308
MLP	60.87915274	0.610446237	0.609444673	0.600964649
KNN	53.02635474	0.553604967	0.533233073	0.515285127
Decision Tree	51.0741809	0.508561817	0.511054756	0.508912903
Random Forest	46.40647652	0.46922437	0.465063736	0.439952827
Adaboost	44.06079751	0.457688023	0.441604779	0.444543543
Gaussian Naive Bayes	41.17035206	0.444675024	0.409996198	0.360853211

5.3 Development of Federated Learning Architecture

5.3.1 Data Partitioning

For developing a system similar to horizontal data-partitioning system, the SIPAKMED dataset was divided into three training sets for three clients, each having 28% images of the whole dataset and a single test set which is 15% of the whole dataset. Table 5.5 shows how many training images of each classes a single client got after dividing the whole dataset.

Table 5.5: Division of images on 3 clients

Class No.	Client_1(28%)	Client_2(28%)	Client_3(28%)	Test (15%)
Class1	230	230	230	123
Class2	230	230	228	137
Class3	215	215	215	148
Class4	215	215	215	142
Class5	230	235	231	135

5.3.2 Data Augmentation

The Keras ImageDataGenerator class was used for numerous augmentation methods, including zooming, shearing, rescaling and horizontal flips.

5.3.3 Development of the Foundation Model

Keras model for cervical cell classification was developed using a Convolutional Neural Networks.

Input Shape

The input_shape argument for the 2 CNN Model was set to (66, 66, 3). This specifies the shape of the input images that the model had been trained on. The input images were expected to have a width of 66 pixels, a height of 66 pixels, and 3 color channels (corresponding to the Red, Green, and Blue color channels).

Output Shape

The final output shape of the Keras model was 5 as in the number of cell classes. These five values were output by the last layer of the model, which has a sigmoid activation function.

Development of the CNN architecture

Description for the two CNN architectures developed as a foundation model for the FL architecture are given below:

Layers utilized in the CNN architecture 1

- *Conv2D(filters=16, kernel˙size= (3,3), activation= 'relu', input˙shape=(66,66,3)):* This is a convolutional layer with 16 filters of size (3, 3) and a ReLU activation function.
- *Conv2D(filters=32, kernel˙size=(3,3), activation='relu'):* This is a convolutional layer with 32 filters of size (3, 3) and a ReLU activation function.

- *MaxPooling2D((2, 2)):*This is a max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2. The layer takes the maximum value in each 2x2 window of the input.
- *Conv2D(filters=64, kernel˙size=(3,3), activation='relu'):* This is another convolutional layer with 64 filters of size (3, 3) and a ReLU activation function.
- *MaxPooling2D((2, 2)):*This is a max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2. The layer takes the maximum value in each 2x2 window of the input.
- *Conv2D(filters=128, kernel˙size=(3,3), activation='relu'):* This is another convolutional layer with 128 filters of size (3, 3) and a ReLU activation function.
- *MaxPooling2D((2, 2)):* This is another max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2.
- *Dropout(rate=0.25)* This is a dropout layer as a regularization technique for overfitting problem.
- *Flatten():* This layer flattens the output of the previous layer into a 1D tensor, which can then be passed to a fully connected layer.
- *Dense(64, activation='relu'):* This is a fully connected layer with 64 units and a ReLU activation function.
- *Dropout(rate=0.25)* This is a dropout layer as a regularization technique for overfitting problem.
- *Dense(5, activation='sigmoid'):* This is the final output layer with 5 units (one for each class) and a sigmoid activation function. The linear activation function means that the output of the layer is not constrained to a particular range, so it can take any real value.

Table 5.6: CNN architecture 1

Name	Type	Shape	Parameters
conv2d_44	Conv2D	(None, 64, 64, 16)	448
conv2d_45	Conv2D	(None, 62, 62, 32)	4640
max_pooling2d_33	MaxPooling2D	(None, 31, 31, 32)	0
conv2d_46	Conv2D	(None, 29, 29, 64)	18496
max_pooling2d_34	MaxPooling2D	(None, 14, 14, 64)	0
conv2d_47	Conv2D	(None, 12, 12, 128)	73856
max_pooling2d_35	MaxPooling2D	(None, 6, 6, 128)	0
dropout_22	Dropout	(None, 6, 6, 128)	0
flatten_11	Flatten	(None, 4608)	0
dense_33	Dense	(None, 64)	294976
dropout_23	Dropout	(None, 64)	0
dense_34	Dense	(None, 5)	325
Total params: 392,741			
Trainable params: 392,741			
Non-trainable params: 0			

Table 5.7: Hyperparameter set 1 used for CNN Model 1

Parameters Used	**Values**
Number of Conv2D layers	4
Number of MaxPooling2D layers	3
Number of Dense layers	2
Number of Dropouts	2
Number of units in the Dense layers	64, 5
Activation functions used	ReLU for Conv2D layers and Dense layers, and sigmoid for the output layer
Loss function	categorical˙crossentropy
Optimizer	Adam
Input image size	66 x 66 x 3 (RGB image with 66 height, 66 width, and 3 channels)
Batch Size	100
Total Communication Round	10
Steps Per Epoch for Local Train	3
Number of Epochs for Local Train	150
Learning Rate	0.013

Table 5.8: Evaluation Metrics for CNN Architecture 1 and Hyper-parameter set 1

Model	Hyper-parameter Set	Test Accuracy	Precision	Recall	f1-score	roc_auc_score
CNN 01	1	82.18%	82.59%	83.47%	82.62%	89%

To increase the performance of CNN model displayed in Table 5.8, there were changes made in the architecture as well as two different hyper-parameter sets were used.

Layers utilized in the CNN architecture 2

- *Conv2D(filters=16, kernel˙size= (3,3), activation= 'relu', input˙shape=(66,66,3)):* This is a convolutional layer with 16 filters of size (3, 3) and a ReLU activation function.
- *Conv2D(filters=32, kernel˙size=(3,3), activation='relu'):* This is a convolutional layer with 32 filters of size (3, 3) and a ReLU activation function.
- *MaxPooling2D((2, 2)):*This is a max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2. The layer takes the maximum value in each 2x2 window of the input.
- *BatchNormalization():* This layer is used to prevent model overfitting.
- *Conv2D(filters=64, kernel˙size=(3,3), activation='relu'):* This is another convolutional layer with 64 filters of size (3, 3) and a ReLU activation function.
- *MaxPooling2D((2, 2)):* This is another max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2.
- *Conv2D(filters=128, kernel˙size=(3,3), activation='relu'):* This is another convolutional layer with 128 filters of size (3, 3) and a ReLU activation function.
- *MaxPooling2D((2, 2)):* This is another max pooling layer that reduces the spatial dimensions of the output of the previous layer by a factor of 2.
- *Dropout(rate=0.25)* This is a dropout layer as a regularization technique for overfitting problem.
- *Flatten():* This layer flattens the output of the previous layer into a 1D tensor, which can then be passed to a fully connected layer.

- *Dense(256, activation='relu'):* This is a fully connected layer with 256 units and a ReLU activation function.
- *Dense(128, activation='relu'):* This is a fully connected layer with 128 units and a ReLU activation function.
- *Dropout(rate=0.25)* This is a dropout layer as a regularization technique for overfitting problem.
- *Dense(5, activation='sigmoid'):* This is the final output layer with 5 units (one for each class) and a sigmoid activation function. The linear activation function means that the output of the layer is not constrained to a particular range, so it can take any real value.

Table 5.9: CNN architecture 2

Name	Type	Shape	Parameters
conv2d_40	Conv2D	(None, 64, 64, 16)	448
conv2d_41	Conv2D	(None, 62, 62, 32)	4640
max_pooling2d_30	MaxPooling2D	(None, 31, 31, 32)	0
batch_normalization_10	BatchNormalization	(None, 31, 31, 32)	128
conv2d_42	Conv2D	(None, 29, 29, 64)	18496
max_pooling2d_31	MaxPooling2D	(None, 14, 14, 64)	0
conv2d_43	Conv2D	(None, 12, 12, 128)	73856
max_pooling2d_32	MaxPooling2D	(None, 6, 6, 128)	0
dropout_20	Dropout	(None, 6, 6, 128)	0
flatten_10	Flatten	(None, 4608)	0
dense_30	Dense	(None, 256)	1179904
dense_31	Dense	(None, 128)	32896
dropout_21	Dropout	(None, 128)	0
dense_32	Dense	(None, 5)	645
Total params: 1,311,013			
Trainable params: 1,310,949			
Non-trainable params: 64			

Table 5.10: Hyperparameter set 2 used for CNN Model 2

Parameters Used	Values
Number of Conv2D layers	4
Number of MaxPooling2D layers	3
Number of Dense layers	3
Number of Dropouts	2
No of BatchNormalization Layer	1
Number of units in the Dense layers	256, 128, 5
Activation functions used	ReLU for Conv2D layers and Dense layers, and sigmoid for the output layer
Loss function	categorical˙crossentropy
Optimizer	Adam
Input image size	66 x 66 x 3 (RGB image with 66 height, 66 width, and 3 channels)
Batch Size	10
Total Communication Round	20
Steps Per Epochs for Local Train	60
Number of Epochs for Local Train	30
Learning Rate	0.001

Table 5.11: Hyperparameter set 3 used for CNN Model 2

Parameters Used	Values
Number of Conv2D layers	4
Number of MaxPooling2D layers	3
Number of Dense layers	3
Number of units in the Dense layers	256, 128, 5
Number of Dropouts	2
No of BatchNormalization Layer	1
Activation functions used	ReLU for Conv2D layers and Dense layers, and sigmoid for the output layer
Loss function	categorical˙crossentropy
Optimizer	Adam
Input image size	66 x 66 x 3 (RGB image with 66 height, 66 width, and 3 channels)
Batch Size	10
Total Communication Round	30
Steps Per Epoch for Local Train	100
Number of Epochs for Local Train	40
Learning Rate	0.001

Table 5.12: Evaluation Metrics for CNN Architecture 2 and Two Hyper-parameter sets

Model	Hyper-parameter Set	Test Accuracy	Precision	Recall	f1-score	roc_auc_score
CNN 02	3	88.46%	89.58%	88.47%	87.83%	93%
CNN 02	2	87.00%	86.38%	84.47%	86.83%	92%

The test results obtained from CNN architecture 02 and hyper-parameter 3 is the best amongst all the classical ML Models and other CNN architectures. Hence proposed CNN architecture 02 along with hyper-parameter set 03 were used in the development of the prototype for testing in the real-world environment.

5.3.4 Result Analysis of CNN-based Federated Learning

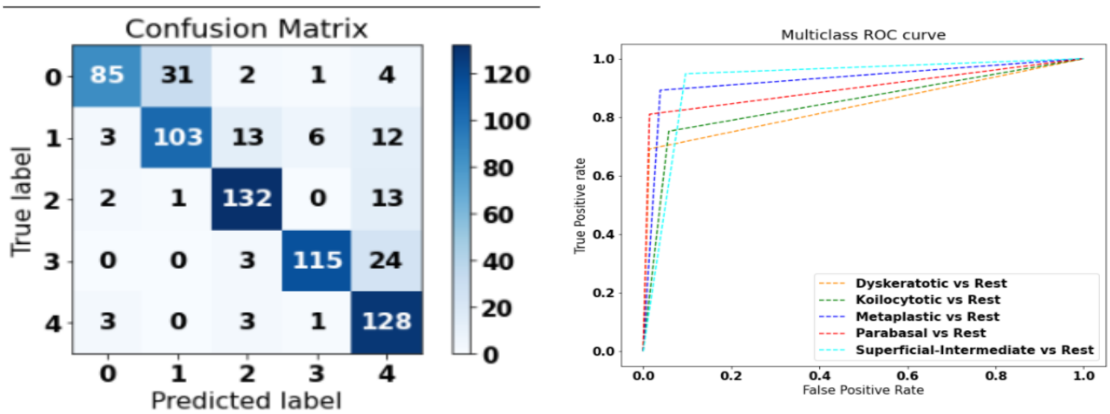


Figure 5.5: Confusion Matrix and ROC_AUC_Curve with CNN 1 + Hyper-parameter Set 1

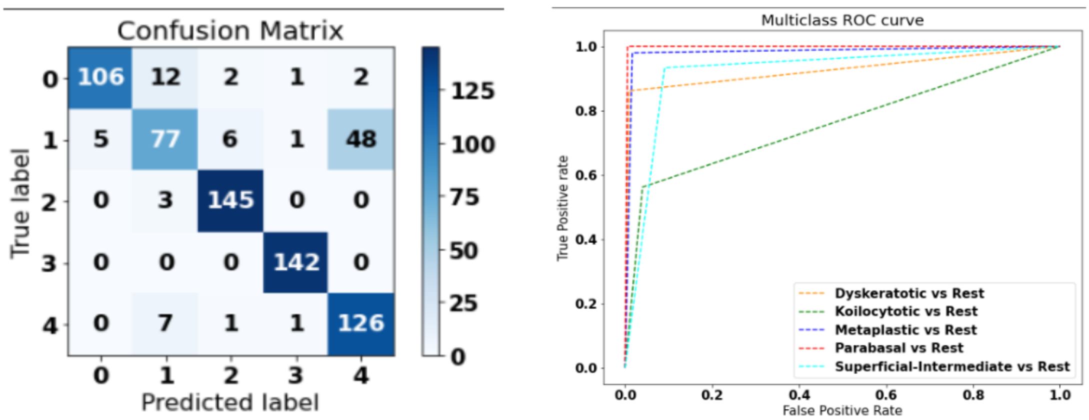


Figure 5.6: Confusion Matrix and ROC_AUC_Curve with CNN 2 + Hyper-parameter Set 2

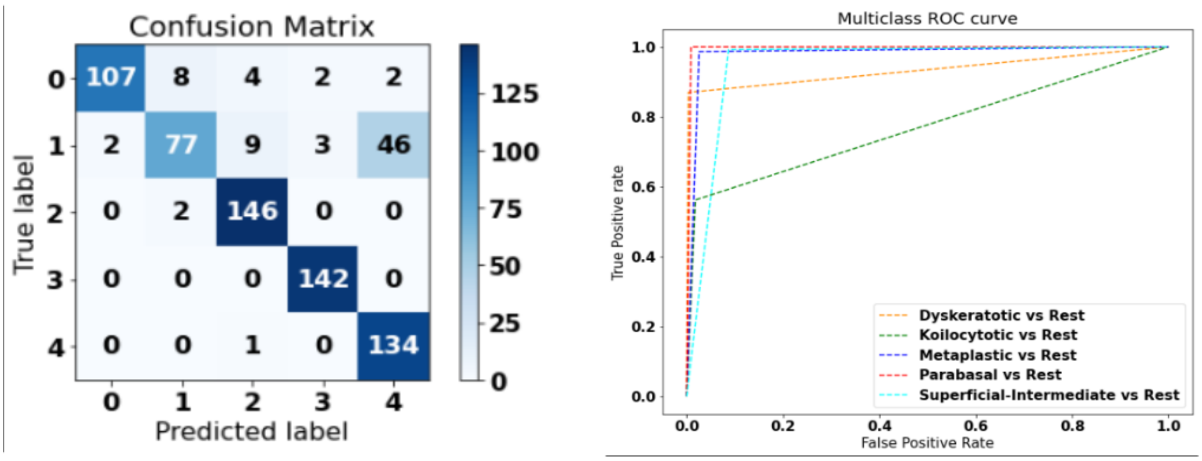


Figure 5.7: Confusion Matrix and ROC_AUC_Curve with CNN 2 + Hyper-parameter Set 3

Figure-5.5, Figure-5.6, and Figure-5.7 show the confusion matrix on test data for the 3 configurations: CNN architecture 01+ Hyper-parameter Set 01, CNN architecture 02+ Hyper-parameter Set 02, and CNN architecture 02+ Hyper-parameter Set 03 respectively. The last configuration performs better as displayed by the One-Vs-Rest Multiclass ROC curve. Computing a ROC curve for each of the n classes constitutes the One-vs-the-Rest (OvR) multiclass technique, sometimes referred to as one-vs-all. A certain class is considered the positive class in each phase, and the remaining classes are collectively considered the negative class. It is evident from the plots that the AUC for each classes in the Configuration 3 ROC curve is higher than Configuration 1 and 2 ROC curves. Therefore, it can be said that CNN architecture 02 + Hyper-parameter set 03 did a better job of classifying amongst all configurations and also classical ML models.

5.3.5 Prototype Development

For deploying the proposed system in real-life environment a prototype was developed using streamlit and ngrok. There are two sides of the UI development:

- **Client Side** : In the client-side UI, a client can follow these necessary steps for training local model: enter the hospital/institution id, request for the global model parameters from the server, upload a zip file of training images, training the local model, send locally trained model parameters to the server and lasty test the global and local model with test images.

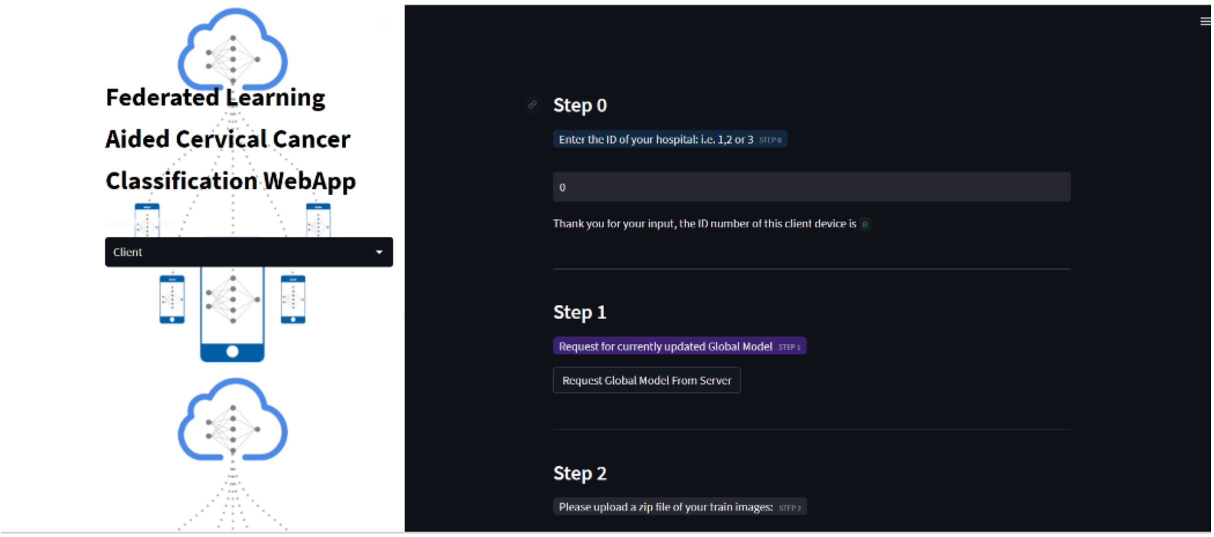


Figure 5.8: Development of Client-Side Frontend

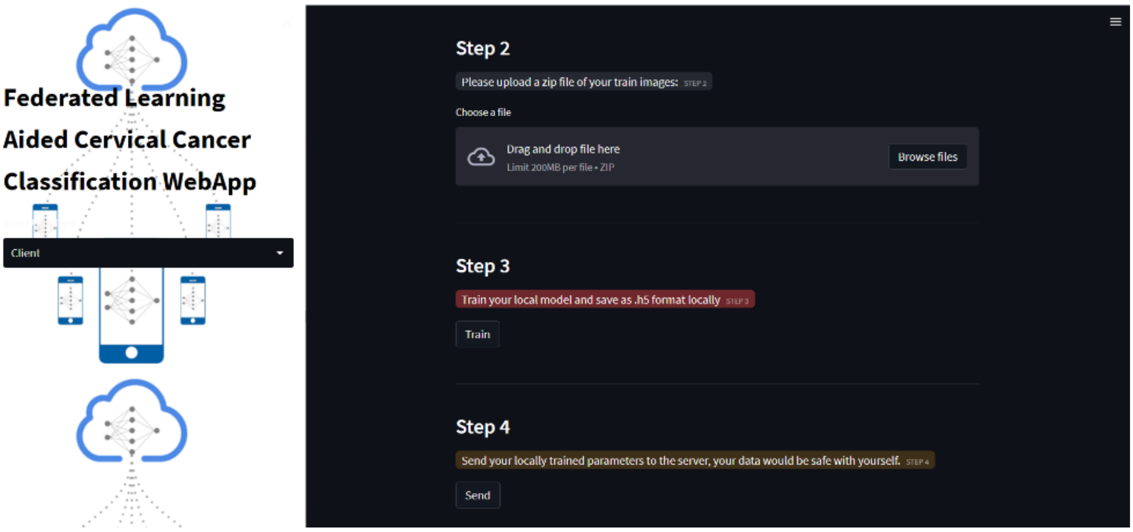


Figure 5.9: Development of Client-Side Frontend

- **Server Side** : In the server-side UI, an admin would first be authorized by entering a credential ID and then he can aggregate the local model updates with the global model.

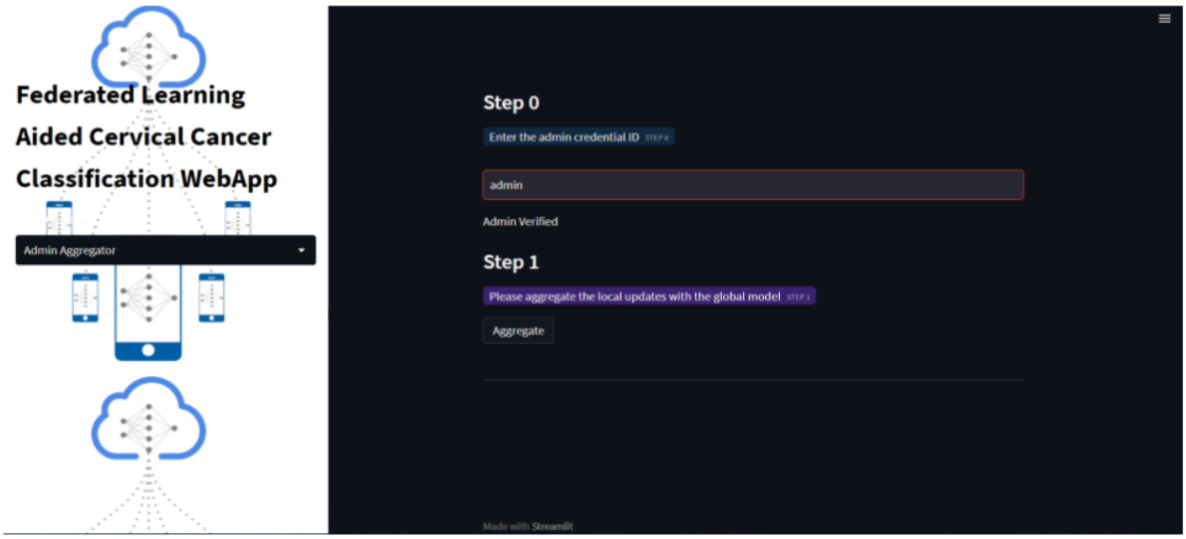


Figure 5.10: Development of Server-Side Frontend

CHAPTER 6

CONCLUSION

This chapter puts forth the summarization of the overall thesis by using four subsections to demonstrate the thesis's overview, contribution, limitations, and scope for future work

6.1 Thesis Overview

The main objective of this thesis is to introduce a system which will be able to classify cervical cancer cells by ensuring privacy. A detailed literature study was conducted to get better knowledge for the development of the system. The exploration aided in knowing about the application of federated learning in classification through data collection. Since very less work has been done, this study tries to incorporate Federated learning in cervical cancer classification. This work helps to avoid data sharing which was of a signifcant challenge in the related works of the past. The work introduces a personalized convolutional neural network for the purpose of classification, which was not introduced in the works before.

The outcome of the thesis is the development of a system that makes use of a novel deep learning architecture, which makes use of the privacy ensured by Federated Learning, for the classification of any input cervical cancer cell.

6.2 Thesis Contribution

The main contribution of this thesis is the development of an interactive web application for classification of cervical cancer cells. The Deep Learning model, incorporated with the FL architecture that can operate well even with small-sized datasets. Thus the system does not require a heavy setup of devices. The system has a centralized global model and initially untrained local models, allocated for each client. The clients train the local models with their dataset, and only the updated local models get aggregated with the global model, thus ensuring privacy as their data is never collected centrally. Thus, through this system, clients from any part of the world can help to improve the accuracy of the model, without having to share their data. This helps to gain an overall enriched model as various types of data get combined for the development of the global model. In some recent works classification of cancer cells incorporating federated can be found, but the approaches have not introduced the development of a customized Convolutional Neural Network. A novel deep learning model, integrated with Federated Learning can result in an improved classification system even with the dearth of a large amount of data.

6.3 Thesis Limitations and Future Work

A few limitations of the thesis are:

1. The method considers the development of a Deep Learning model, which is very time consuming owing to its high computational capacity.
2. The proposed system incorporated data from only three hospitals. The effectiveness of the system may be improvised by the introduction of more hospitals in the future.
3. Although crucial for accurate results, acquiring sufficient quantity of data for training purposes poses a significant limitation.

The future work will focus on introducing differential privacy which would help to give as accurate results as possible while maintaining privacy by enabling the quantification of the extent of privacy of a database. By deploying in the real world, the system can be upgraded through continuous user feedback. And finally, different algorithms can be tested in the future to find out which may result in better accuracy.

References

Adnan, M., Kalra, S., Cresswell, J. C., Taylor, G. W., & Tizhoosh, H. R. (2022). Federated learning and differential privacy for medical image analysis. *Scientific reports*, *12*(1), 1953.

Bhatt, A. R., Ganatra, A., & Kotecha, K. (2021). Cervical cancer detection in pap smear whole slide images using convnet with transfer learning and progressive resizing. *PeerJ Computer Science*, *7*, e348.

Bruni L, S. B. M. M. C. J. G. D. M. J. B. F. d. S. S., Albero G. (2021). *Human papillomavirus and related diseases in bangladesh. summary report 22 october 2021.* ICO/IARC Information Centre on HPV and Cancer (HPV Information Centre).

Chakrabarty, N., Kundu, T., Dandapat, S., Sarkar, A., & Kole, D. K. (2019). Flight arrival delay prediction using gradient boosting classifier. In *Emerging technologies in data mining and information security: Proceedings of iemis 2018, volume 2* (pp. 651–659).

Federated learning. (2023, Feb). Wikimedia Foundation. Retrieved from

Gardner, M. W., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), 2627–2636.

Ghoneim, A., Muhammad, G., & Hossain, M. S. (2020). Cervical cancer classification using convolutional neural networks and extreme learning machines. *Future Generation Computer Systems*, *102*, 643–649.

Huang, C., Huang, J., & Liu, X. (2022). Cross-silo federated learning: Challenges and opportunities. *arXiv preprint arXiv:2206.12949*.

Kaissis, G. A., Makowski, M. R., Rückert, D., & Braren, R. F. (2020). Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, *2*(6), 305–311.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Korkmaz, C., Kocas, H. E., Uysal, A., Masry, A., Ozkasap, O., & Akgun, B. (2020). Chain fl: Decentralized federated machine learning via blockchain. In *2020 second international conference on blockchain computing and applications (bcca)* (pp. 140–146).

Li, H., Chung, F.-l., & Wang, S. (2015). A svm based classification method for homogeneous data. *Applied Soft Computing*, *36*, 228–235.

Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., ... others (2019). Privacy-preserving federated brain tumour segmentation. In *Machine learning in medical imaging: 10th international workshop, mlmi 2019, held in conjunction with miccai 2019, shenzhen, china, october 13, 2019, proceedings 10* (pp. 133–141).

Linardos, A., Kushibar, K., Walsh, S., Gkontra, P., & Lekadir, K. (2022). Federated learning for multi-center imaging diagnostics: a simulation study in cardiovascular disease. *Scientific Reports*, *12*(1), 3551.

Ma, Z., Zhang, M., Liu, J., Yang, A., Li, H., Wang, J., ... Li, M. (2022). An assisted diagnosis model for cancer patients based on federated learning. *Frontiers in Oncology*, 713.

Ong, Y. J., Zhou, Y., Baracaldo, N., & Ludwig, H. (2020). Adaptive histogram-based gradient boosted trees for federated learning. *arXiv preprint arXiv:2012.06670*.

Pandey, P., & Prabhakar, R. (2016). An analysis of machine learning techniques (j48 & adaboost)-for classification. In *2016 1st india international conference on information processing (iicip)* (pp. 1–6).

Pati, S., Baid, U., Edwards, B., Sheller, M., Wang, S.-H., Reina, G. A., ... others (2022). Federated learning enables big data for rare cancer boundary detection. *Nature communications*, *13*(1), 7346.

Pfitzner, B., Steckhan, N., & Arnrich, B. (2021a). Federated learning in a medical context: A systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, *21*(2), 1–31.

Pfitzner, B., Steckhan, N., & Arnrich, B. (2021b). Federated learning in a medical context: A systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, *21*(2), 1–31.

Pimple, S., & Mishra, G. (2022). Cancer cervix: Epidemiology and disease burden. *Cytojournal*, *19*.

Plissiti, M. E., Dimitrakopoulos, P., Sfikas, G., Nikou, C., Krikoni, O., & Charchanti, A. (2018a). Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images. In *2018 25th ieee international conference on image processing (icip)* (pp. 3144–3148).

Plissiti, M. E., Dimitrakopoulos, P., Sfikas, G., Nikou, C., Krikoni, O., & Charchanti, A. (2018b). Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images. In *2018 25th ieee international conference on image processing (icip)* (pp. 3144–3148).

Rehman, A.-u., Ali, N., Taj, I. A., Sajid, M., & Karimov, K. S. (2020). An automatic mass screening system for cervical cancer detection based on convolutional neural network. *Mathematical Problems in Engineering*, *2020*, 1–14.

Shanthi, P., Hareesha, K., & Kudva, R. (2022). Automated detection and classification of cervical cancer using pap smear microscopic images: a comprehensive review and future perspectives. *Engineered Science*, *19*, 20–41.

Sheller, M. J., Reina, G. A., Edwards, B., Martin, J., & Bakas, S. (2019). Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries: 4th international workshop, brainles 2018, held in conjunction with miccai 2018, granada, spain, september 16, 2018, revised selected papers, part i 4* (pp. 92–104).

Silva, E. L., Sampaio, A. F., Teixeira, L. F., & Vasconcelos, M. J. M. (2021). Cervical cancer detection and classification in cytology images using a hybrid approach. In *Advances in visual computing: 16th international symposium, isvc 2021, virtual event, october 4-6, 2021, proceedings, part ii* (pp. 299–312).

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. doi: 10.3322/caac.21660

Taha, B., Dias, J., & Werghi, N. (2017). Classification of cervical-cancer using pap-smear images: a convolutional neural network approach. In *Medical image understanding and analysis: 21st annual conference, miua 2017, edinburgh, uk, july 11–13, 2017, proceedings 21* (pp. 261–272).

Tyree, S., Weinberger, K. Q., Agrawal, K., & Paykin, J. (2011). Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th international conference on world wide web* (pp. 387–396).