A Proof of Proposition 1

For convenience, we restate the proposition.

Proposition 4. *For $m \in \mathbb{N}$ clients with local datasets $D^1, \dots, D^m$ and unlabeled dataset U drawn iid. from $\mathcal{D}$, let $\mathcal{A}$ be a learning algorithm that achieves a linearly increasing training accuracy a_t for all labelings of U, i.e., there exists $c \in \mathbb{R}_+$ such that $a_t = 1 - c/t$, then there exists $t_0 \in \mathbb{N}$ such that $a_t \geq 1/2$) and* AIMHI *with majority vote converges with probability $1 - \delta$, where*

$$\delta \leq |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}, t_0 + 1\right)$$

and $\zeta(x, q)$ is the Hurwitz zeta function.

Proof. Let P_t denote the consensus label at time $t \in \mathbb{N}$. We first show that the probability δ_t of $P_t \neq P_{t-1}$ is bounded. Since the learning algorithm $\mathcal{A}$ at time $t \geq t_0$ achieves a training accuracy $a_t \geq 0.5$, the probability can be determined via the CDF of the binomial distribution, i.e.,

$$\begin{aligned}\delta_t =& \mathbb{P}\left[\exists u \in U : \sum_{i=1}^{m} \mathbb{1}_{h_t^i(u)=v} < \left\lfloor \frac{m}{2} \right\rfloor\right] \\ =& F\left(\left\lfloor \frac{m}{2} \right\rfloor - 1, m, a_t\right) = \sum_{i=1}^{\left\lfloor \frac{m}{2} \right\rfloor - 1} \binom{m}{i} a_t^i (1 - a_t)^{m-i} \quad .\end{aligned}$$

Applying the Chernoff bound and denoting by $D(\cdot\|\cdot)$ the Kullback-Leibler divergence yields

$$\begin{aligned}\delta_t \leq& \exp\left(-mD\left(\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m} \middle\| a_t\right)^2\right) \\ =& \exp\left(-m\left(\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m} \log \frac{\frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}}{a_t} + \left(1 - \frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}\right) \log \frac{1 - \frac{\left\lfloor \frac{m}{2} \right\rfloor - 1}{m}}{1 - a_t}\right)\right) \\ \leq& \exp\left(-m\left(\frac{\frac{m}{2}}{m} \log \frac{\frac{\frac{m}{2}}{m}}{a_t} + \left(1 - \frac{\frac{m}{2}}{m}\right) \log \frac{1 - \frac{\frac{m}{2}}{m}}{1 - a_t}\right)\right) \\ =& \exp\left(-m\left(\frac{1}{2} \log \frac{\frac{1}{2}}{a_t} + \frac{1}{2} \log \frac{\frac{1}{2}}{1 - a_t}\right)\right) = \exp\left(-\frac{m}{2} \log \frac{1}{2a_t} - \frac{m}{2} \log \frac{1}{2(1 - a_t)}\right) \\ =& \exp\left(\frac{m}{2}\left(\log 2a_t + \log 2(1 - a_t)\right)\right) = (2a_t)^{\frac{m}{2}} \left(2(1 - a_t)\right)^{\frac{m}{2}} = 4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}} \quad .\end{aligned}$$

The union bound over all $u \in U$ yields

$$\delta_t \leq |U| 4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}} \quad .$$

To show convergence, we need to show that for $t_0 \in \mathbb{N}$ it holds that

$$\sum_{t=t_0}^{\infty} \delta_t \leq \delta$$

for $0 \leq \delta < 1$. Since we assume that a_t grows linearly, we can write wlog. $a_t = 1 - c/t$ for some $c \in \mathbb{R}_+$ and $t \geq 2c$. With this, the sum can be written as

$$\begin{aligned}\sum_{t=t_0}^{\infty} \delta_t \leq& |U| \sum_{t=t_0}^{\infty} 4^{\frac{m}{2}} \left(1 - \frac{c}{t}\right)^{\frac{m}{2}} \left(\frac{c}{t}\right)^{\frac{m}{2}} = |U| 4^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{\frac{t}{c} - 1}{\frac{t^2}{c^2}}\right)^{\frac{m}{2}} \\ \leq& |U| 4^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{\frac{t}{c}}{\frac{t^2}{c^2}}\right)^{\frac{m}{2}} = (4c)^{\frac{m}{2}} \sum_{t=t_0}^{\infty} \left(\frac{1}{t}\right)^{\frac{m}{2}} = |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}\right) - H_{t_0}^{\left(\frac{m}{2}\right)} \quad , \quad ,\end{aligned}$$

where $\zeta(x)$ is the Riemann zeta function and $H_n^{(x)}$ is the generalized harmonic number. Note that $H_n^{(x)} = \zeta(x) - \zeta(x, n+1)$, where $\zeta(x, q)$ is the Hurwitz zeta function, so that this expression can be simplified to

$$\sum_{t=t_0}^{\infty} \delta_t \leq |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}\right) - \zeta\left(\frac{m}{2}\right) + \zeta\left(\frac{m}{2}, t_0 + 1\right) = |U|(4c)^{\frac{m}{2}} \zeta\left(\frac{m}{2}, t_0 + 1\right) \quad .$$

□

B Details on Experiments

B.1 Details on Privacy Vulnerability Experiments

We measure privacy vulnerability by performing membership inference attacks against AIMHI and FEDAVG In both attacks, the attacker creates an attack model using a model it constructs from its training and test datasets. Similar to previous work [20], we assume that the training data of the attacker has a similar distribution to the training data of the client. Once the attacker has its attack model, it uses this model for membership inference. In blackbox attacks (in which the attacker does not have access to intermediate model parameters), it only uses the classification scores it receives from the target model (i.e., client's model) for membership inference. On the other hand, in whitebox attacks (in which the attacker can observe the intermediate model parameters), it can use additional information in its attack model. Since the proposed AIMHI does not reveal intermediate model parameters to any party, it is only subject to blackbox attacks. Vanilla federated learning on the other hand is subject to whitebox attacks. Each inference attack produces a membership score of a queried data point, indicating the likelihood of the data point being a member of the training set. We measure the success of membership inference as ROC AUC of these scores. The **vulnerability (VUL)** of a method is the ROC AUC of membership attacks over K runs over the entire training set (also called attack epochs) according to the attack model and scenario. A vulnerability of 1.0 means that membership can be inferred with certainty, whereas 0.5 means that deciding on membership is a random guess.

We assume the following attack model: clients are honest and the server may be semi-honest (follow the protocol execution correctly, but it may try to infer sensitive information about the clients). The main goal of a semi-honest server is to infer sensitive information about the local training data of the clients. This is a stronger attacker assumption compared to a semi-honest client since the server receives the most amount of information from the clients during the protocol, and a potential semi-honest client can only obtain indirect information about the other clients. We also assume that parties do not collude.

The attack scenario for AIMHI and DD is that the attacker can send a (forged) unlabeled dataset to the clients and observe their predictions, equivalent to one attack epoch ($K = 1$); the one for FEDAVG and DP-FEDAVG is that the attacker receives model parameters and can run an arbitrary number of attacks—we use $K = 500$ attack epochs.

B.2 Datasets

We use 3 standard image classification datasets: CIFAR10 [12], FashionMNIST [27], and SVHN [17]. We describe the datasets and our preprocessing briefly.

CIFAR10 consists of $50\,000$ training and $10\,000$ test 32×32 color images in 10 classes with equal distribution (i.e., a total of $6\,000$ images per class). Images are normalized to zero mean and unit

Dataset	training size	testing size	unlabeled size $\lvert U \rvert$	communication period b	number of rounds T
CIFAR10	$40 \cdot 10^3$	$10 \cdot 10^3$	$10 \cdot 10^3$	10	$3 \cdot 10^3$
FashionMNIST	$10 \cdot 10^3$	$10 \cdot 10^3$	$50 \cdot 10^3$	50	$20 \cdot 10^3$
Pneumonia	4386	624	900	20	$20 \cdot 10^3$
MRI	30	53	170	6	$2 \cdot 10^3$
SVHN	38 257	26 032	$35 \cdot 10^3$	10	$20 \cdot 10^3$

Table 3: Experimental setup for Sections 5.3, 5.4, and 5.3,

variance. *FashionMNIST* consists of 60 000 training and 10 000 test 28×28 grayscale images of clothing items in 10 classes with equal distribution. Images are not normalized. *SVHN* (Street View House Numbers) consists of 630 420 32×32 color images of digits from house numbers in Google Street View, i.e., 10 classes. The datasest is partitioned into 73 257 for training, 26 032 for testing, and 531 131 additional training images. In our experiments, we use only the training and testing set. Images are not normalized.

We use two standard datasets from the UCI Machine Learning repository for our experiments on collaboratively training interpretable models: WineQuality [3] and BreastCancer [23]. A short description of both datasets follows. *WineQuality* is a tabular dataset of 6 497 instances of wine with 11 features describing the wine (e.g., alcohol content, acidity, pH, and sulfur dioxide levels) and the label is a wine quality score from 0 to 10. We remove duplicate rows and transform the categorial type attribute to a numerical value. We then normalize all features to zero mean and unit variance. *BreastCancer* is a medical diagnostics tabular dataset with 569 instances of breast cell samples with 30 features describing cell nuclei with 2 classes (malignant and benign). We followed the same preprocessing steps as WineQuality dataset.

Furthermore, we use 2 medical image classification datasets, Pneumonia [11], and MRI[3]. *Pneumonia* consists of 5 286 training and 624 test chest x-rays with labels *normal*, *viral pneumonia*, and *bacterial pneumonia*. We simplify the labels to *healthy* and *pneumonia* with a class imbalance of roughly 3 pneumonia to 1 healthy. The original images in the Pneumonia dataset do not have a fixed resolution as they are sourced from various clinical settings and different acquisition devices. We resize all images to a resolution of 224×224 pixels without normalization. *MRI* consists of 253 MRI brain scans with a class imbalance of approximately 1.5 brain tumor scans to 1 healthy scan. Out of the total 253 images, we use 53 images as testing set. Similar to the pneumonia dataset, the original images have no fixed resolution and are thus resized to 150×150 without normalization.

B.3 Experimental Setup

Layer	Output Shape	Activation	Parameters
Conv2D	(32, 32, 32)	ReLU	896
BatchNormalization	(32, 32, 32)	-	128
Conv2D	(32, 32, 32)	ReLU	9248
BatchNormalization	(32, 32, 32)	-	128
MaxPooling2D	(16, 16, 32)	-	-
Dropout	(16, 16, 32)	-	-
Conv2D	(16, 16, 64)	ReLU	18496
BatchNormalization	(16, 16, 64)	-	256
Conv2D	(16, 16, 64)	ReLU	36928
BatchNormalization	(16, 16, 64)	-	256
MaxPooling2D	(8, 8, 64)	-	-
Dropout	(8, 8, 64)	-	-
Conv2D	(8, 8, 128)	ReLU	73856
BatchNormalization	(8, 8, 128)	-	512
Conv2D	(8, 8, 128)	ReLU	147584
BatchNormalization	(8, 8, 128)	-	512
MaxPooling2D	(4, 4, 128)	-	-
Dropout	(4, 4, 128)	-	-
Flatten	(2048,)	-	-
Dense	(128,)	ReLU	262272
BatchNormalization	(128,)	-	512
Dropout	(128,)	-	-
Dense	(10,)	Linear	1290

Table 4: CIFAR10 architecture

We now describe the details of the experimental setup used in our empirical evaluation.

[3] https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

In Section 5.3, we use $m = 5$ clients for all datasets. We report the split into training, test, and unlabeled dataset per dataset, as well as the used communication period b and number of rounds T in Table 3.

For all experiments, we use Adam as an optimization algorithm with a learning rate 0.01 for CIFAR10, and 0.001 for the remaining datasets. A description of the DNN architecture for each dataset follows. In section 5.4 we use the same setup for section 5.3 but we sample the local dataset from a Dirichlet distribution as described in 5.4. In section 5.5, We use the same setup for section 5.3 but we use $m \in \{5, 10, 20, 40, 80\}$ clients as described in 5.5.

The neural network architectures used for each dataset are given in the following. For CIFAR10 we use a CNN with multiple convolutional layers with batch normalization and max pooling. The details of the architecture are described in Table 4. For FashionMNIST, we use a simple feed forward architecture on the flattened input. The details of the architecture are described in Table 5. For

Layer	Output Shape	Activation	Parameters
Flatten	(784,)	-	-
Linear	(784, 512)	-	401,920
ReLU	(512,)	ReLU	-
Linear	(512, 512)	-	262,656
ReLU	(512,)	ReLU	-
Linear	(512, 10)	-	5,130

Table 5: FashionMNIST architecture

Pneumonia, we use a simple CNN, again with batch normalization and max pooling, with details given in Table 6. For MRI we use an architecture similar to pneumonia with details described in

Layer	Output Shape	Activation	Parameters
Conv2d	(3, 32, 32)	-	896
BatchNorm2d	(32, 32, 32)	-	64
Conv2d	(32, 32, 32)	-	18,464
BatchNorm2d	(64, 32, 32)	-	128
MaxPool2d	(64, 16, 16)	-	-
Conv2d	(64, 16, 16)	-	36,928
BatchNorm2d	(64, 16, 16)	-	128
MaxPool2d	(64, 8, 8)	-	-
Flatten	(4096,)	-	-
Linear	(2,)	-	4,194,306

Table 6: Pneumonia architecture

Table 7. For SVHN, we use again a standard CNN with batch normalization and max pooling,

Layer	Output Shape	Activation	Parameters
Conv2d	(3, 32, 32)	-	896
BatchNorm2d	(32, 32, 32)	-	64
Conv2d	(32, 32, 32)	-	18,464
BatchNorm2d	(64, 32, 32)	-	128
MaxPool2d	(64, 16, 16)	-	-
Conv2d	(64, 16, 16)	-	36,928
BatchNorm2d	(64, 16, 16)	-	128
MaxPool2d	(64, 8, 8)	-	-
Flatten	(32768,)	-	-
Linear	(2,)	-	2,636,034

Table 7: MRI architecture

detailed in Table 8.

In section 5.6, we use $m = 5$ clients. For decision trees (DT), we split by the Gini index with at least 2 samples for splitting. For RuleFit, we use a tree size of 4 and a maximum number of rules of 200.

Layer	Output Shape	Parameters
Conv2d	(3, 32, 32)	896
BatchNorm2d	(32, 32, 32)	64
Conv2d	(32, 32, 32)	9,248
MaxPool2d	(32, 16, 16)	-
Dropout2d	(32, 16, 16)	-
Conv2d	(32, 16, 16)	18,464
BatchNorm2d	(64, 16, 16)	128
Conv2d	(64, 16, 16)	36,928
MaxPool2d	(64, 8, 8)	-
Dropout2d	(64, 8, 8)	-
Conv2d	(64, 8, 8)	73,856
BatchNorm2d	(128, 8, 8)	256
Conv2d	(128, 8, 8)	147,584
MaxPool2d	(128, 4, 4)	-
Dropout2d	(128, 4, 4)	-
Flatten	(2048,)	-
Linear	(128,)	262,272
Dropout	(128,)	-
Linear	(10,)	1,290

Table 8: SVHN architecture

For the WineQuality dataset, we use unlabeled dataset size of $U = 4100$, a training set size of 136, and a test set size of 1059. For BreastCancer, we use an unlabeled dataset of size $U = 370$, a training set of size 85, and a test set of size 114.