Privacy-Preserving Machine Learning Models

A report submitted in partial fulfilment of the requirements
for the award of the degree of

Bachelor of Science
in
Electrical Engineering and Computer Science

by
ADITYA MISHRA
21013

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL
Bhopal - 462 066

April, 2025

भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान भोपाल

Indian Institute of Science Education and Research
Bhopal
(Estb. By MHRD, Govt. of India)

CERTIFICATE

This is to certify that **Aditya Mishra**, BS (Electrical Engineering and Computer Science), has worked on the project entitled **'Privacy-Preserving Machine Learning Models'** under my supervision and guidance. The content of this report is original and has not been submitted elsewhere for the award of any academic or professional degree.

April, 2025
IISER Bhopal

Dr. Haroon Lone
Assistant Professor, EECS Department
IISER Bhopal

ACADEMIC INTEGRITY AND COPYRIGHT DISCLAIMER

I hereby declare that this thesis is my own work and, to the best of my knowledge, it contains no materials previously published or written by any other person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at IISER Bhopal or any other educational institution, except where due acknowledgement is made in the thesis.

I certify that all copyrighted material incorporated into this thesis is in compliance with the Indian Copyright (Amendment) Act, 2012 and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless IISER Bhopal from any and all claims that may be asserted or that may arise from any copyright violation.

April, 2025
IISER Bhopal

Aditya Mishra

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Haroon Lone for his invaluable support throughout my research and for being my supervisor. His expertise and insights have been instrumental in my research. My heartfelt thanks go to my parents - Dr. Satanand Mishra and Mrs. Snehlata Mishra, my elder brother, Aayush, and my other family members who have always been there for me, providing me with their unwavering support and encouragement. Without their constant love and support, I could not have accomplished this. I would like to thank Riya, and all my friends for their unwavering support throughout my journey at IISERB. Thank you all for your contributions to my academic journey and the completion of my thesis.

Aditya Mishra

ABSTRACT

In academic settings, the demanding environment often forces students to prioritize academic performance over their physical well-being. Moreover, privacy concerns and the inherent risk of data breaches hinder the deployment of traditional machine learning techniques for addressing these health challenges. In this study, we introduce **RiM: Record, Improve, and Maintain**, a mobile application which incorporates a novel personalized machine learning framework that leverages federated learning to enhance students' physical well-being by analyzing their lifestyle habits.

Our approach involves pre-training a multilayer perceptron (MLP) model on a large-scale simulated dataset to generate personalized recommendations. Subsequently, we employ federated learning to fine-tune the model using data from IISER Bhopal students, thereby ensuring its applicability in real-world scenarios. The federated learning approach guarantees differential privacy by exclusively sharing model weights rather than raw data. Experimental results show that the FedAvg–based RiM model achieves an average accuracy of 60.71% and a mean absolute error of 0.91—outperforming the FedPer variant (average accuracy 46.34%, MAE 1.19)—thereby demonstrating its efficacy in predicting lifestyle deficits under privacy-preserving constraints.

Table of Contents

List of Symbols & Abbreviations

δ	Stride Length
τ	Step Threshold
τ_j	Deficit Threshold
θ	Risk Score Threshold
d	Deficit
R	Composite Risk Score
w	Weights
FedAvg	Federated Averaging
FedPer	Federated Personalization
FL	Federated Learning
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MTL	Multi-task Learning
RiM	Record, Improve and Maintain

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The intense competitive environment in academic settings has driven students to prioritize their studies and academic goals, often at the expense of their physical well-being. The rise of technological distractions, such as video games and binge-watching, has further exacerbated this issue. Furthermore, privacy concerns and the risk of data breaches in traditional machine learning approaches have limited the potential of AI to address these challenges for students.

To address these issues, we present RiM: Record, Improve, and Maintain, a personalized federated learning-based model designed to enhance students' physical well-being by analyzing their lifestyle habits. By leveraging federated learning, RiM ensures differential privacy, guaranteeing that we have no access to the data collected from students, thus addressing both privacy and wellness concerns effectively.

1.2 Literature Review

Recent meta-analyses and randomized trials underscore the efficacy of digital interventions in promoting physical well-being. [1] conducted a meta-analysis of 15 studies targeting university students and found that personalized SMS reminders and smartphone applications led to significant increase in daily step counts, though effects on moderate-to-vigorous physical activity and sedentary behavior varied across studies. [2] underscore the significance of mobile health applications in enhancing students' motivation, effectively promoting physical activity levels. [3] explored that mobile apps and fitness trackers are positively associated with increased physical activity during the pandemic. It highlighting the pivotal role digital tools play in supporting health-related behaviors. Furthermore, studies [4, 5] demonstrate the effectiveness of mobile phone–based physical activity programs in reducing symptoms of depression and perceived stress, as well as promoting healthier lifestyle choices among young individuals experiencing mental health conditions.

Multi-task learning (MTL) is a paradigm that trains a single model to perform multiple related tasks by sharing representations, improving generalization especially when per-task data are limited. The work by [6] depicts that MTL mitigates overfitting and leverages inter-task correlations. [7] developed an MTL-enhanced convolutional-RNN architecture that jointly predicts sleep stages and heart-rate variability from ECG and PPG data. Their model achieved the accuracy similar to single-task baselines while using 75% less input data and 7.5 times fewer parameters. In the clinical domain, [8] introduced a multimodal LSTM-based MTL model to predict both hospital length-of-stay (regression) and 30-day readmission (classification) using wrist-worn sensor data. Their joint model significantly outperformed separate single-task models on both objectives. Together, these works illustrate MTL's ability to handle concurrent goals under constrained data and compute budgets—an approach we adopt by fine-tuning an MLP to jointly predict sleep and distance deficits from shared lifestyle features.

The **federated learning** (FL) paradigm enables on-device training by exchanging only model updates, thereby keeping each user's data on their own device. [9] formalized this approach with the FedAvg algorithm, showing that iterative averaging of locally-computed updates can train deep networks across non-IID mobile data while reducing communication rounds by 10-100 times while preserving data locality. Building formal privacy guarantees into FL, [10] proposed DP-FedAvg. It applies per-client gradient clipping and Gaussian noise to each update, achieving user-level (ϵ, δ)-differential privacy with only minor accuracy degradation given a sufficiently large cohort. Furthermore, [11] developed a federated f-differential privacy framework that leverages Gaussian differential privacy and tight composition analyses to offer both record-level and group-level privacy guarantees under federated settings. This further tightens the privacy-utility trade-off. To support research and real-world deployment at scale, [12] released Flower. It is an open-source platform that provides high-level building blocks to prototype FL workflows quickly and works across many different devices and environments, from mobile phones to edge servers. Also, it scales seamlessly from single-node simulations to millions of clients on real devices.

1.3 Introduction to RiM

The decentralized approach of training models by distributing the data on different clients and learning a shared model by aggregating locally computed weights is known as **federated learning**. This concept was first introduced by [13]. It preserves the sensitive user information by sharing only the weights of the model and not the raw data, thus ensuring differential privacy.

The developed mobile application is designed to capture the user's daily step counts, distance traveled, sleep hours, and meal information. The pre-trained MLP model is then fine-tuned on the user's data and the updates are shared to the **Flower framework**[1]. Flower is an open-source framework designed to simplify and streamline federated learning tasks on a cluster of machines. This Python-based framework offers a user-friendly solution for training a wide range of models, including deep neural networks. The shared weights are combined via **Federated Averaging** [9] and **Federated Personalization** [14] to update the global model. Subsequently, the updated weights are sent to each client device.

After fine-tuning, the MLP model predicts sleep deficit and distance deficit from the ideal range by using a variety of features, such as: user's height, weight, age, gender, sleep hours, distance traveled, and meal information. A combination of rule-based and ML-based approaches enables the model to learn the inter-parameter relations and provides the user with how changes in one aspect of the lifestyle can affect the other parameters of physical well-being. Furthermore, priority-based mechanism is incorporated in order to provide only relevant and most important recommendation.

We describe our data pipeline in Chapter 2: first the generation of a large-scale synthetic dataset for pre-training, then the fine-tuning dataset collected via the RiM Android app and its processing. In Chapter 3, we present the RiM mobile application, detail the MLP architecture and rule-based recommendation system, explain our use of FedAvg and FedPer for privacy-preserving fine-tuning, outline the evaluation metrics and key implementation settings. Chapter 4 reports our experimental findings—comparing accuracy and MAE for FedAvg versus FedPer, analyzing class-imbalance and personalization effects, and illustrating trade-offs. It is followed by an in-depth discussion of the results. Finally, Chapter 5 summarizes our contributions, acknowledges the Android 13–only limitation, and proposes future work on broadening version support, on-device model integration, and advanced bi-level federated algorithms like Ditto.

[1] https://flower.ai

Chapter 2

Data

2.1 Pre-training on Simulated Data

The lack of availability of a real-world data set with the set of features taken into consideration in this study makes us use simulated data set for pre-training the MLP model. The data set is generated for each feature separately, as each feature can be approximated to follow different distribution in the real-world scenario.

Creating a realistic synthetic dataset for the study requires assigning appropriate statistical distributions to each feature based on empirical data. Table 2.1 illustrates the distribution each feature follow.

Features	**Distribution**	**References**
Step Count	Negative Binomial Distribution	[15]
Distance Traveled	Log-Normal Distribution	[16]
Sleep Hours	Normal Distribution	[17]
Meal Consumption	Bernoulli Distribution	[18]
Height	Normal Distribution	[19]
Weight	Log-Normal Distribution	[20]
Age	Truncated Normal (Empirical Distribution)	[21]
Gender	Bernoulli Distribution	[22]

Table 2.1: Probabilistic distribution followed by each feature in the simulated data.

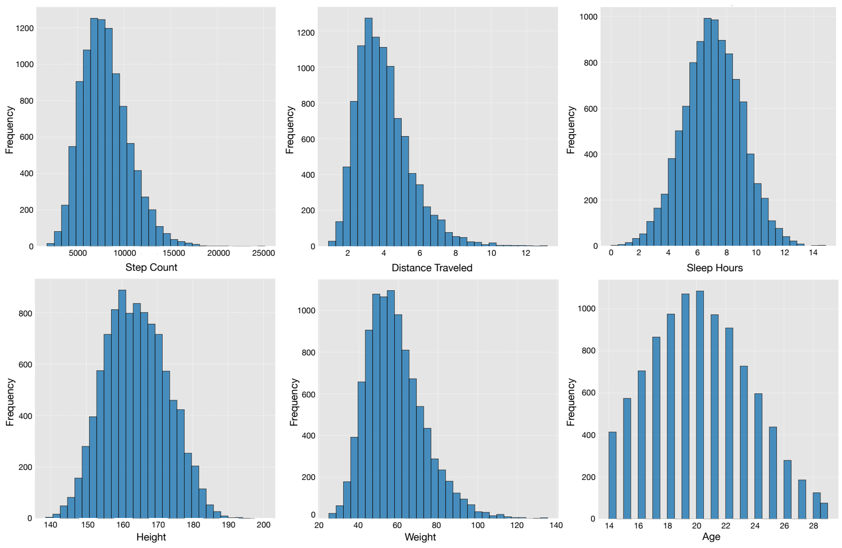

Figure 2.1: The figure illustrates the distributions of simulated features used for pre-training the MLP model. Simulated step counts follow a negative binomial distribution, while distance traveled and weight are drawn from log-normal distributions. In addition, meal consumption and gender are generated using Bernoulli distributions, and age is modeled based on an empirical distribution. The figure also illustrates the normal distributions for sleep hours and height.

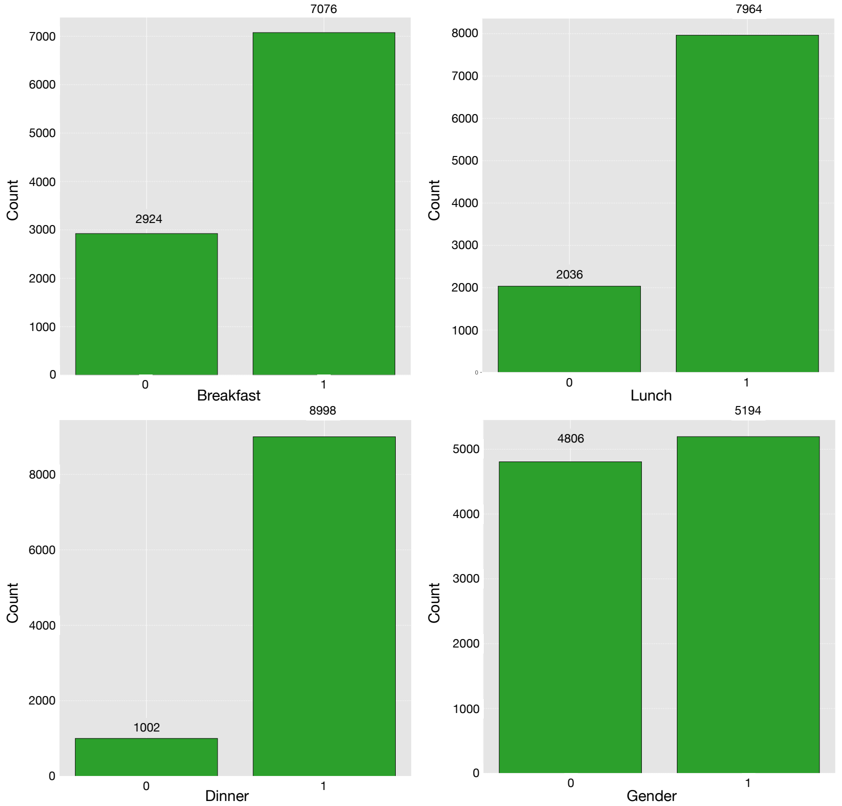

Figure 2.2: The figure presents the distribution of meal consumption—breakfast, lunch, and dinner—where a value of 1 indicates the meal was taken and 0 indicates it was skipped. It also shows the gender distribution within the simulated dataset. All variables follow a Bernoulli distribution.

2.2 Fine-tuning on Real-world Data

Through the developed mobile application, we collect the user's physical activity and demographic data locally. The model is fine-tuned on this data to adapt better to the real-world aspect. Table 2.2 presents an example of the collected data. We collected user data over a 15-day period. An MLP model was fine-tuned using the data from the first 8 days and subsequently used to generate recommendations for the following 7 days. FedPer algorithm is used for fine-tuning the MLP model.

Table 2.2: The table depicts the collected data for a period of 1 week of a female user. † represent that a feature is binary.

Date	Steps (#)	Distance (km)	Sleep (hrs)	Breakfast†	Lunch†	Dinner†	Age (yrs)	Height (cm)	Weight (kg)	Gender†
2025-04-18	9657	4.83	8.04	0	1	1	22	165	59	0
2025-04-19	8234	4.11	8.21	1	1	1	22	165	59	0
2025-04-20	7698	3.84	7.83	0	0	1	22	165	59	0
2025-04-21	11345	5.67	7.94	1	1	1	22	165	59	0
2025-04-22	12876	6.43	6.54	1	1	1	22	165	59	0
2025-04-23	6456	3.22	7.37	1	0	1	22	165	59	0
2025-04-24	9825	4.91	6.18	1	1	0	22	165	59	0

Eighteen volunteers (14 male and 4 female) initially enrolled in the study and downloaded the mobile application. However, attrition and data-quality issues reduced the evaluable data to ten participants. The primary factors contributing to this reduction in evaluable datasets are as follows:

- Volunteer withdrawals (n=4) occurred because of scheduling conflicts and competing personal or professional commitments.
- Technical failures (n=2) malfunctions of participants' accelerometer sensors produced incomplete and corrupted datasets that failed to meet our predefined quality criteria.
- Insufficient engagement (n=2) was observed in participants who failed to meet the minimum interaction threshold (entering meal information), rendering their data unusable for reliable analysis.

Consequently, only the ten remaining datasets—each meeting compliance, completeness, and quality criteria—were included in the final analysis.

Since step count and distance exhibit a near-perfect linear relationship (Pearson's $r \approx 0.98$) [15], we retain only the distance feature to mitigate multicollinearity and reduce dimensionality. Rather than passing height and weight separately, we compute the body mass index (BMI) to encapsulate overall physical fitness in a single metric—a practice shown to enhance physiological-model performance in large-scale cohort studies [23]. Breakfast consumption has been identified as the principal daily meal influencing metabolic health and cognitive performance, and thus is modeled as a distinct binary feature [24]. To capture overall dietary patterns without inflating model complexity, we aggregate lunch and dinner into a single "meal" feature representing the total number of meals per day, consistent with approaches in nutritional epidemiology [25].

Chapter 3

Method

3.1 RiM: Mobile Application

The onset of smartphones has revolutionized how we monitor our lifestyles and collect data effortlessly. In today's technology-driven world, our devices remain with us at all times, providing continuous insights into our daily habits. Thus, we develop an android application that tracks and stores users' lifestyle data locally, including physical activity, sleep duration, and dietary information. This real-world data plays a crucial role in fine-tuning our model to adapt to everyday scenarios.

We use **React Native**[1] to develop the application. It is an open-source UI software framework developed by Meta for creating mobile applications. The Android application exclusively utilizes accelerometer data to monitor and store parameters, ensuring computational efficiency and reduced battery consumption. Upon launch, the application requests the user's demographic details, such as height, weight, gender, and age. These details further help the MLP model tailor personalized recommendations.

We subscribe to the phone's accelerometer for calculating the user's daily **step count** and **distance traveled**. Algorithm 1 demonstrates how the step count is calculated. The algorithm continuously monitors accelerometer readings by calculating the magnitude of the acceleration vector from its x, y, and z components. It compares the change in magnitude from the previous reading against a predefined threshold (τ). When this difference exceeds the threshold, a step is recorded. To prevent noisy data and sensor fluctuations from triggering false step counts, a **debounce mechanism** is implemented. This mechanism ensures that a new step is only recorded if at least 300 milliseconds have passed since the last detected step. By imposing this time delay, the algorithm filters out rapid, minor changes that are unlikely to be actual steps, thereby enhancing the accuracy and reliability of the step detection process.

[1] https://reactnative.dev

Algorithm 1 Step Counter with Debounce

```
$lastMagnitude \leftarrow 0$
$lastStepTime \leftarrow 0$        ▷ Timestamp of the last detected step in milliseconds
for each accelerometer reading $(x, y, z)$ do
    $magnitude \leftarrow \sqrt{x^2 + y^2 + z^2}$
    $currentTime \leftarrow$ current timestamp in milliseconds
    if $|magnitude - lastMagnitude| > \tau$ and $(currentTime - lastStepTime) > 300$
        $lastStepTime \leftarrow currentTime$
        $stepCount \leftarrow stepCount + 1$
        $distance \leftarrow distance + \delta$
        Update the last activity time with the current time
    end if
    $lastMagnitude \leftarrow magnitude$
end for
```

We record the user's daily **sleep hours** by monitoring the inactivity in the night. As the algorithm 2 illustrates, the sleep tracking algorithm adjusts sleep hours by combining two mechanisms. First, within the designated sleep window (10 PM to 10 AM), it calculates the time elapsed since the last recorded activity. If the inactivity period exceeds 2 hours, the user is marked as sleeping and sleep hours are incrementally increased at a rate of one minute. Second, if the user is flagged as sleeping and is then detected moving (between midnight and 10 AM), the algorithm compensates for lost sleep by adding 2 hours to the sleep total and resets the sleeping state. This dual approach ensures that short interruptions in activity do not skew sleep measurements, while still accurately capturing extended periods of inactivity as sleep.

Algorithm 2 Sleep Hours Tracking

```
$now \leftarrow$ current time
if $now$.hour $\geq 22$ or $now$.hour $< 10$        ▷ Check for sleep window
    $timeSinceLastActivity \leftarrow \frac{now - lastActivityTime}{1000 \times 60}$        ▷ Convert to minutes
    if $timeSinceLastActivity > 120$        ▷ Inactivity for more than 2 hours
        if $\neg isSleeping$
            $isSleeping \leftarrow$ true
        end if
        $sleepHours \leftarrow sleepHours + \frac{1}{60}$        ▷ Increment sleep hours by 1 minute
    else
        $currentTime \leftarrow$ current time
        if $isSleeping$ and $0 \leq currentTime$.hour $< 10$
            $sleepHours \leftarrow sleepHours + 2$        ▷ Add 2 hours to compensate lost sleep
            $isSleeping \leftarrow$ false
        end if
    end if
end if
```

To record the user's **meal information**, we create a drop-down input box with yes/no options that asks whether the user has taken their meal. To reduce user burden, a notification is scheduled at 21:45 hours to remind the user to enter their meal information once a day. This simple interface ensures that meal data is consistently captured without causing fatigue, while all other parameters are automatically tracked in the background, requiring no user input. For more details on the development of the Android application please visit GitHub[2].

3.2 Machine Learning Frameworks

The choice of a multi-layer perceptron (MLP) model is motivated by its ability to be fine-tuned on a relatively small dataset (7 days in our case), its low computational overhead, and ease of integration with Android applications. The proposed MLP architecture consists of five hidden layers. The first hidden layer contains 32 neurons and the number of neurons is halved in each subsequent layer, resulting in 4 neurons in the final hidden layer. Figure 3.1 illustrates the architecture of the proposed method.

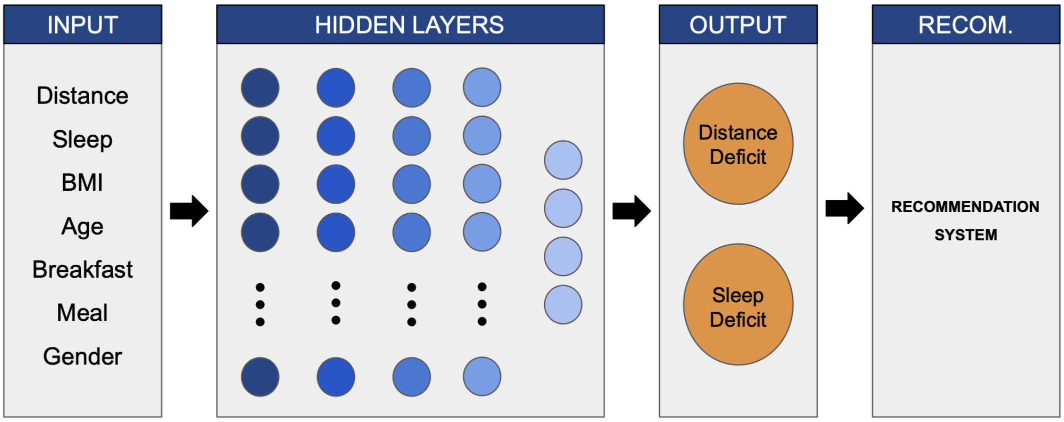

Figure 3.1: Architecture of the proposed MLP model, which consists of five hidden layers and outputs sleep and distance deficits that feed into the recommendation system to generate personalized recommendations.

We employ a two-stage approach that (i) predicts lifestyle deficits via a pretrained MLP regression model and (ii) applies rule-based severity and interaction checks to compute per-parameter risk scores. Only the highest-risk and most relevant recommendations are presented to the user.

Given a user feature vector

$$\mathbf{x} = [\text{distance}, \text{sleep}, \text{bmi}, \text{age}, \text{breakfast}, \text{meal}, \text{gender}],$$

we first standardize:

$$\tilde{\mathbf{x}} = \text{StandardScaler}(\mathbf{x}).$$

[2] https://github.com/adityamishraaaa/RiM.git

The MLP model f then predicts deficits

$$\mathbf{d} = f(\tilde{\mathbf{x}}) = \Big[d_{\text{sleep}},\, d_{\text{distance}}\Big],$$

where each deficit d_j is defined as

$$d_j = \begin{cases} (\text{ideal}_{j,\min} - x_j), & x_j < \text{ideal}_{j,\min}, \\ -(x_j - \text{ideal}_{j,\max}), & x_j > \text{ideal}_{j,\max}, \\ 0, & \text{otherwise.} \end{cases}$$

For each primary parameter $j \in \{\text{sleep, distance}\}$, we only generate a recommendation if

$$|d_j| > \tau_j,$$

where τ_j is a deficit threshold. In that case, we assign

$$r_j = w_j\,|d_j|, \tag{3.1}$$

with weights w_{sleep} and w_{distance}. A corresponding message m_j (e.g. "Try to get more sleep") is paired with each r_j.

We also include rule-based risks for meal skipping, abnormal BMI and inter-parameter interactions. Each rule contributes an additional risk score by the same formula (3.1), with its own weight. Finally, we compute the composite risk score summing over all parameters and interactions:

$$R = \sum_j r_j, \tag{3.2}$$

If $R > R_{\text{high}}$, we prepend a high priority tag. To ensure conciseness, we filter out any $r_j < \theta$. Sort the remaining risk–message pairs $\{(r_j, m_j)\}$ in descending order of r_j and finally select the top N messages. If no r_j survives filtering, a generic **"Your lifestyle parameters are close to ideal. Keep it up!"** message is shown instead.

This design guarantees that only parameters with meaningful model-predicted deficits or significant rule violations yield recommendations, that compound risks are captured via interaction rules, and that the user sees only the most pressing, relevant advice. The MLP model is pre-trained on simulated data using separate training and validation sets, along with an early stopping mechanism to prevent overfitting. Figure 3.2 depicts the training curves. Pre-training allows the model to learn general patterns and representations. Furthermore, it helps to mitigate the limitations posed by the scarcity of real-world data.

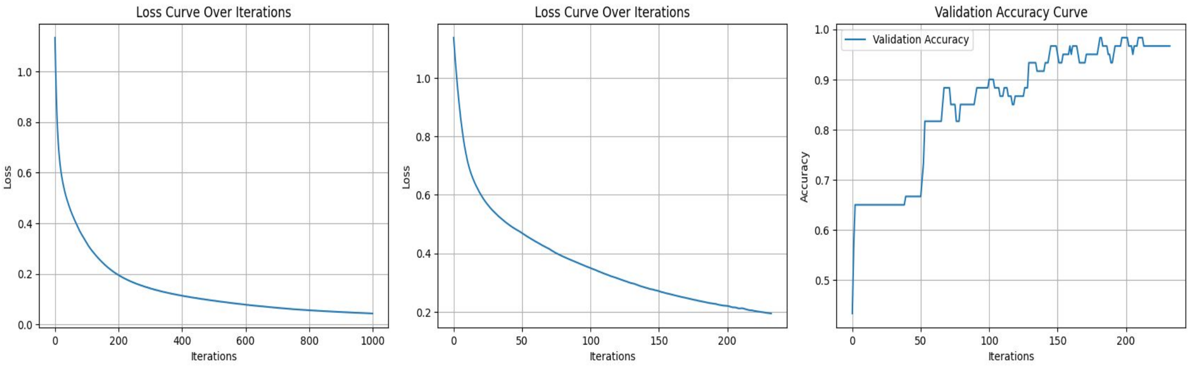

Figure 3.2: Training loss and validation accuracy curves plotted across pre-training iterations of the MLP model.

3.3 Federated Learning

We use the Flower framework to implement federated learning. A central server is set up to load the pre-trained model and distribute its weights to individual clients, where each client represents a single user. Each client then fine-tunes the pre-trained model using its own real-world data. The data remains on the client side, ensuring that users' data stays isolated and does not interfere with other clients.

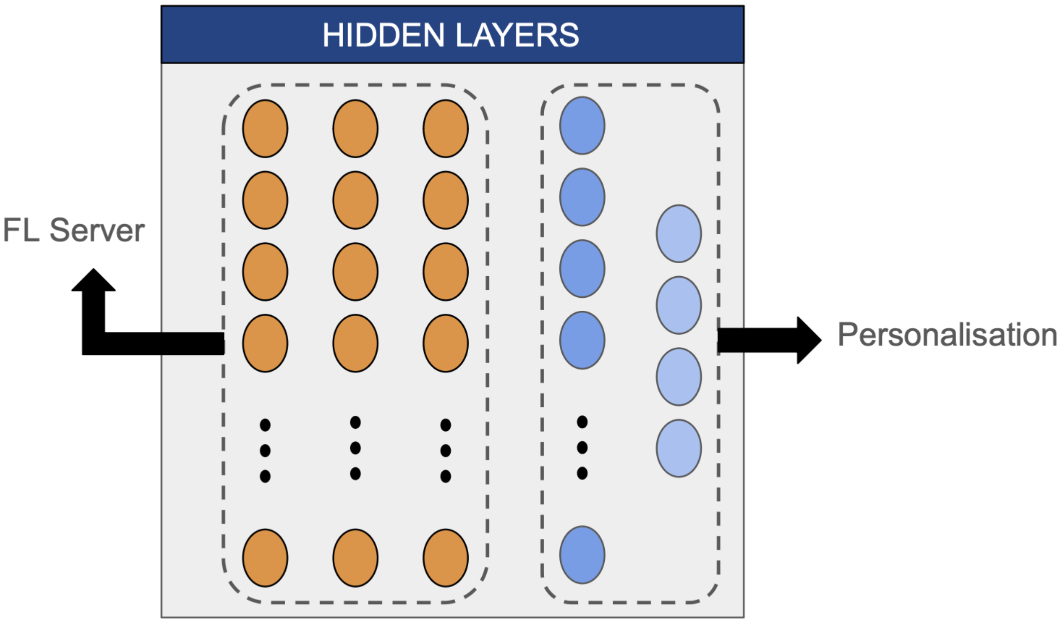

Figure 3.3: The figure illustrates the FedPer weight-update process: orange neurons denote the shared root layers that are sent to the server averaged via FedAvg, while blue neurons denote the personal head layers that remain on the client device to preserve personalization.

We employ both FedAvg and FedPer algorithms to fine-tune the model. The FedAvg algorithm builds a global model by having each client train a local copy on its own data, then sending only the updated weights (not the data) back to a central server. The server aggregates these local updates by computing a weighted average to form a new global model. This global model is then redistributed to clients for the next round, repeating until convergence. FedPer splits the MLP model into a **shared root** (the first three layers) and **personal head** (the remaining two layers), so that clients collaboratively train only the root while keeping their heads local. In each round, every client fine-tunes the full model on its own data, then sends only the updated root weights to the server; the server averages these root updates (as in FedAvg) and broadcasts the new shared root back. This way, clients benefit from global feature learning but retain personalized output layers that capture local data heterogeneity. The FedPer fine-tuning process is illustrated by the Figure 3.3. The orange neurons denote the shared root layers, while the blue neurons denote the personalized head layers.

3.4 Evaluation Metrics

We evaluated the performance of the models with two evaluation metrics – accuracy and mean absolute error (MAE).

Accuracy: The accuracy metric here measures the fraction of samples for which the predicted and true deficits share the same sign (positive, zero, or negative). In other words, it evaluates how often our model correctly predicts whether each deficit is above, equal to, or below zero.
Mathematically, if we have n samples, true targets y and predictions $\hat{y}$, then:

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\Big(\text{sign}(y_i) = \text{sign}(\hat{y}_i)\Big) \tag{3.3}$$

Where, sign(x) is the signum function returning +1 if $x > 0$, 0 if $x = 0$ and -1 if $x < 0$. $\mathbf{1}(.)$ is the indicator function, equal to 1 if its argument is true, and 0 otherwise.

MAE: It calculates the average absolute difference between predicted values ($\hat{y}$) and actual values (y). Mathematically,

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{3.4}$$

Where n refers to the sample size. MAE is the loss function that need to be minimized in the process of training a machine learning model. Lower MAE values suggest better model performance.

3.5 Implementation Details

We use simulated data to pre-train the model and fine-tune and test in on the collected real-world dara. We employ T4 GPUs with a batch size of 32. The models are trained using the Adam Optimizer with a learning rate of 1×10^{-3} over 200 epochs. To obtain final results, we average the scores of all the clients. Table 3.1 contains the values of all other hyperparameter used in the study.

Table 3.1: List of all the hyperparameter used in the study. The list encompasses values used in MLP model, federated learning and mobile application development (in order).

Hyperparameter	**Value**
Batch Size	32
Learning Rate	1×10^{-3}
Pre-training Epochs	200
Number of Units in Hidden Layer	[64, 32, 16, 8, 4]
Ideal Sleeping Range	[7 hours, 9 hours]
Ideal Distance Traveled	[5 km, 8 km]
w_{sleep}	1
$w_{distance}$	0.8
w_{bmi}	1
w_{meal}	1
R_{high}	3
θ	0.5
Number of Recommendations (*top_n*)	2
Fine-tuning Epochs	10
Number of Clients	10
τ	1.8
δ	0.5m
Sleep Window	[10 PM, 10 AM]

Chapter 4

Results and Discussion

The results for FedAvg fine-tuning approach and FedPer approach are presented by Table 4.1 and Table 4.2, respectively. It is evident that FedAvg outperforms the FedPer approach. FedAvg achieves an accuracy of 60.71% and MAE of 0.91 (↓ 0.28 compared to FedPer). It achieves a 14.37% more accurate predictions compared to FedPer algorithm.

Table 4.1: Client-wise accuracy (%) and mean absolute error for the FedAvg approach.

Client ID	**Accuracy (%)**	**Mean Absolute Error**
Client 0	81.42	0.82
Client 1	67.14	0.79
Client 2	34.63	0.81
Client 3	65.67	0.62
Client 4	25.18	0.90
Client 5	74.28	0.87
Client 6	74.28	0.41
Client 7	38.57	1.91
Client 8	69.28	1.32
Client 9	76.67	0.69
Average	**60.71**	**0.91**

FedAvg exhibits lower accuracy on Clients 2, 4, and 7. These clients' data are dom-

Table 4.2: Client-wise accuracy (%) and mean absolute error for the FedPer approach.

Client ID	**Accuracy (%)**	**Mean Absolute Error**
Client 0	63.57	1.26
Client 1	47.32	2.48
Client 2	31.33	0.12
Client 3	60.72	1.52
Client 4	25.71	0.29
Client 5	25.91	1.46
Client 6	60.83	0.97
Client 7	46.67	1.14
Client 8	47.14	1.24
Client 9	54.16	1.47
Average	**46.34**	**1.19**

inated by ideal case (most of their true deficits are zero) - so the model, when averaged across all participants, struggles to reproduce this all-zero pattern. In effect, the global update dilutes each client's local bias toward the ideal range, and the resulting model under-predicts the absence of a deficit. Conversely, FedAvg is better at identifying non-ideal behavior: for example, Client 0—whose data include a larger proportion of true deficits—achieves the highest accuracy (81.42%) under the same fine-tuning regime. This discrepancy highlights two key issues:

1. When a client's labels are majorly zero, the global model—trained on a more heterogeneous mix of deficits fails to specialize on the no deficit case.

2. FedAvg merges all client updates indiscriminately, which benefits common patterns (e.g. predicting deficits) but washes out client-specific biases toward ideal behavior.

Together, these factors suggest that purely averaging model weights may not adequately capture per-client idiosyncrasies, especially for users whose lifestyle metrics remain within recommended ranges.

FedPer yields a slight improvement in accuracy for clients whose data are dominated by ideal zero-deficit observations, but it conversely underperforms on predicting non-ideal behaviors. By isolating the last two layers as client-specific heads, FedPer allows each participant to specialize on their all-zero patterns—hence the slight gain for ideal cases—yet this same decoupling limits the transfer of deficit-prediction expertise for rarer non-ideal events.

This behavior can be explained by:

1. Personalized heads faithfully reproduce the zero-deficit bias present in many clients, driving up ideal-case accuracy.
2. Since deficit-related weights in the personal head are never shared, clients with mixed or skewed distributions fail to benefit from peers' learning on non-ideal patterns, resulting in lower accuracy on true deficits.

These findings underline the trade-off inherent in personalization: preserving local idiosyncrasies can boost performance on dominant patterns while potentially impairing the modeling of less frequent but clinically important non-ideal behaviors.

Chapter 5

Conclusion

In this work, we presented RiM (Record, Improve, and Maintain), a privacy-preserving mobile application for promoting student physical well–being by combining federated learning with a lightweight MLP recommendation engine. We first pre-trained our MLP on a large, simulated dataset to learn general patterns of sleep and walking-distance deficits, then fine-tuned it on real-world data collected from IISER Bhopal students using both FedAvg and FedPer strategies. Our experimental results demonstrated that FedAvg achieves higher accuracy on non-ideal behaviors (average accuracy 60.7% and MAE 0.91), whereas FedPer better captures client-specific ideal zero-deficit patterns (average accuracy 46.3%, MAE 1.19). To deliver actionable guidance, we devised a hybrid recommendation layer that combines these learned deficits with rule-based severity and interaction checks, computing composite risk scores to prioritize the top two most urgent, personalized recommendations.

This approach ensures differential privacy by never transmitting raw user data; model personalization via per-client heads (FedPer) or global aggregation (FedAvg) as appropriate; and actionable insights through a priority-based, rule-enhanced recommender that surfaces only the most critical lifestyle adjustments. However, one **limitation** of the current RiM implementation is that the Android application supports only Android version 13 or lower, which restricts compatibility with newer devices.

Future Work: To broaden usability and enhance on-device intelligence, we will first update the mobile application to support all modern Android API levels. Next, we plan to integrate the pretrained MLP model directly within the app—enabling real-time, offline inference without server communication. Finally, we will investigate advanced bi-level optimization algorithms such as Ditto for federated learning, which balance global aggregation with per-client personalization more effectively, further improving both fairness and predictive performance.

Bibliography

[1] Siyuan Bi, Junfeng Yuan, Yanling Wang, Wenxin Zhang, Luqin Zhang, Yongjuan Zhang, Rui Zhu, and Lin Luo. Effectiveness of digital health interventions in promoting physical activity among college students: Systematic review and meta-analysis. *Journal of medical Internet research*, 26:e51714, 2024.

[2] Caroline A Figueroa, Laura Gomez-Pathak, Imran Khan, Joseph Jay Williams, Courtney R Lyles, and Adrian Aguilera. Ratings and experiences in using a mobile application to increase physical activity among university students: implications for future design. *Universal Access in the Information Society*, 23(2):821–830, 2024.

[3] Huong Ly Tong, Carol Maher, Kate Parker, Tien Dung Pham, Ana Luisa Neves, Benjamin Riordan, Clara K Chow, Liliana Laranjo, and Juan C Quiroz. The use of mobile apps and fitness trackers to promote healthy behaviors during covid-19: A cross-sectional survey. *PLOS Digital Health*, 1(8):e0000087, 2022.

[4] Hyungsook Kim, Kikwang Lee, Ye Hoon Lee, Yoonjung Park, Yonghyun Park, Yeonwoo Yu, Jaeyoung Park, and Sihyeon Noh. The effectiveness of a mobile phone–based physical activity program for treating depression, stress, psychological well-being, and quality of life among adults: quantitative study. *JMIR mHealth and uHealth*, 11:e46286, 2023.

[5] Joseph Firth, Chelsea Sawyer, John Sainsbury, Rachel Morell, Hamish Fibbins, Sandra Bucci, Lamiece Hassan, Josh A Firth, Henry Onyweaka, John Torous, et al. Using physical health apps to promote healthy lifestyles in youth mental healthcare: A nationwide perspective-gathering exercise of over 400 service users. *Psychiatry Research*, 342:116187, 2024.

[6] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

[7] Hao-Yi Chih, Tanveer Ahmed, Amy P Chiu, Yu-Ting Liu, Hsin-Fu Kuo, Albert C Yang, and Der-Hsien Lien. Multitask learning for automated sleep staging and wearable technology integration. *Advanced Intelligent Systems*, 6(1):2300270, 2024.

[8] Sajid Ali, Shaker El-Sappagh, Farman Ali, Muhammad Imran, and Tamer Abuhmed. Multitask deep learning for cost-effective prediction of patient's length of stay and readmission state using multimodal physical activity sensory data. *IEEE Journal of Biomedical and Health Informatics*, 26(12):5793–5804, 2022.

[9] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[10] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[11] Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *International conference on artificial intelligence and statistics*, pages 2251–2259. PMLR, 2021.

[12] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[13] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[14] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[15] Catrine Tudor-Locke and David R Bassett. How many steps/day are enough? preliminary pedometer indices for public health. *Sports medicine*, 34:1–8, 2004.

[16] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[17] Diane S Lauderdale, Kristen L Knutson, Lijing L Yan, Kiang Liu, and Paul J Rathouz. Self-reported and measured sleep duration: how similar are they? *Epidemiology*, 19(6):838–845, 2008.

[18] JOHANNA T DWYER, MARGUERITE EVANS, ELAINE J STONE, HENRY A FELDMAN, LESLIE LYTLE, DEANNA HOELSCHER, CAROLYN JOHNSON, MICHELLE ZIVE, and MINHUA YANG. Adolescents' eating patterns influence their nutrient intakes. *Journal of the American Dietetic Association*, 101(7):798–802, 2001.

[19] Karri Silventoinen. Determinants of variation in adult body height. *Journal of biosocial science*, 35(2):263–285, 2003.

[20] Katherine M Flegal, Margaret D Carroll, Brian K Kit, and Cynthia L Ogden. Prevalence of obesity and trends in the distribution of body mass index among us adults, 1999-2010. *Jama*, 307(5):491–497, 2012.

[21] United Nations, Department of Economic and Social Affairs, Population Division. World population prospects 2019: Highlights. https://population.un.org/wpp/Download/Standard/, 2019. Accessed: 2025-04-03.

[22] World Health Organization. Global health observatory data repository: Sex ratio at birth. https://www.who.int/data/gho/data/indicators/indicator-details/GHO/sex-ratio-at-birth, 2018. Accessed: 2025-04-03.

[23] World Health Organization. Obesity: preventing and managing the global epidemic. Technical Report 894, World Health Organization, Geneva, Switzerland, 2000.

[24] Gail C Rampersaud, Mark A Pereira, Beverly L Girard, Judi Adams, and Jordan D Metzl. Breakfast habits, nutritional status, body weight, and academic performance in children and adolescents. *Journal of the american dietetic association*, 105(5):743–760, 2005.

[25] A. K. Kant and B. I. Graubard. A little change goes a long way: the impact of meal definition on nutrient intake estimates. *Journal of Nutrition*, 140(2):353–358, 2010.