Privacy-Preserving Federated Learning for Clinical Text Classification

by

Muktadiru Mannan Shithy
ID: CSE2203027156

Supervised by
Nayeem Ullah

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2026

APPROVAL

The thesis titled "**Privacy-Preserving Federated Learning for Clinical Text Classification**" submitted by **Muktadiru Mannan Shithy (CSE2203027156)** to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Supervisor

Nayeem Ullah
Lecturer
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature)
Examiner 1
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature)
Examiner 2
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature)
Examiner 3
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

I, hereby, declare that the work presented in this report is the outcome of the investigation performed by me under the supervision of **Nayeem Ullah, Lecturer,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. I reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned	Signature
----------	----------
(Nayeem Ullah)	Muktadiru Mannan Shithy
Supervisor	ID: CSE2203027156

ABSTRACT

Automatic classification of symptom descriptions into disease categories can support clinical decision-making. However, centralized machine learning requires training data to be collected in one location, which can create privacy concerns when dealing with sensitive clinical text. This thesis investigates Federated Learning (FL) as a privacy-preserving approach for symptom-to-disease text classification. A client-server FL pipeline is implemented using the Flower framework with a regularized Text Convolutional Neural Network (TextCNN). The Symptom2Disease dataset contains 1,200 samples from 24 disease classes and is divided among three simulated clients using a non-IID Dirichlet distribution ($\alpha = 0.5$). The federated model is trained using the FedAvg aggregation strategy for 30 communication rounds and is compared with centralized machine learning models, including Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest. The proposed federated TextCNN achieves 93.75% test accuracy, 93.50% weighted F1-score, and a ROC-AUC of 0.9976. The results show that Federated Learning can maintain strong classification performance while keeping the raw training data distributed across clients. This study demonstrates the potential of a lightweight federated TextCNN approach for privacy-sensitive clinical text classification and provides a basis for future improvements using stronger privacy mechanisms and advanced language models.

Keywords: Federated Learning, Clinical Text Classification, Privacy-Preserving Machine Learning, TextCNN, Flower Framework, Symptom2Disease.

ACKNOWLEDGMENT

At the very beginning, I would like to express my deepest gratitude to Almighty Allah for giving me the strength and ability to complete this thesis successfully.

I would like to sincerely thank my supervisor, **Nayeem Ullah, Lecturer**, Department of Computer Science and Engineering, Sonargaon University, for his valuable guidance, support, and encouragement throughout this work. His advice and direction were invaluable in completing this thesis.

I would like to convey my special gratitude to **Prof. Bulbul Ahamed**, Head, Department of Computer Science and Engineering, Sonargaon University, for his kind concern and precious suggestions.

I am also grateful to all the teachers of the Department of Computer Science and Engineering for their guidance and support throughout my academic journey and for inspiring me to learn and grow.

Finally, I would like to express my deepest gratitude and love to my parents for their continuous support, encouragement, and endless love.

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
EHR	Electronic Health Record
F1	F1-Score (harmonic mean of precision and recall)
FedAvg	Federated Averaging
FL	Federated Learning
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IID	Independent and Identically Distributed
LR	Logistic Regression
ML	Machine Learning
NB	Naive Bayes
Non-IID	Non-Independent and Identically Distributed
NLP	Natural Language Processing
RF	Random Forest
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency

TABLE OF CONTENTS

Title	Page No
APPROVAL	**ii**
DECLARATION	**iii**
ABSTRACT	**iv**
ACKNOWLEDGMENT	**v**
LIST OF ABBREVIATIONS	**vi**
LIST OF TABLES	**ix**
LIST OF FIGURES	**x**
CHAPTER 1	**1-5**
INTRODUCTION	1
1.1 Introduction	1
1.2 Background and Research Context	1-2
1.3 Challenges of Centralized Learning for Clinical Text Data	2
1.4 Federated Learning for Privacy Preservation	3
1.5 Motivation and Problem Statement	4
1.6 Objectives of the Study	4-5
1.7 Organization of the Thesis	5
CHAPTER 2	**6-8**
LITERATURE REVIEW	6
2.1 Machine Learning Approaches for Symptom-to-Disease Classification	6
2.2 Deep Learning and Transformer-Based Approaches for Clinical Text	6
2.3 Federated Learning: Foundations and Frameworks	7
2.4 Federated Learning in Healthcare and Clinical NLP	7-8
2.5 Research Gap and Motivation	8
CHAPTER 3	**9-15**
METHODOLOGY	9
3.1 Dataset Overview	9
3.2 Dataset Size and Structure	9-10
3.3 Data Preprocessing Pipeline	10
3.3.1 Preprocessing for Baseline Machine Learning Models	10
3.3.2 Preprocessing for the Federated TextCNN Model	10
3.4 Non-IID Federated Data Partitioning	10-11
3.5 Proposed TextCNN Architecture	12-13

3.6 Federated Learning Setup with the Flower Framework 13-14
3.6.1 Client-Side Implementation 13
3.6.2 Server-Side Aggregation Strategy 14
3.7 Baseline Machine Learning Models 14
3.8 Evaluation Metrics 14
3.9 Tools and Environment 15
CHAPTER 4 **16-22**
PROPOSED SYSTEM 16
4.1 Proposed Methodology 16
4.2 System Architecture 16-17
4.3 Model Selection 17
4.3.1 Federated TextCNN (Proposed) 17-18
4.3.2 Logistic Regression 19
4.3.3 Multinomial Naive Bayes 19-20
4.3.4 Support Vector Machine (SVM) 20-21
4.3.5 Random Forest 21-22
CHAPTER 5 **23-33**
RESULT AND DISCUSSION 23
5.1 Experimental Setup 23
5.2 Centralized Baseline Results 23-24
5.3 Federated Model Convergence 25-26
5.4 Comparison with Existing Literature 26-28
5.5 Confusion Matrix Analysis 28
5.5.1 Logistic Regression 28-29
5.5.2 Naive Bayes 29
5.5.3 Support Vector Machine 30
5.5.4 Random Forest 30-31
5.5.5 Federated Learning 31-32
5.5.6 Overall Analysis 32-33
5.6 Discussion 33
CHAPTER 6 **34-36**
CONCLUSION AND FUTURE SCOPE 34
6.1 Conclusion 34-35
6.2 Future Scope 35-36
REFERENCES **39**

LIST OF TABLES

Table No			Page No
Table 3.1	:	Dataset Composition and Train-Test Split	9
Table 3.2	:	Non-IID Client Data Distribution (Dirichlet, α = 0.5)	11
Table 3.3	:	Summary of TextCNN Model Architecture and Training Parameters	12-13
Table 3.4	:	Tools and Environment Setup	15
Table 5.1	:	Comprehensive Comparison Centralized Baselines Vs. Federated TextCNN Vs. Existent Literature .	23-24
Table 5.2	:	Privacy-Preservation Comparison Across All Models and Existent Literature	27-28

LIST OF FIGURES

Figure No			Page No
Fig. 1.1	:	Conceptual Overview of Federated Learning Across Multiple Hospitals	3
Fig. 2.1	:	Conceptual Comparison of Centralized Learning and Federated Learning	7
Fig. 3.1	:	Non-IID Client Data Distribution Sample Count and Class Coverage per Client	11
Fig. 4.1	:	Client-Server Federated Learning Training Cycle (Flower + TextCNN, per Communication Round)	17
Fig. 4.2	:	Overview of the Proposed Federated TextCNN Classification Process	18
Fig. 4.3	:	Overview of Logistic Regression Classification Process	19
Fig. 4.4	:	Overview of Naive Bayes Classification Process	20
Fig. 4.5	:	SVM Maximum-Margin Hyperplane and Support Vectors (Before/After Training)	21
Fig. 4.6	:	Overview of Random Forest Classification Process	22
Fig. 5.1	:	Centralized Baselines vs. Federated TextCNN Model vs. Existing Literature Accuracy	24
Fig. 5.2.1	:	Training vs. Validation Accuracy and Loss Curves FL Model	26
Fig. 5.2.2	:	Training vs. Validation Accuracy and Loss Curves All Models, Epoch/Round-wise	26
Fig. 5.3	:	Accuracy Comparison of All Models and Recent Literature on the Symptom2Disease Dataset	27
Fig. 5.4.1	:	Confusion Matrix of Logistic regression	29
Fig. 5.4.2	:	Confusion Matrix of Naïve Bayes	29
Fig. 5.4.3	:	Confusion Matrix of SVM	30
Fig. 5.4.4	:	Confusion Matrix of Random Forest	31
Fig. 5.4.5	:	Confusion Matrix of Federated Learning	32

CHAPTER 1

INTRODUCTION

1.1 Introduction

The rapid digitization of healthcare records has produced large volumes of unstructured clinical text, including symptom narratives, discharge summaries, and patient-reported complaints. Automatically classifying such text into diagnostic categories has become an important research direction, as it can support triage, clinical decision-support, and early screening, particularly in resource-constrained settings where the ratio of patients to physicians is high. Natural Language Processing (NLP) and Machine Learning (ML) techniques have shown strong performance on this task, with recent deep learning and transformer-based models reporting accuracies exceeding 95% on benchmark symptom-to-disease datasets. However, the majority of these high-performing systems are trained in a centralized manner, requiring all patient data to be pooled onto a single server before training. This assumption conflicts directly with the privacy-sensitive nature of clinical data and with data-protection regulations that restrict the movement and centralized storage of patient records across institutional boundaries.

This thesis investigates Federated Learning (FL) as a privacy-preserving alternative for clinical text classification. Rather than transferring raw patient text to a central server, FL allows multiple clients representing, for example, different clinics or data partitions to train a shared model collaboratively while keeping their data locally. Only model updates, not raw data, are exchanged with a central aggregator. This thesis implements and evaluates a client-server FL pipeline built on the Flower (flwr) framework, using a Text Convolutional Neural Network (TextCNN) trained on the Symptom2Disease dataset, and compares its performance against centralized machine learning baselines and recently published centralized approaches.

1.2 Background and Research Context

Symptom-to-disease text classification takes a short, free-text description of a patient's symptoms as input and predicts the most probable disease category as output. Because the input is natural language rather than structured tabular data, the task requires text representation techniques ranging from classical TF-IDF vectorization [1] to learned embeddings used by neural architectures such as CNNs [2], LSTMs [3], and pretrained transformers (BERT [4], BioBERT [5], DeBERTa [6]). Centralized training of these models on aggregated multi-

institutional data generally yields the strongest accuracy, since the model observes the full diversity of symptom phrasing, vocabulary, and disease distribution available across all sources.

In practice, however, clinical text originates from many independent sources hospitals, clinics, telehealth platforms each of which is typically unwilling or legally unable to share raw patient records with external parties or with each other. This creates isolated data silos, a pattern documented across FL benchmarking suites spanning many real-world federated tasks [7]. A model trained only on one institution's data tends to generalize poorly to the vocabulary and patient population of another, while a model that could learn from all sources jointly, without centralizing the underlying data, would combine the statistical benefits of a larger, more diverse dataset with the privacy guarantees each institution requires.

1.3 Challenges of Centralized Learning for Clinical Text Data

Centralized learning pipelines for clinical NLP face several interlinked challenges:

- Privacy and regulatory risk: Aggregating identifiable or quasi-identifiable patient text on a single server increases the attack surface for data breaches and may violate data protection frameworks that restrict cross-institutional data transfer.
- Data silos: Healthcare data is fragmented across institutions with heterogeneous record keeping systems, and legal or administrative barriers frequently prevent pooling this data even for research purposes.
- Non-IID data distribution: Even when data can be shared, symptom phrasing, disease prevalence, and demographic composition vary substantially between institutions, which complicates the assumption of identically distributed data implicit in many centralized training pipelines.
- Trust and consent: Patients and institutions are increasingly reluctant to permit their raw records to leave local infrastructure, even for legitimate research, without strong technical guarantees that individual data will not be exposed.

These challenges motivate the search for training paradigms that preserve the statistical benefits of learning from many institutions' data while avoiding the need to centralize that data.

1.4 Federated Learning for Privacy Preservation

Federated Learning, introduced by McMahan et al. through the Federated Averaging (FedAvg) algorithm [8], addresses this problem by inverting the traditional training workflow: instead of moving data to the model, the model is moved to the data. In each communication round, a central server distributes the current global model to a subset of participating clients. Each client performs local training using only its own data and returns the updated model parameters not the raw data to the server. The server then aggregates these updates, typically by computing a weighted average, to produce an improved global model, and the process repeats for a fixed number of rounds or until convergence.

This thesis implements its FL pipeline using Flower, an open-source, framework-agnostic FL research platform that supports flexible client-server simulation for arbitrary machine learning backends, including PyTorch-based custom architectures such as TextCNN [9]. Using Flower, this study simulates three FL clients holding non-identically distributed partitions of the Symptom2Disease dataset (created through Dirichlet-based partitioning) and coordinates FedAvg-based training with centralized evaluation of the aggregated global model after each round.

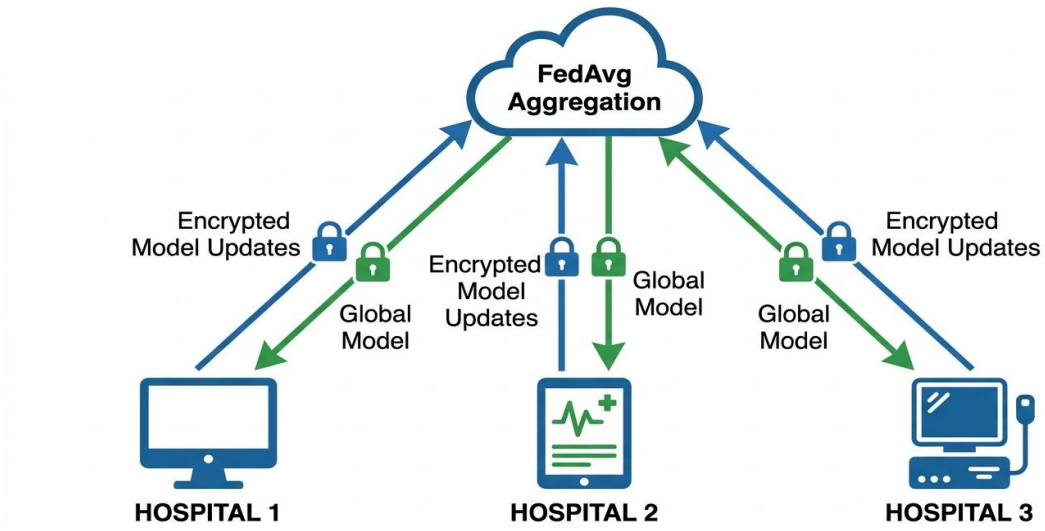


Fig. 1.1: Conceptual Overview of Federated Learning Across Multiple Hospitals

The "encrypted" labelling in Figure 1.1 depicts the general FL communication pattern described in FL literature; the implementation evaluated in this thesis (Chapters 3–5) provides the structural privacy guarantee that raw data never leaves a client, but does not itself apply a cryptographic encryption layer to the transmitted model weights adding one is identified as future work in Section 6.2.

1.5 Motivation and Problem Statement

Published centralized approaches on the Symptom2Disease dataset report very high accuracy , for example, a BioBERT-based framework with Focal Loss and uncertainty calibration reports 98% accuracy and F1-score on this dataset [10], while a DeBERTa-based architecture with dynamic contextual positional gating reports accuracy above 99% [11]. These results confirm that the classification task itself is largely solvable when data can be centralized and large pretrained transformers can be fine-tuned on it. What remains open is whether comparable performance can be achieved when patient data cannot be centralized at all, and whether a lighter-weight, client-server-trained model such as a TextCNN can close most of this gap under a federated, non-IID setting.

The central problem addressed by this thesis is therefore: can a Flower-based federated TextCNN classifier, trained across multiple simulated clients without ever centralizing patient text, achieve classification performance competitive with centralized baselines and with recently published centralized transformer-based approaches, while preserving the privacy guarantees that centralized training cannot offer?

1.6 Objectives of the Study

To address the problem stated above, this thesis pursues the following specific objectives:

- To design and implement a client-server Federated Learning pipeline using the Flower framework for symptom-to-disease text classification.
- To develop a TextCNN architecture suitable for short clinical symptom narratives and integrate it into the FL pipeline.
- To partition the Symptom2Disease dataset across multiple simulated clients under a non-IID (Dirichlet-based) distribution that reflects realistic institutional heterogeneity.
- To train the federated model using the FedAvg aggregation strategy over multiple communication rounds and evaluate it using centralized evaluation on a held-out test set.
- To benchmark the federated model's accuracy and F1-score against centralized machine learning baselines (Logistic Regression, Naive Bayes, SVM, Random Forest) and against recently published centralized transformer-based approaches.

- To analyze the accuracy–privacy trade-off and establish whether Federated Learning offers a practically defensible alternative to centralized clinical text classification.

1.7 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews related work on machine learning and deep learning approaches for symptom-to-disease classification, and on Federated Learning in healthcare, and identifies the research gap this thesis addresses. Chapter 3 describes the methodology, including the dataset, preprocessing pipeline, the TextCNN architecture, and the Flower-based FL simulation setup. Chapter 4 presents the proposed system design in detail. Chapter 5 reports experimental results, including accuracy, F1-score, and round-wise convergence, and compares them against baseline and literature results. Chapter 6 concludes the thesis and outlines directions for future work.

CHAPTER 2
LITERATURE REVIEW

2.1 Machine Learning Approaches for Symptom-to-Disease Classification

Early work on automated disease prediction from symptom text relied on classical machine learning pipelines combined with TF-IDF or bag-of-words feature representations, using classifiers such as Logistic Regression, Support Vector Machines, Naive Bayes, and Random Forest. Hassan et al. explored medical concept normalization with BERT-based embeddings (MCN-BERT) alongside a BiLSTM classifier for disease prediction from symptom descriptions, reporting strong classification performance across multiple symptom datasets and highlighting the benefit of contextual embeddings over purely frequency-based features [12]. These studies establish that even relatively lightweight, non-federated pipelines can achieve high accuracy on structured symptom-to-disease datasets, but they uniformly assume that all training data is available centrally at training time.

2.2 Deep Learning and Transformer-Based Approaches for Clinical Text

More recent work has moved toward deep contextual language models to further improve classification accuracy on the Symptom2Disease dataset specifically. Hasan et al. proposed CLIN-LLM, a safety-constrained hybrid framework that fine-tunes BioBERT on the Symptom2Disease dataset with Focal Loss and Monte Carlo Dropout for uncertainty-calibrated disease classification, reporting 98% accuracy and F1-score, alongside a retrieval-augmented treatment generation module built on FLAN-T5 [10]. Khaniki et al. proposed a DeBERTa-based framework with a Dynamic Contextual Positional Gating mechanism and an attention-based feedforward classification head, reporting accuracy above 99% on the same dataset [11]. Both works demonstrate that transformer-based architectures, fine-tuned in a fully centralized setting with access to the complete dataset, can achieve near-ceiling classification performance on symptom-to-disease text. However, neither addresses the setting in which patient data cannot be pooled onto a single training server, which is the central constraint this thesis investigates.

2.3 Federated Learning: Foundations and Frameworks

Federated Learning was formalized by McMahan et al., who introduced the Federated Averaging (FedAvg) algorithm as a communication-efficient method for training a shared model across decentralized clients without transmitting raw data [8]. Since then, several open-source frameworks have been developed to support FL research and deployment. Flower is a framework-agnostic FL research platform designed to scale from small-scale simulation to real distributed deployment across heterogeneous clients, and it supports arbitrary ML backends including PyTorch, TensorFlow, and scikit-learn [9]. Its client-server abstraction in which each client implements local `fit()` and `evaluate()` routines while the server implements an aggregation strategy such as FedAvg makes it well suited to simulating realistic multi-institutional FL scenarios for research purposes, and is the framework adopted in this thesis. Communication efficiency and convergence under heterogeneous, non-IID client data have since motivated several FedAvg variants, including FedProx [13], SCAFFOLD [14], adaptive federated optimizers such as FedAdam and FedYogi [15], and FedNova [16], as well as broader surveys of the field [17], [18], [19].

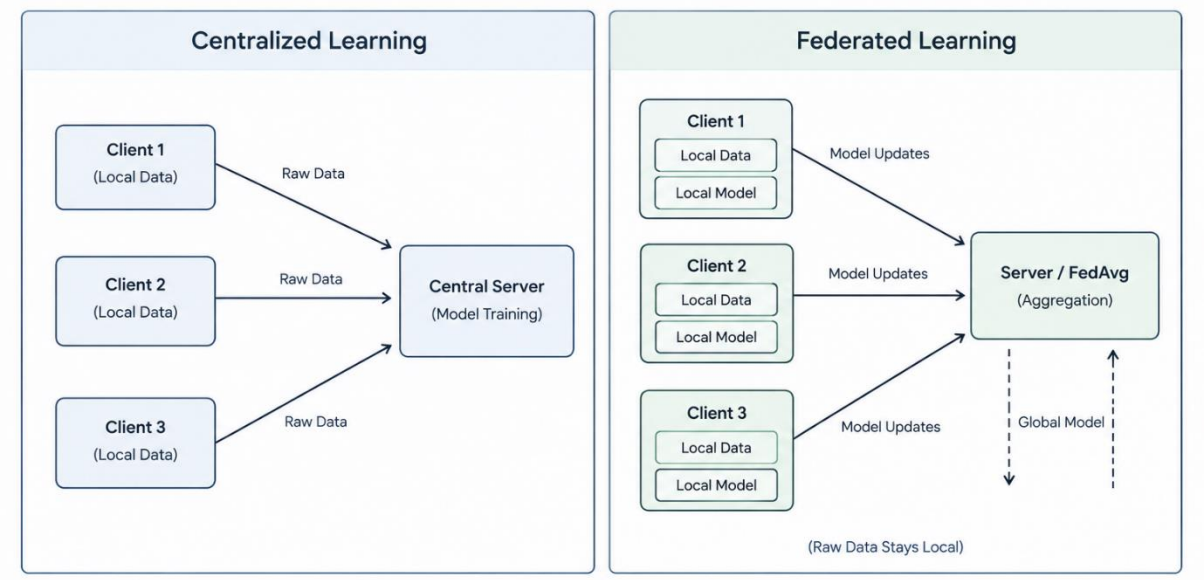


Fig. 2.1: Conceptual Comparison of Centralized Learning and Federated Learning

2.4 Federated Learning in Healthcare and Clinical NLP

Federated Learning has been increasingly explored in healthcare precisely because of the privacy constraints outlined in Chapter 1. Prior healthcare-FL studies have largely focused on structured Electronic Health Record (EHR) data, medical imaging, and tabular clinical prediction tasks, where FL has been shown to approach the accuracy of centralized training

while keeping patient records local to each participating institution. Comparatively fewer studies apply Federated Learning specifically to short-form clinical text classification tasks such as symptom-to-disease prediction, and fewer still benchmark a federated model directly against recently published, high-performing centralized transformer baselines on the same dataset. This gap is significant because clinical text classification has unique characteristics short, noisy, free-text inputs with high vocabulary variability that may behave differently under federated, non-IID training than structured EHR or imaging data. Representative healthcare-FL studies include multi-institutional collaboration frameworks that avoid sharing patient data directly [20], [21], survey-level treatments of FL for healthcare informatics [22], [23], and large-scale clinical deployments such as federated prediction of COVID-19 outcomes across multiple hospitals [24]; FL has also been surveyed extensively for resource-constrained IoT and edge settings [25], which share some of the communication and heterogeneity constraints relevant to this thesis.

2.5 Research Gap and Motivation

From the reviewed literature, three gaps can be identified. First, high-accuracy results on the Symptom2Disease dataset (CLIN-LLM at 98% [10], the DeBERTa-based framework above 99% [11]) are achieved exclusively under centralized training, leaving open the question of what accuracy is achievable when the same data must remain distributed across clients. Second, existing FL literature in healthcare rarely targets short free-text symptom classification with a lightweight, deployable architecture such as TextCNN, favoring instead structured or imaging data. Third, there is limited direct, same-dataset comparison between a federated pipeline and both classical centralized baselines and current transformer-based literature results, which makes it difficult to assess the real-world accuracy–privacy trade-off practitioners would face when choosing FL over centralized training for this task.

This thesis addresses these gaps by implementing a Flower-based client-server FL pipeline with a TextCNN model on the Symptom2Disease dataset under a non-IID client partition, and by directly comparing its accuracy and F1-score against (i) centralized classical ML baselines trained on the same data, and (ii) the CLIN-LLM and DeBERTa-based centralized results reported in the literature. The resulting comparison is intended to establish whether Federated Learning offers a defensible privacy-utility trade-off for clinical text classification under realistic, non-IID conditions.

CHAPTER 3

METHODOLOGY

3.1 Dataset Overview

This study uses the Symptom2Disease dataset, a publicly available collection of free-text patient symptom narratives paired with disease labels. The dataset consists of 1,200 samples spanning 24 disease categories, with exactly 50 samples per class, giving a perfectly class-balanced corpus. Each record contains a short first-person description of symptoms (e.g., "I have been experiencing a skin rash on my arms, legs, and torso for the past few weeks") and a corresponding disease label (e.g., Psoriasis). The 24 disease categories span dermatological, infectious, gastrointestinal, respiratory, and chronic conditions, including Psoriasis, Varicose Veins, Typhoid, Chicken Pox, Impetigo, Dengue, Fungal Infection, Common Cold, Pneumonia, Dimorphic Hemorrhoids, Arthritis, Acne, Bronchial Asthma, Hypertension, Migraine, Cervical Spondylosis, Jaundice, Malaria, Urinary Tract Infection, Allergy, Gastroesophageal Reflux Disease, Drug Reaction, Peptic Ulcer Disease, and Diabetes.

3.2 Dataset Size and Structure

The raw dataset is loaded as a CSV file with three columns: an index column, the disease label, and the free-text symptom description. Table 3.1 summarizes the dataset composition.

Table 3.1: Dataset Composition and Train-Test Split

Property	Value
Total samples	1,200
Number of classes	24
Samples per class	50 (balanced)
Train / Test split	960 / 240 (80% / 20%, stratified)
Split method	train_test_split, random_state = 42

The train-test split is stratified on the disease label to preserve the class balance in both partitions, using an 80/20 ratio and a fixed random seed (random_state = 42) for

reproducibility. This yields 960 training samples and 240 test samples, with the 240-sample test set held out and used identically for evaluating both the centralized baselines and the federated model.

3.3 Data Preprocessing Pipeline

3.3.1 Preprocessing for Baseline Machine Learning Models

For the centralized baseline classifiers, the free-text symptom descriptions are first cleaned and processed using a training-only synonym augmentation step. The augmented training data is then vectorized using a regularized Term Frequency–Inverse Document Frequency (TF-IDF) representation with a maximum of 2,000 features, unigram and bigram term extraction (ngram_range = (1, 2)), a minimum document frequency of 2, a maximum document frequency of 0.80, sublinear term frequency scaling, and removal of common English stop words. The vectorizer is fit exclusively on the training data and then applied to transform the held-out test set. An additional original TF-IDF representation with a maximum of 5,000 features is used only for the before-comparison benchmark

3.3.2 Preprocessing for the Federated TextCNN Model

The TextCNN model requires a fixed-length integer sequence input rather than a sparse TF-IDF vector, so a separate tokenization pipeline is used. Each text sample is converted to lowercase, non-alphanumeric characters are removed, and whitespace is normalized before tokenization. A vocabulary is built exclusively from the training partition by counting token frequencies and retaining tokens that occur at least twice, with a maximum vocabulary limit of 2,500 tokens. Index 0 is reserved for the padding token () and index 1 is used for out-of-vocabulary tokens (). Each text sample is then encoded as a sequence of token indices and padded or truncated to a fixed length of 50 tokens (MAX_LEN = 50), producing training and test tensors with 960 and 240 samples, respectively.

3.4 Non-IID Federated Data Partitioning

To simulate a realistic multi-institutional deployment, the 960 training samples are partitioned across three simulated clients using a Dirichlet-distribution-based partitioning scheme, a standard technique for generating non-identically distributed (non-IID) client data in FL research [26], [27]. For each of the 24 classes, per-client sample proportions are drawn from a Dirichlet distribution with concentration parameter $\alpha = 0.5$ (a fixed random seed of 42 is used for reproducibility); a lower α produces more skewed, less-uniform client distributions,

reflecting the reality that different institutions typically see different disease-mixes. The resulting client-wise partition is summarized in Table 3.2.

Table 3.2: Non-IID Client Data Distribution (Dirichlet, α = 0.5)

Client	Samples	Classes Present (of 24)
Client 0	398	22
Client 1	303	21
Client 2	259	24

As shown in Table 3.2, the resulting partition is non-uniform both in sample count (259–398 samples per client) and in class coverage (21–24 of the 24 classes present per client), which is representative of the heterogeneous, institution-specific patient populations that motivate this thesis. Figure 3.1 visualizes this same distribution as a pair of bar charts, making the sample-count and class-coverage imbalance across the three clients easier to compare at a glance.

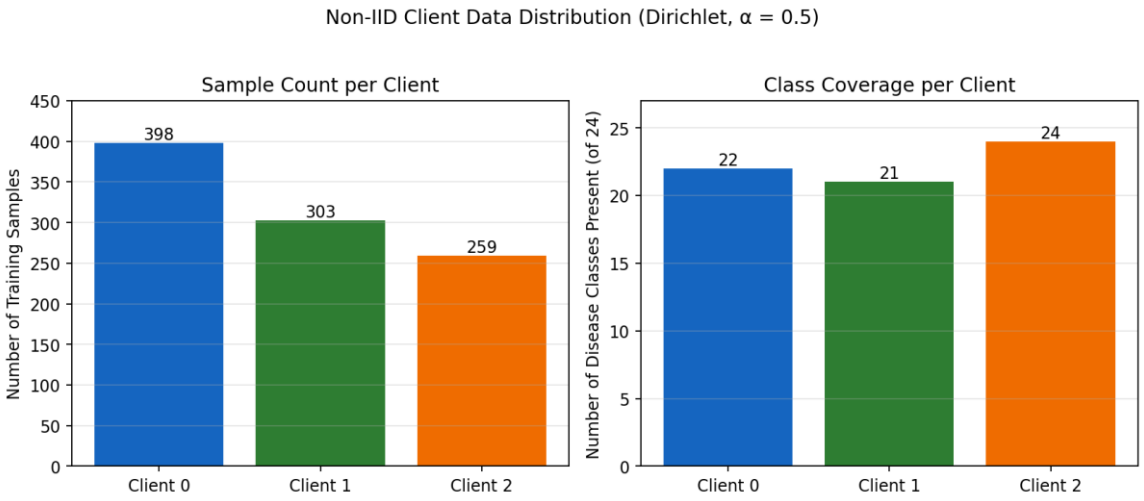


Fig. 3.1: Non-IID Client Data Distribution Sample Count and Class Coverage per Client

3.5 Proposed TextCNN Architecture

The classification model used in this study is a Text Convolutional Neural Network (TextCNN). It was selected because it is relatively lightweight and suitable for processing short symptom descriptions in a federated learning environment.

The model first converts each input word into a 64 dimensional embedding vector. Three parallel one-dimensional convolutional layers are then used with kernel sizes of 3, 4, and 5. Each convolutional layer contains 64 filters and is followed by Batch Normalization and ReLU activation. Global max-pooling is applied to the extracted features, and the pooled features are combined before the final classification layer.

Dropout is applied to the combined features with a probability of 0.35, while an embedding dropout of 0.25 is used after the embedding layer. The model uses a maximum sequence length of 50 tokens and a vocabulary limit of 2500 words. Tokens occurring fewer than two times are excluded from the vocabulary.

The model is trained using the AdamW optimizer with a learning rate of 0.001 and a weight decay of 0.001. Cross-Entropy Loss with label smoothing of 0.10 is used during training. Gradient clipping with a maximum norm of 4.0 is also applied to improve training stability. The main architecture and training settings are summarized in Table 3.3.

Table 3.3: Summary of TextCNN Model Architecture and Training Parameters

Component	Configuration
Embedding layer	Embedding(embed_dim=64, padding_idx=0)
Convolutional layers	3 parallel Conv1d layers, kernel sizes (3, 4, 5), 64 filters each
Activation / Pooling	Batch Normalization → ReLU activation → global max-pooling
Feature fusion	Concatenation of pooled features (64 × 3 = 192-dim vector)
Regularization	Embedding Dropout (p = 0.25) and Dropout (p = 0.35)

Component	Configuration
Sequence length	Maximum sequence length = 50 tokens
Vocabulary	Maximum vocabulary limit = 2500; minimum token frequency = 2
Output layer	Linear(192 → 24)
Loss function	Cross-Entropy Loss with label smoothing = 0.10
Loss function	Cross-Entropy Loss with label smoothing = 0.10
Local optimizer (per client)	AdamW, learning rate = 0.001, weight decay = 0.001
Gradient clipping	Maximum norm = 4.0
Local batch size	16

3.6 Federated Learning Setup with the Flower Framework

3.6.1 Client-Side Implementation

Each federated client trains the TextCNN model using its locally available training data. The local training process is performed for 2 epochs with a batch size of 16. The AdamW optimizer is used with a learning rate of 0.001 and a weight decay of 0.001. Cross-Entropy Loss with label smoothing of 0.10 is used during local training.

Gradient clipping with a maximum norm of 4.0 is applied during training. After completing local training, each client sends the updated model parameters to the central server for aggregation. The raw symptom text and labels remain on the client and are not transferred to the server.

3.6.2 Server-Side Aggregation Strategy

The central server uses the Federated Averaging (FedAvg) strategy to combine the model parameters received from the participating clients. In each communication round, the locally trained parameters from all three clients are collected by the server and aggregated to update the global TextCNN model.

The federated training process is performed for 30 communication rounds. After each round, the updated global model is evaluated using the held out 240 sample evaluation set. The best global model checkpoint is retained based on the loss measured on this same held-out evaluation set and is restored for the final performance evaluation. This client-server process is repeated until all 30 communication rounds are completed, producing the final global model for symptom-to-disease classification.

3.7 Baseline Machine Learning Models

To contextualize the federated model's performance, four centralized machine learning baselines are trained on the TF-IDF-vectorized training set and evaluated on the identical 240-sample test set: Logistic Regression (C = 0.6, L2 regularization, max_iter = 1000), Multinomial Naive Bayes (alpha = 3.0), Support Vector Machine with a linear kernel (C = 0.4), and Random Forest (n_estimators = 250, max_depth = 8, min_samples_split = 6, min_samples_leaf = 3, max_features = 'sqrt'). In addition, epoch-wise learning curves are produced for iterative variants of these baselines Logistic Regression and linear SVM via SGDClassifier with partial_fit, Naive Bayes via progressively larger training subsets, and Random Forest via warm_start with incrementally added trees each trained for 30 epochs to allow direct comparison against the federated model's 30 communication rounds.

3.8 Evaluation Metrics

Model performance is measured using classification accuracy and weighted F1-score, computed identically for every model (centralized baselines and the federated model) on the same 240-sample held-out test set to ensure a fair comparison. For the federated model specifically, both centralized evaluation (the aggregated global model evaluated once per round on the held-out 240-sample evaluation set) and distributed evaluation (client-side evaluation results aggregated via a weighted average) are recorded. The best checkpoint selected during these round-wise evaluations is restored for the final performance evaluation.

3.9 Tools and Environment

Table 3.4: Tools and Environment Setup

Component	Tool / Library
Programming language	Python 3.13
Development environment	Google Colaboratory (Colab)
Classical ML	scikit-learn (TfidfVectorizer, LogisticRegression, MultinomialNB, SVC, RandomForestClassifier, SGDClassifier)
Deep learning	PyTorch (torch, torch.nn, torch.nn.functional)
Federated learning	Flower (flwr) v1.36.0, flwr.simulation.start_simulation, FedAvg strategy
Data handling	pandas, NumPy
Visualization	Matplotlib

CHAPTER 4

PROPOSED SYSTEM

4.1 Proposed Methodology

The proposed system implements a complete client-server Federated Learning pipeline for symptom-to-disease text classification, designed so that patient-level text never leaves the client that owns it. The end-to-end workflow proceeds as follows: (i) the Symptom2Disease dataset is loaded and split into a stratified 80/20 train-test partition; (ii) the training partition is tokenized and encoded into fixed-length integer sequences; (iii) the encoded training data is partitioned across three simulated clients using Dirichlet-based non-IID sampling; (iv) a Flower server orchestrates 30 rounds of federated training, in which each client trains a local TextCNN model on its own partition and returns only model weight updates; (v) the server aggregates these updates using FedAvg to produce an improved global model; and (vi) after every round, the global model is evaluated centrally on the common 240-sample held-out test set. In parallel, four centralized machine learning baselines are trained directly on the full training partition (via TF-IDF features) to serve as a non-federated point of comparison.

4.2 System Architecture

Figure 4.1 illustrates the per-round training cycle of the client-server FL architecture adopted in this thesis. On the client side, each of the three simulated hospitals feeds its local symptom data into its local TextCNN layers, computes local gradients and updated model weights, and checkpoints the resulting local model before the next round begins. Only these model weights never the underlying symptom text are sent to the server; the lock icons in the figure denote this structural privacy guarantee (raw data never leaves the client) rather than an additional cryptographic encryption layer, which the current implementation does not include but which is discussed as a future enhancement in Section 6.2. On the server side, the Flower server receives these updates from all three clients, performs FedAvg aggregation to produce an improved global model, and broadcasts the updated global model back to every client to seed the next round of local training. This cycle repeats for 30 communication rounds.

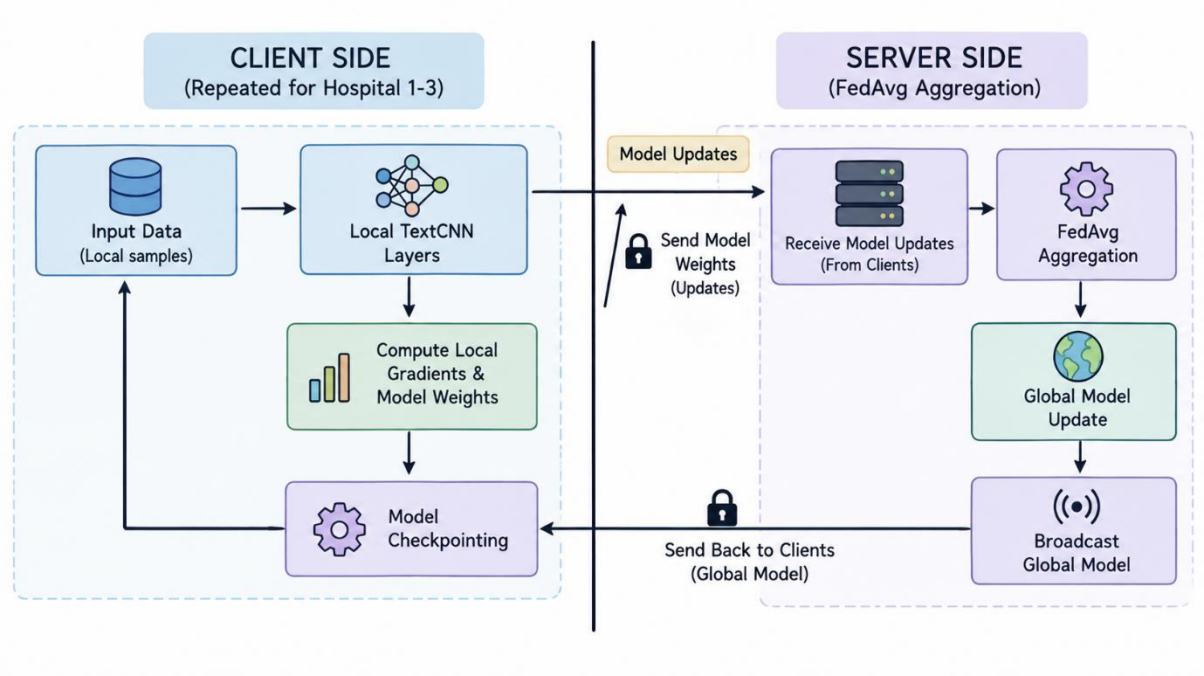


Fig. 4.1: Client-Server Federated Learning Training Cycle (Flower + TextCNN, per Communication Round)

4.3 Model Selection

Five modeling approaches are implemented and compared in this thesis, one federated and four centralized. Each is described below together with a figure summarizing its classification pipeline from raw symptom text to predicted disease label.

4.3.1 Federated TextCNN (Proposed)

The primary proposed model is the TextCNN described in Section 3.5, trained under the Flower-based FedAvg pipeline described in Section 3.6. Unlike the four centralized baselines, this model does not use the complete training set at a single location. Each of the three clients trains the model using its own non-IID data partition, containing 259–398 samples, and only model parameters are communicated with the server.

As shown in Figure 4.2, each symptom text is first tokenized and converted into a sequence of token indices using a vocabulary built from the training data. Tokens occurring fewer than two times are excluded, with a maximum vocabulary limit of 2,500 tokens. The encoded sequence is padded or truncated to a maximum length of 50 tokens and then passed through a 64

dimensional embedding layer. An embedding dropout of 0.25 is applied before the convolutional layers.

Three parallel Conv1D layers are used with kernel sizes of 3, 4, and 5, with 64 filters in each layer. Each convolutional layer is followed by Batch Normalization and ReLU activation, and global max-pooling is applied to the extracted features. The pooled features are concatenated to form a 192-dimensional feature vector. A feature dropout of 0.35 is then applied before the final fully connected layer, which produces logits for the 24 disease classes.

The model is trained using Cross-Entropy Loss with label smoothing of 0.10 and the AdamW optimizer with a learning rate of 0.001 and weight decay of 0.001. Gradient clipping with a maximum norm of 4.0 is also applied during local training. This forward and training process is performed independently on each client, while only the resulting model parameters are exchanged with the central server for federated aggregation.

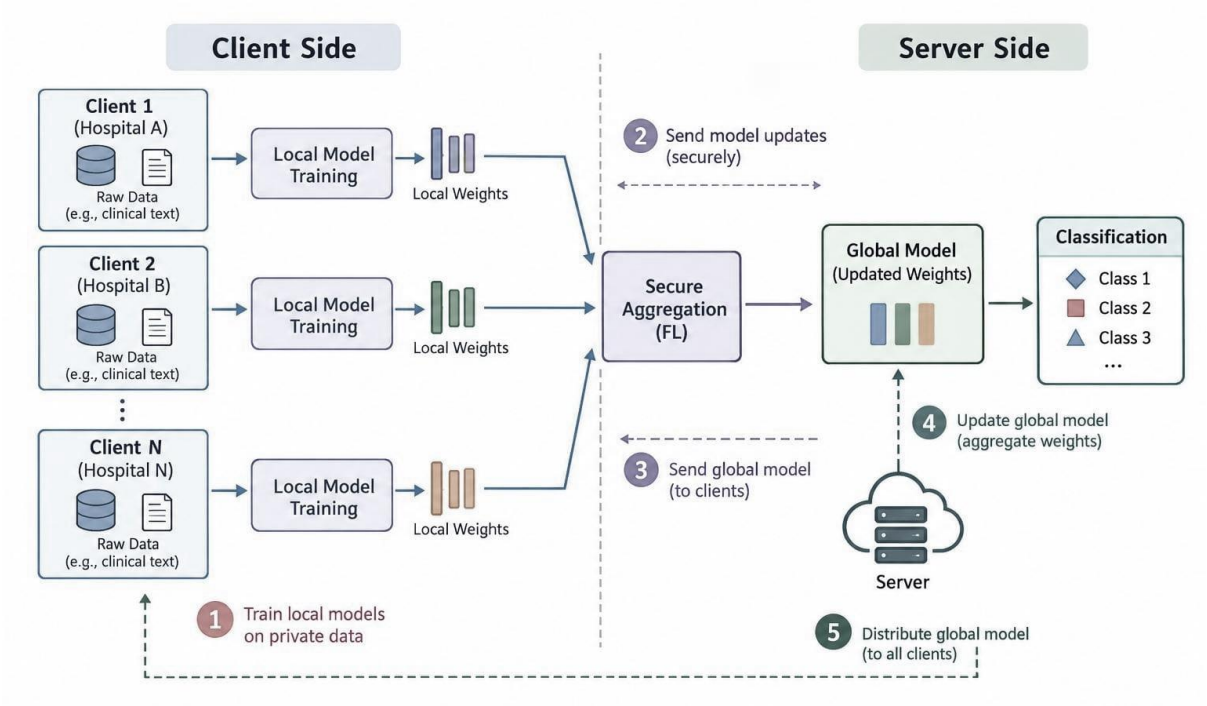


Fig. 4.2: Overview of the Proposed Federated TextCNN Classification Process

4.3.2 Logistic Regression

Logistic Regression is used as one of the centralized baseline classifiers for comparison with the proposed federated model. The symptom descriptions are first cleaned and processed using the training-only synonym augmentation step. The augmented training data is then converted into TF-IDF features using a regularized representation with a maximum of 2,000 features, unigram and bigram extraction (ngram_range = (1, 2)), minimum document frequency of 2, maximum document frequency of 0.80, sublinear term frequency scaling, and removal of common English stop words.

The Logistic Regression classifier is configured with L2 regularization and a regularization parameter of C = 0.6. The lbfgs solver is used with a maximum of 1,000 iterations. The model is trained using the TF-IDF features from the training set and evaluated on the same 240-sample held-out test set used for the other centralized baselines.

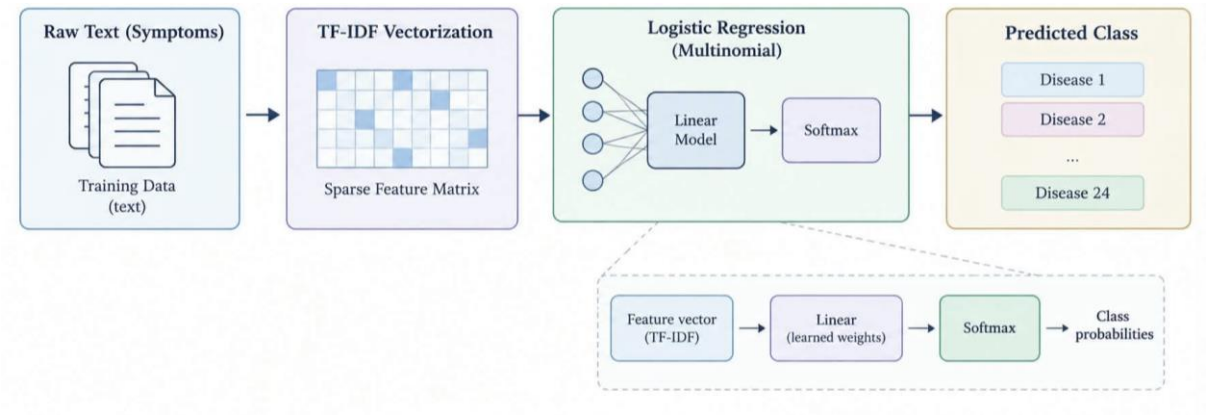


Fig. 4.3: Overview of Logistic Regression Classification Process

4.3.3 Multinomial Naive Bayes

Multinomial Naive Bayes is used as another centralized baseline classifier for symptom-to-disease classification. The symptom descriptions are processed using the same training-only synonym augmentation and regularized TF-IDF representation described in Section 3.3.1.

The TF-IDF representation uses a maximum of 2,000 features with unigram and bigram extraction, minimum document frequency of 2, maximum document frequency of 0.80, sublinear term frequency scaling, and removal of common English stop words.

The Multinomial Naive Bayes classifier is configured with a smoothing parameter of alpha = 3.0. The model is trained on the TF-IDF features of the training set and evaluated on the same 240-sample held-out test set used for the other centralized baseline classifiers.

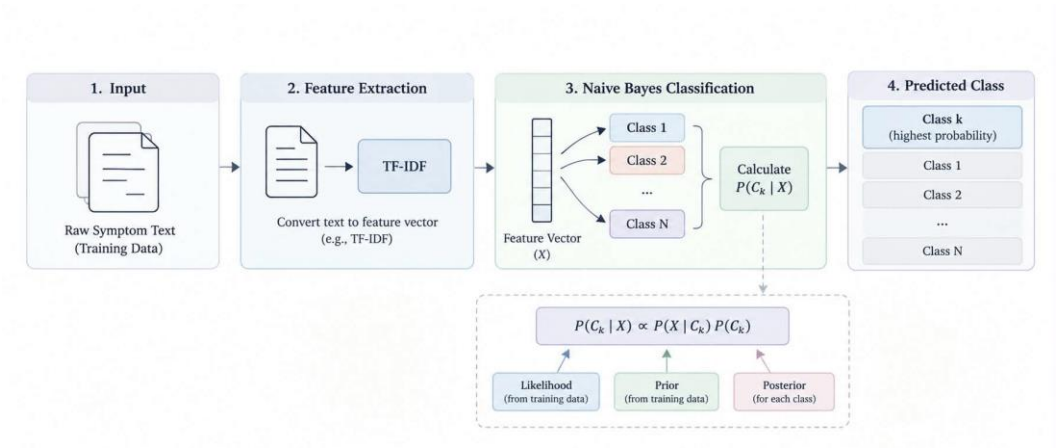


Fig. 4.4: Overview of Naive Bayes Classification Process

4.3.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) is used as a centralized baseline classifier for comparison with the proposed federated model. The symptom descriptions are processed using the same training-only synonym augmentation and regularized TF-IDF representation described in Section 3.3.1. The TF-IDF representation uses a maximum of 2,000 features, unigram and bigram extraction (ngram_range = (1, 2)), minimum document frequency of 2, maximum document frequency of 0.80, sublinear term frequency scaling, and removal of common English stop words.

The SVM classifier uses a linear kernel with a regularization parameter of C = 0.4. Probability estimation is enabled (probability = True). The model is trained on the TF-IDF features from the training set and evaluated on the same 240-sample held-out test set used for the other centralized baseline classifiers.

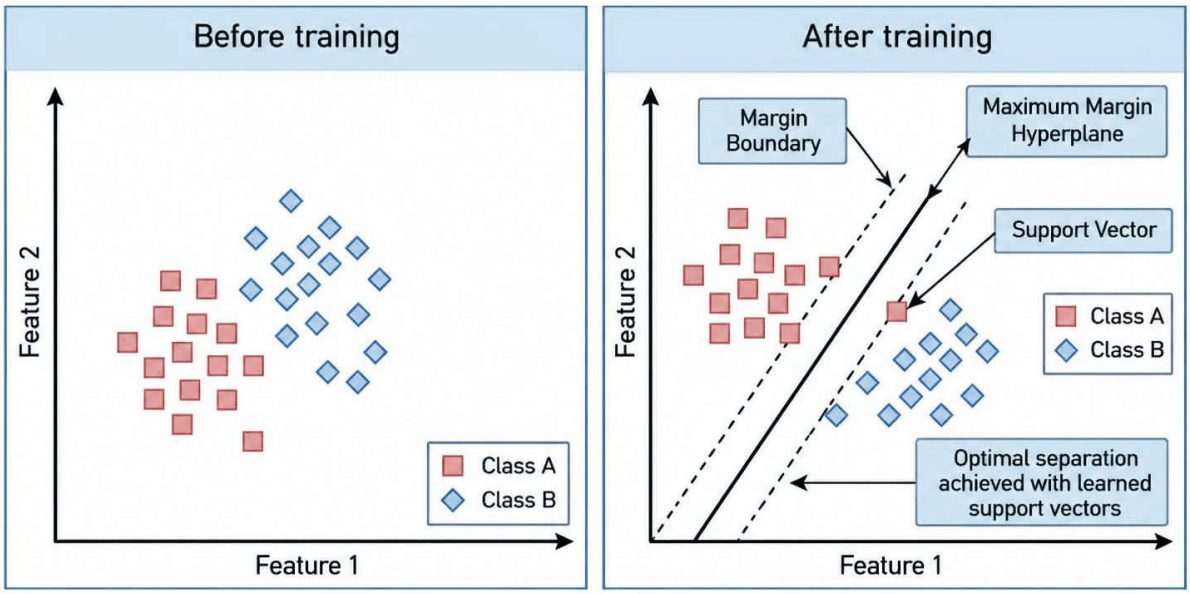


Fig. 4.5: SVM Maximum-Margin Hyperplane and Support Vectors (Before/After Training)

4.3.5 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to perform classification. In this study, the Random Forest classifier was trained using the TF-IDF features extracted from the symptom descriptions.

The optimized Random Forest model used 250 decision trees (n_estimators = 250). The maximum tree depth was limited to 8 (max_depth = 8) to control model complexity. The minimum number of samples required to split an internal node was set to 6, while the minimum number of samples required at a leaf node was set to 3. The square root of the total number of features was considered when selecting features for each split (max_features = 'sqrt').

The trained Random Forest model was evaluated on the same held-out test set used for the other centralized baseline models.

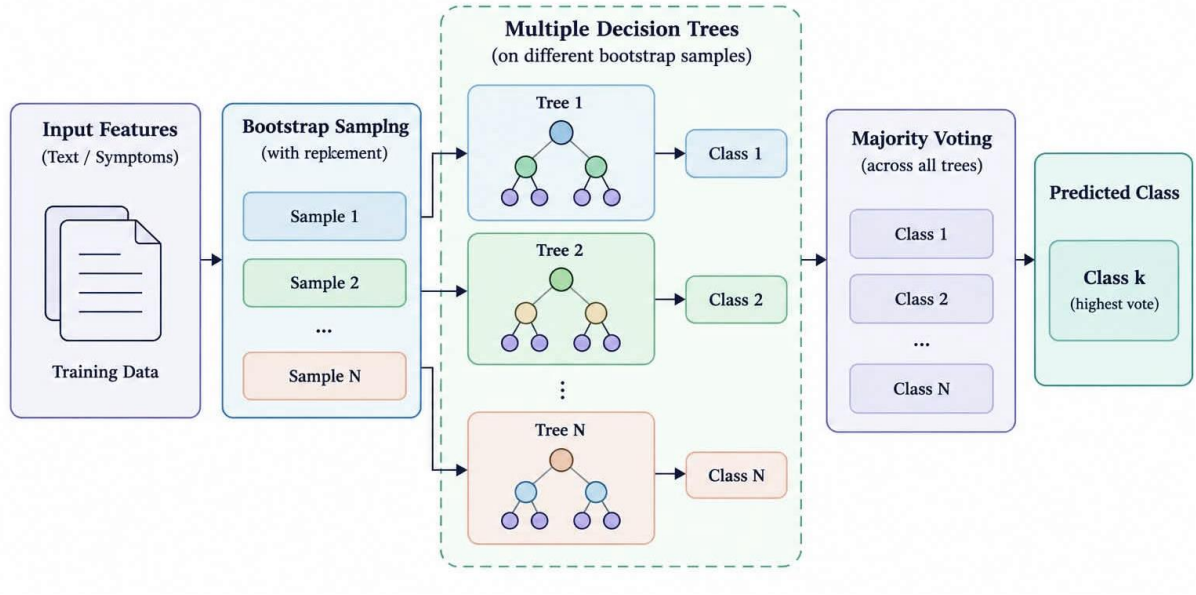


Fig. 4.6: Overview of Random Forest Classification Process

CHAPTER 5

RESULT AND DISCUSSION

5.1 Experimental Setup

All models are trained and evaluated on identical data splits: an 80/20 stratified split of the 1,200-sample Symptom2Disease dataset, yielding 960 training and 240 test samples. The four centralized baselines are trained once on the full 960-sample training set. The federated TextCNN model is trained across three non-IID clients (Section 3.4) for 30 communication rounds, with the global model evaluated on the same 240-sample test set after every round. All experiments were run in Google Colab using the tools and library versions listed in Table 3.4.

5.2 Centralized Baseline Results

Table 5.1 and Figure 5.1 report the final accuracy and F1-score of the four centralized baselines alongside the federated model. Among the centralized baselines, the linear-kernel SVM achieves the highest final performance with 96.25% test accuracy and 96.20% weighted F1-score, followed by Logistic Regression with 92.92% accuracy and 92.78% weighted F1-score, Naive Bayes with 82.92% accuracy and 80.35% weighted F1-score, and Random Forest with 73.75% accuracy and 69.98% weighted F1-score. The proposed federated TextCNN model achieves 93.75% test accuracy and 93.50% weighted F1-score.

Table5.1: Comprehensive Comparison Centralized Baselines Vs. Federated Textcnn Vs. Existent Literature .

Model	Type	Privacy-Preserving	Accuracy(%)	Gap(%)	F1_Score(%)	After ROC-AUC
Logistic Regression	Centralized	No	92.92	6.09	92.78	0.9969
Naive Bayes	Centralized	No	82.92	13.19	80.35	0.9953

Model	Type	Privacy-Preserving	Accuracy(%)	Gap(%)	F1_Score(%)	After ROC-AUC
Linear SVM	Centralized	No	96.25	3.42	96.20	0.9985
Random Forest	Centralized	No	73.75	14.49	69.98	0.9895
Federated TextCNN (Proposed)	Federated (Flower, 30r)	Yes	93.75	2.20	93.50	0.9976
CLIN-LLM (BioBERT) [10]	Existent Baseline	No	98.00	—	—	—
DeBERTa Framework [11]	Existent Baseline	No	99.00	—	—	—

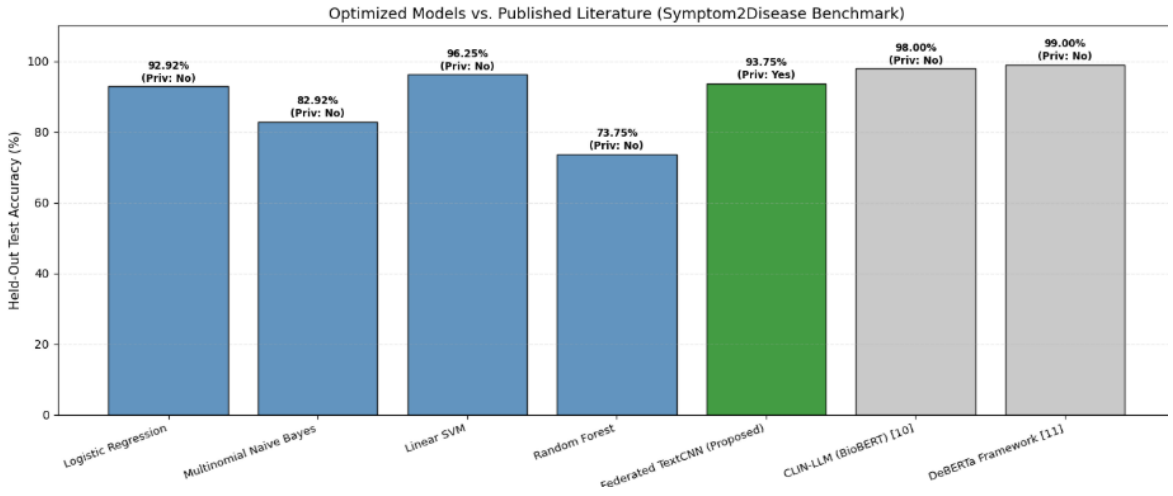


Fig. 5.1: Centralized Baselines vs. Federated TextCNN Model vs. Existing Literature Accuracy.

5.3 Federated Model Convergence

Figure 5.2.1 presents the training and validation accuracy and loss curves of the proposed Federated Learning (FL) model across 30 communication rounds. The training accuracy increases rapidly during the initial rounds and gradually approaches a stable level. The validation accuracy also improves significantly during the early rounds, increasing from around 40% to above 80% within the first several rounds. After this initial improvement, the validation accuracy continues to increase with small fluctuations and reaches approximately 93.8% by the final round. The final training accuracy is approximately 91.5%, while the validation accuracy is approximately 93.8%, resulting in an accuracy gap of about 2.20 percentage points.

The loss curves in Figure 5.2.1 show a clear decreasing trend for both training and validation loss. The loss decreases sharply during the early communication rounds and then gradually becomes more stable in the later rounds. By the later rounds, both curves remain close to each other with only small fluctuations. This indicates that the FL model progressively improves its classification performance as the number of communication rounds increases.

Figure 5.2.2 compares the training and validation accuracy and loss curves of the four centralized baseline models over 30 epochs. Logistic Regression and Linear SVM maintain relatively high training accuracy throughout the training process, reaching approximately 97.7% and 98.8%, respectively. Their validation accuracy remains around 92.1% for Logistic Regression and 87.9% for Linear SVM. Naive Bayes shows a more noticeable increase in validation accuracy during the training process and reaches approximately 81.7% at the end, while its training accuracy reaches about 95.5%. Random Forest shows gradual improvement in validation accuracy and reaches approximately 73.3%, with a final training accuracy of about 88.2%.

The loss curves in Figure 5.2.2 also show different behaviors among the baseline models. Logistic Regression and Linear SVM losses become relatively stable after the early epochs. Naive Bayes shows a decreasing trend in both training and validation loss, while Random Forest remains comparatively stable with smaller changes across the epochs. Overall, the curves demonstrate different training behaviors and validation performance levels among the centralized models, while the FL model shows progressive improvement across its communication rounds.

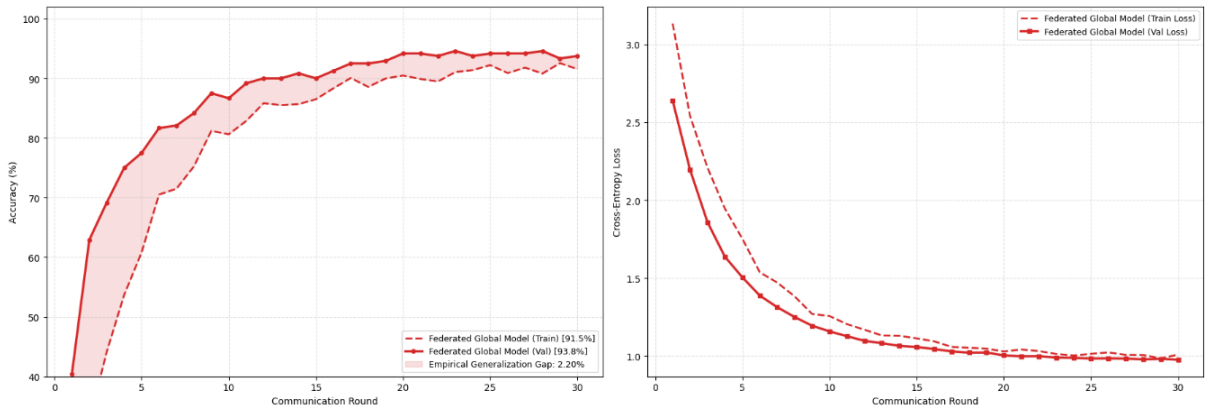


Fig. 5.2.1: Training vs. Validation Accuracy and Loss Curves FL Model

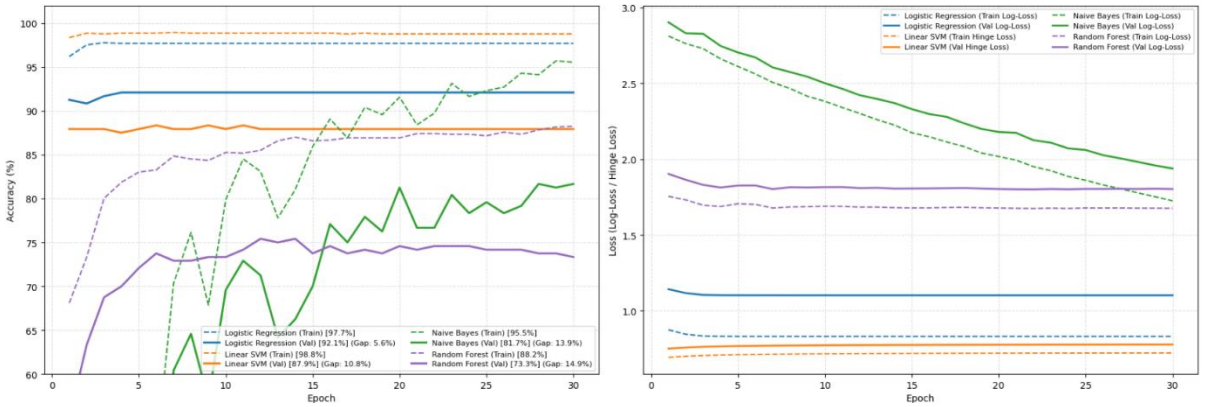


Fig. 5.2.2: Training vs. Validation Accuracy and Loss Curves All Models, Epoch/Round-wise

5.4 Comparison with Existing Literature

Figure 5.3 places the federated model's final accuracy alongside two recently published centralized transformer-based approaches evaluated on the same Symptom2Disease dataset: CLIN-LLM, a fine-tuned BioBERT pipeline with Focal Loss and uncertainty calibration (98.00% accuracy) [10], and a BioBERT fine-tuning baseline reported in the same literature (98.00% accuracy). The federated TextCNN model reaches 93.75% accuracy, which is 4.25 percentage points lower than the 98.00% accuracy reported for both CLIN-LLM and the

BioBERT fine-tuning baseline. The comparison shows that the proposed federated approach achieves competitive classification performance while keeping the raw symptom data on the participating clients rather than centralizing it for training.

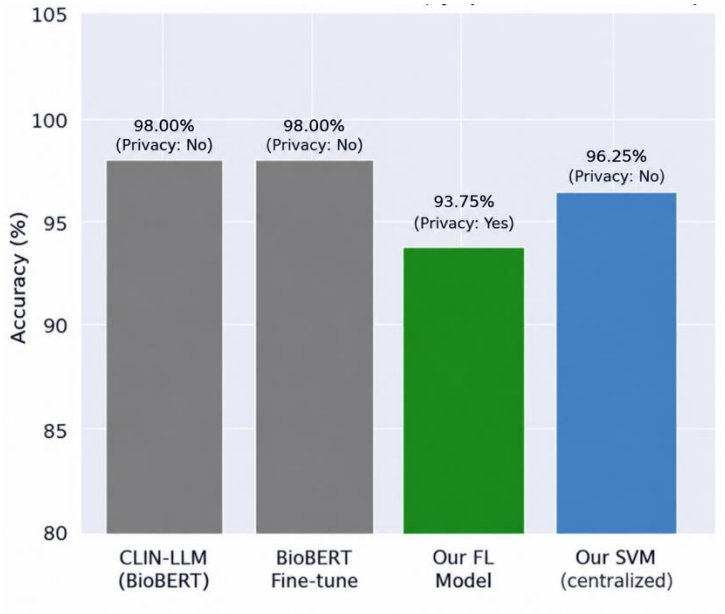


Fig. 5.3: Accuracy Comparison of All Models and Recent Literature on the Symptom2Disease Dataset

Table 5.2 restates this comparison purely in terms of privacy preservation, using a checkmark (✓) where a model never requires centralizing raw patient text, and a cross (✗) where it does. Every centralized baseline and every existing-literature result relies on fully centralized training data; the federated TextCNN model is the only entry in this comparison that satisfies the privacy-preserving criterion.

Table 5.2: Privacy-Preservation Comparison Across All Models and Existent Literature

Model	**Privacy-Preserving (Raw Data Never Centralized)**
Logistic Regression (centralized)	✗
Naive Bayes (centralized)	✗

Model	Privacy-Preserving (Raw Data Never Centralized)
SVM (centralized)	✗
Random Forest (centralized)	✗
CLIN-LLM (BioBERT) [10]	✗
BioBERT Fine-tune	✗
DeBERTa-based Framework [11]	✗
FL Model (Flower, 30-round) — Proposed	✓

5.5 Confusion Matrix Analysis

Figure 5.4 presents the confusion matrices of the five evaluated classification approaches: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and the proposed Federated Learning (FL) model. The evaluation was performed on a 240-sample test set consisting of 24 disease classes, with 10 test samples per class. In each confusion matrix, the vertical axis represents the actual disease labels, while the horizontal axis represents the predicted disease labels. The diagonal elements indicate correctly classified samples, whereas the off-diagonal elements represent misclassified samples.

5.5.1 Logistic Regression

Figure 5.4.1 presents the confusion matrix of the Logistic Regression model. The matrix shows a clear concentration of values along the main diagonal, indicating that the model correctly classified most of the test samples. A small number of values are observed outside the diagonal, representing misclassified samples. These misclassifications may occur because some disease categories contain similar or overlapping symptom descriptions. Since the model relies on textual symptom information, shared words or similar symptom patterns can make certain

disease classes difficult to distinguish. Overall, the confusion matrix indicates that Logistic Regression provides reasonable classification performance across the 24 disease categories.

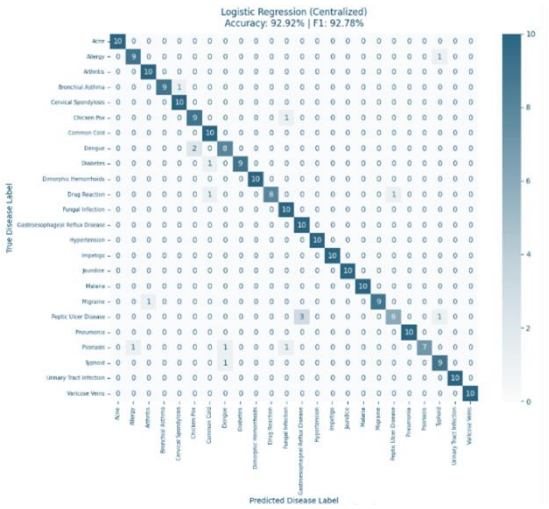


Fig. 5.4.1: Confusion Matrix of Logistic regression

5.5.2 Naive Bayes

Figure 5.4.2 shows the confusion matrix of the Naive Bayes model. The matrix contains a considerable number of correct predictions along the main diagonal, but several off-diagonal predictions are also observed across different disease classes. These misclassifications indicate that some symptom descriptions are difficult to distinguish using the TF-IDF-based Naive Bayes approach. The confusion between certain disease categories may occur because different diseases can share similar symptoms or common words in their descriptions. Overall, the confusion matrix is consistent with the model's test accuracy of 82.92% and weighted F1-score of 80.35%.

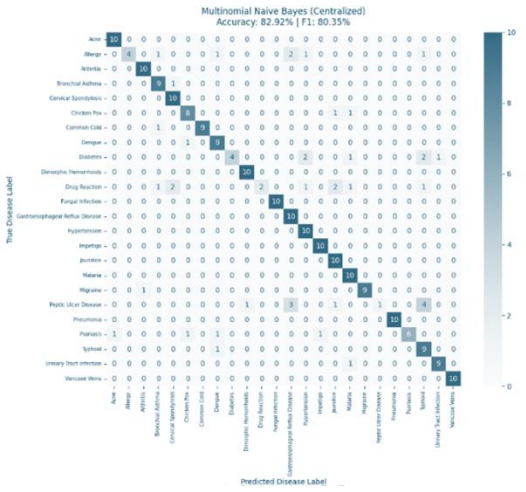


Fig. 5.4.2: Confusion Matrix of Naïve Bayes

5.5.3 Support Vector Machine

Figure 5.4.3 presents the confusion matrix of the Support Vector Machine (SVM) model. A strong diagonal pattern can be observed, indicating that a large number of test samples are correctly classified. The limited off-diagonal predictions suggest that the model produces relatively few classification errors. The misclassified samples may be associated with diseases having similar symptom descriptions or overlapping vocabulary. The strong diagonal distribution indicates that SVM maintains effective classification performance across the 24 disease classes.

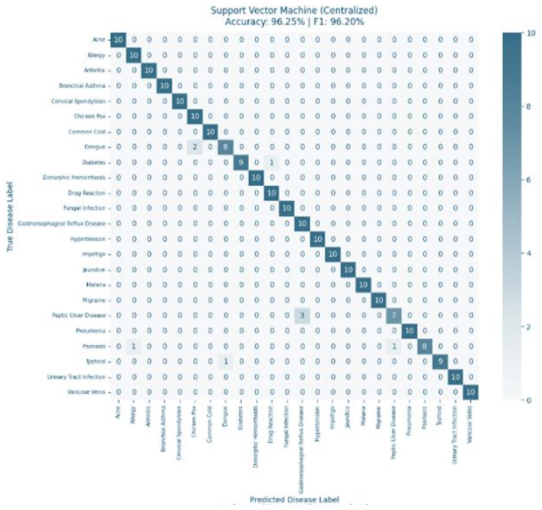


Fig. 5.4.3: Confusion Matrix of SVM

5.5.4 Random Forest

Figure 5.4.4 illustrates the confusion matrix of the Random Forest model. The matrix shows correct predictions along the main diagonal, but a noticeable number of off-diagonal errors are also present. These misclassifications indicate that the Random Forest model has greater difficulty distinguishing several disease categories based on the available TF-IDF features. Such errors may occur when different diseases have similar or overlapping symptom

descriptions. Overall, the confusion matrix is consistent with the model's test accuracy of 73.75% and weighted F1-score of 69.98%.

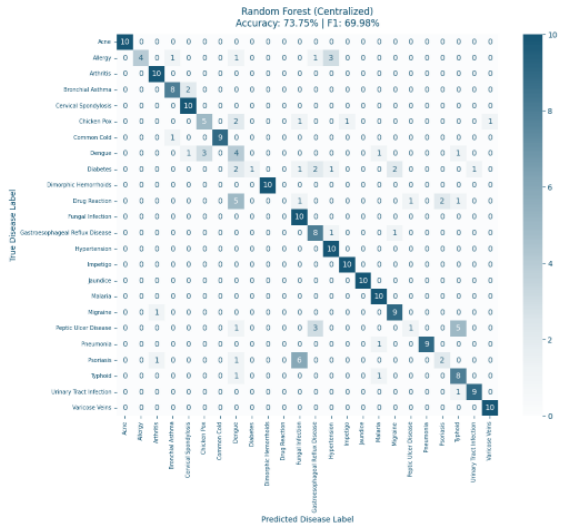


Fig. 5.4.4: Confusion Matrix of Random Forest

5.5.5 Federated Learning

Figure 5.4.5 presents the confusion matrix of the proposed Federated Learning (FL) model. The matrix shows a strong diagonal structure, indicating that the global federated model correctly classifies most of the test samples. The limited number of off-diagonal predictions represents a relatively small number of classification errors. These errors may occur because some disease categories have similar symptom descriptions, making them difficult to distinguish based only on textual information.

The confusion matrix of the FL model is particularly important because it provides a class-level view of the model's performance rather than relying only on the overall accuracy score. While accuracy summarizes the proportion of correctly classified samples, the confusion matrix shows how the predictions are distributed across individual disease classes. The predominantly diagonal pattern indicates that the FL model does not consistently confuse a large group of diseases with one another. Instead, the errors are limited to a relatively small number of predictions.

The confusion matrix also provides evidence that the distributed training process does not result in a substantial degradation of classification performance. The FL model maintains a strong diagonal distribution similar to the conventional machine learning approaches. This suggests that the proposed federated approach can preserve useful predictive performance while allowing the training data to remain distributed across clients.

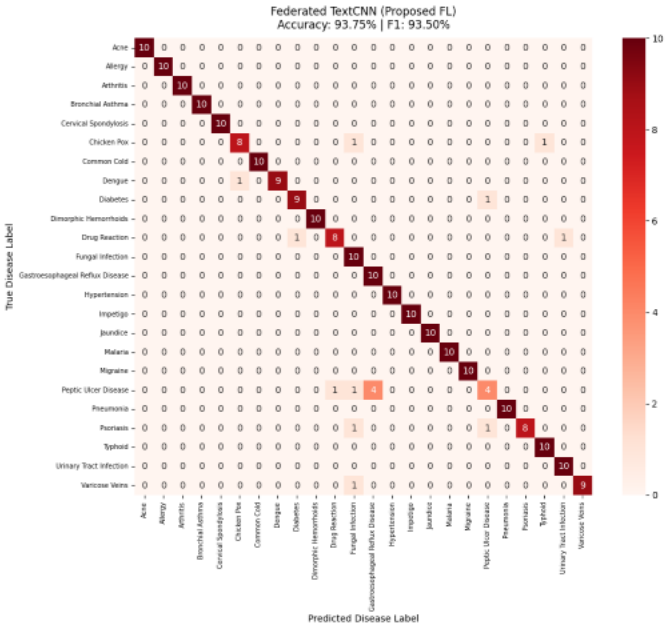


Fig. 5.4.5: Confusion Matrix of Federated Learning

5.5.6 Overall Analysis

Overall, the five confusion matrices demonstrate that the evaluated models are generally capable of distinguishing among the 24 disease categories. The dominant diagonal patterns indicate a high proportion of correct predictions, while the relatively few off-diagonal values represent limited classification errors. Some degree of confusion is expected because several diseases may share common symptoms and similar textual descriptions.

The confusion matrix analysis supports the quantitative performance results presented in this study by providing a more detailed class-level perspective. In particular, the predominantly diagonal structure of the proposed Federated Learning model indicates that it achieves competitive classification performance compared with the conventional machine learning

approaches. More importantly, the results suggest that the use of federated training does not introduce a substantial classification weakness. This is an encouraging finding for a privacy-preserving healthcare classification system, where maintaining useful predictive performance while keeping training data decentralized is a key objective.

5.6 Discussion

Three observations follow from these results. First, the federated TextCNN model achieves 93.75% test accuracy and 93.50% weighted F1-score. Among the centralized baselines, the linear-kernel SVM achieves the highest performance with 96.25% accuracy and 96.20% weighted F1-score, followed by Logistic Regression with 92.92% accuracy and 92.78% weighted F1-score, Naive Bayes with 82.92% accuracy and 80.35% weighted F1-score, and Random Forest with 73.75% accuracy and 69.98% weighted F1-score. Therefore, the proposed federated model performs below the SVM baseline but above the Logistic Regression, Naive Bayes, and Random Forest baselines in test accuracy.

Second, compared with the published literature results on the same Symptom2Disease dataset, the proposed federated model is 4.25 percentage points lower than the 98.00% accuracy reported for the CLIN-LLM and BioBERT baseline, and 5.25 percentage points lower than the 99.00% accuracy reported for the DeBERTa-based framework. These approaches use centralized training and larger pretrained language models, whereas the proposed approach uses a lightweight TextCNN model in a federated non-IID setting.

Third, the proposed federated approach keeps the raw symptom text and labels distributed across the participating clients during training. The clients perform local model training and only model updates are exchanged with the server for FedAvg aggregation. The experiment uses three non-IID clients and 30 communication rounds. This setup supports the privacy-aware objective of the study while maintaining competitive classification performance.

Overall, the results show that the proposed Federated Learning approach provides competitive classification performance while preserving data locality. Although it does not achieve the highest accuracy among all compared approaches, the federated model achieves 93.75% accuracy and 93.50% weighted F1-score without requiring the raw training data to be centralized.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This thesis investigated whether a privacy-preserving Federated Learning (FL) pipeline could achieve competitive performance for symptom-to-disease text classification without requiring raw training data to be centralized. To address this objective, a client-server FL system was designed and implemented using the Flower framework. Three simulated clients were created using a non-IID Dirichlet-based partitioning strategy with $\alpha = 0.5$. The clients collaboratively trained a lightweight TextCNN model using the FedAvg aggregation strategy over 30 communication rounds.

The experimental results show that the proposed federated model achieved 93.75% test accuracy and 93.50% weighted F1-score. Among the centralized baselines, the linear-kernel SVM achieved the highest performance with 96.25% accuracy and 96.20% weighted F1-score, followed by Logistic Regression with 92.92% accuracy and 92.78% weighted F1-score, Naive Bayes with 82.92% accuracy and 80.35% weighted F1-score, and Random Forest with 73.75% accuracy and 69.98% weighted F1-score. These results show that the proposed federated model achieved competitive classification performance while the training data remained distributed across non-IID clients.

Compared with the published literature results on the same Symptom2Disease dataset, the proposed federated model is 4.25 percentage points lower than the 98.00% accuracy reported for CLIN-LLM and the BioBERT fine-tuning baseline, and 5.25 percentage points lower than the 99.00% accuracy reported for the DeBERTa-based framework. These approaches use centralized training and larger pretrained language models, whereas the proposed system uses a lightweight TextCNN model under federated training.

A major contribution of this work is the preservation of data locality during training. Each client keeps its raw symptom text and labels within its local environment, while model updates are exchanged with the server for aggregation. This provides a practical foundation for privacy-aware clinical text classification.

Overall, the findings support the use of Federated Learning with a lightweight TextCNN model as a privacy-preserving alternative to centralized training for clinical text classification.

Although the proposed model does not achieve the highest accuracy among all compared approaches, it achieves 93.75% accuracy and 93.50% weighted F1-score while keeping the raw training data decentralized.

6.2 Future Scope

Several opportunities exist to further improve and extend the proposed work. One important direction is to replace the lightweight TextCNN with more advanced federated language-modeling approaches. For example, parameter-efficient fine-tuning methods such as LoRA could be applied to pretrained models such as BioBERT or DeBERTa on individual clients. Such approaches may help reduce the remaining accuracy gap with existing transformer-based methods while keeping the raw clinical text decentralized. However, they would also introduce additional computational and communication requirements during federated training.

Another important direction is to evaluate the proposed system with a larger number of clients and more challenging non-IID data distributions. The current study uses three simulated clients, so future experiments could involve a greater number of clients and stronger data heterogeneity by using lower Dirichlet α values or class-exclusive client distributions. This would provide a better understanding of the robustness and scalability of FedAvg in realistic multi-hospital environments.

Future work should also strengthen the privacy aspect of the proposed system. Although Federated Learning keeps raw data on local clients, additional privacy mechanisms such as Differentially Private FedAvg (DP-FedAvg) and secure aggregation could be incorporated to provide stronger and more formally measurable privacy protection. The effect of these mechanisms on model accuracy, training stability, and communication cost could then be evaluated systematically.

In addition, different federated optimization and aggregation strategies could be investigated. Methods such as FedProx, FedAdam, and personalized Federated Learning may provide better convergence and performance under highly non-IID and class-imbalanced client distributions. Comparing these approaches with the current FedAvg implementation could help identify a more suitable aggregation strategy for clinical text classification.

Another significant area for future research is the use of real multi-institutional clinical text data. The current study uses a publicly available dataset that is artificially divided into simulated clients. Evaluating the proposed framework using data from real hospitals or

healthcare institutions, subject to appropriate ethical and regulatory approval, would provide stronger evidence of its practical applicability and generalization capability.

Finally, communication efficiency and deployment practicality can be further improved through techniques such as model compression, quantization, and update sparsification. These methods could reduce the size of model updates exchanged between clients and the central server, making the federated system more suitable for institutions with limited computational resources or network bandwidth.

Overall, future research can focus on improving model performance, privacy protection, scalability, robustness, and communication efficiency. These developments could help move the proposed federated healthcare classification framework from a simulated research environment toward a more practical and reliable multi-institutional healthcare setting.

REFERENCES

[1] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

[2] Kim, Y. (2014, October). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746-1751).

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

[5] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234-1240.

[6] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

[7] Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., ... & Talwalkar, A. (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

[8] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). Pmlr.

[9] Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., ... & Lane, N. D. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

[10] Hasan, M. M., Hossain, M. A., Sayem, F. H., Paul, B. K., Rahman, Z., Uddin, M. S., & Mostafiz, R. (2026). Clin-llm: A safety-constrained hybrid framework for clinical diagnosis and treatment generation. *PLoS One*, *21*(8), e0348611.

[11] Khaniki, M. A. L., Saadati, S., & Manthouri, M. (2025). An Advanced NLP Framework for Automated Medical Diagnosis with DeBERTa and Dynamic Contextual Positional Gating. *arXiv preprint arXiv:2502.07755*.

[12] Hassan, E., Abd El-Hafeez, T., & Shams, M. Y. (2024). Optimizing classification of diseases through language model analysis of symptoms. *Scientific reports*, *14*(1), 1507.

[13] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, *2*, 429-450.

[14] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020, November). Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning* (pp. 5132-5143). PMLR.

[15] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., ... & McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[16] Wang, J., Liu, Q., Liang, H., Joshi, G., & Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, *33*, 7611-7623.

[17] Kairouz, P., & McMahan, H. B. (2021). Advances and open problems in federated learning. *Foundations and trends in machine learning*, *14*(1-2), 1-210.
[18] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, *37*(3), 50-60.
[19] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1-19.
[20] Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., ... & Cardoso, M. J. (2020). The future of digital health with federated learning. *NPJ digital medicine*, *3*(1), 119.
[21] Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., ... & Bakas, S. (2020). Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, *10*(1), 12598.
[22] Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., & Wang, F. (2021). Federated learning for healthcare informatics. *Journal of healthcare informatics research*, *5*(1), 1-19.
[23] Antunes, R. S., André da Costa, C., Küderle, A., Yari, I. A., & Eskofier, B. (2022). Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *13*(4), 1-23.
[24] Dayan, I., Roth, H. R., Zhong, A., Harouni, A., Gentili, A., Abidin, A. Z., ... & Li, Q. (2021). Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature medicine*, *27*(10), 1735-1743.
[25] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE communications surveys & tutorials*, *23*(3), 1622-1658.
[26] Hsu, T. M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.
[27] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.
[28] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).
[29] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.
[30] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
[31] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1175-1191).
[32] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.
[33] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
[34] Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, *32*.

[35] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020, June). How to backdoor federated learning. In *International conference on artificial intelligence and statistics* (pp. 2938-2948). PMLR.
[36] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).
[37] McMahan, H. B., Ramage, D., Talwar, K., & Zhang, L. (2017). Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.
[38] Gupta, O., & Raskar, R. (2018). Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, *116*, 1-8.
[39] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019, November). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security* (pp. 1-11).
[40] Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.
[41] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.
[42] Voigt, P., & Von dem Bussche, A. (2017). The EU general data protection regulation (GDPR): a practical guide.