Machine Learning with Applications 25 (2026) 100949

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Bayesian models in Federated Learning architectures

Zsolt András Romhányi [a], Alex Kummer [a,b,*]

[a] University of Pannonia, Department of Process Engineering, Veszprém, 8200, Hungary
[b] HUN-REN-PE Complex Systems Monitoring Research Group, Department of Process Engineering, University of Pannonia, Veszprém, 8200, Hungary

ARTICLE INFO

Keywords:
Naive Bayes
Federated Learning
Federated Discretization
Non-IID

ABSTRACT

This work investigates the use of Naive Bayes models in Federated Learning settings and proposes an FLNB procedure for decentralized structured classification tasks. The motivation comes from application domains such as healthcare, finance and industrial analytics, where models trained from data originating from multiple institutions would be useful, but the centralization of raw data is constrained by legal, ethical or organizational limitations. The proposed procedure combines federated preprocessing with Naive Bayes training. Continuous variables are handled by iterative federated binning, while discrete and categorical values are encoded by deterministic hashing, which provides cross-client representational consistency without requiring a shared plaintext category dictionary. For model aggregation, we introduce the DirSign method, which updates global class-conditional distributions based on correction directions rather than by directly aggregating full local count tables. The method was evaluated on four public structured classification datasets under IID and Dirichlet-based non-IID client partitions. The results show that FLNB-DirSign approaches the performance of the centralized reference model on several datasets while using less directly interpretable local distributional information. The experiments highlight the importance of discretization, smoothing, hash-space size, learning rate and client heterogeneity.

1. Introduction

Different machine learning models and other data-driven models are playing an increasingly important role in domains where predictive performance could be substantially improved by jointly exploiting samples originating from multiple institutions, organizations or data sources. This is particularly relevant in healthcare, finance and other industrial applications, where local data often contain sensitive personal, business or technological information. In such situations, collecting raw data, or even storing it in a single location, is often not permitted for legal, ethical, security or organizational reasons, and may also be impractical. Federated Learning (FL) addresses this problem by allowing institutions, organizations or clients to keep raw data locally, while the central server receives only locally produced model updates or aggregatable information (Li et al., 2020; B. Liu et al., 2024; McMahan et al., 2017).

This approach can reduce data-protection risks; however, it is important to emphasize that it does not by itself provide a formal privacy guarantee. Although raw data do not leave the clients, model updates, metadata and aggregated information may also carry information about the local data. The literature on gradient inversion and membership inference attacks shows that, under certain conditions, partial reconstruction or inference of local data can be possible from communicated information (Carletti et al., 2025; Chen et al., 2024; Huang et al., 2021). Stronger privacy-preserving mechanisms can be used to reduce information leakage, that is, the amount of information that communicated messages contain about the underlying raw data. Examples include secure aggregation and differential privacy, but these mechanisms may increase system complexity and communication requirements, and can also lead to measurable losses in predictive performance (Banse et al., 2024; Bonawitz et al., 2017; Marchioro et al., 2022; Zhang et al., 2025). For this reason, methods that do not provide formal cryptographic or differential privacy guarantees, but aim to reduce direct data, semantic and model-update exposure, are also of practical interest.

The present work investigates the applicability of Naive Bayes classifiers in such Federated Learning architectures. Naive Bayes (NB) models are simple, interpretable, computationally lightweight, and their parameters can be directly estimated from occurrence frequencies or conditional probability estimates (Mitchell, 1997; Murphy, 2012). These properties make them particularly suitable for studying Federated Learning procedures in which the aim is to develop a comprehensive, low-communication and lower-exposure classification model architecture. The federated use of the Naive Bayes model family can generally be understood as follows: clients locally estimate class priors and the parameters or sufficient statistics of class-conditional

* Corresponding author.
E-mail addresses: romhanyi.zsolt.andras@mk.uni-pannon.hu (Z.A. Romhányi), kummer.alex@mk.uni-pannon.hu (A. Kummer).

https://doi.org/10.1016/j.mlwa.2026.100949
Received 2 May 2026; Received in revised form 28 June 2026; Accepted 2 July 2026
Available online 9 July 2026

distributions, and the server constructs a global model from these quantities. The different NB variants differ in the assumptions they make about the distributions of input variables and, accordingly, in the types of local statistics that must be aggregated. Thus, federated NB training is not restricted to a single model variant, but can be interpreted as a broader training logic based on aggregating locally estimated NB parameters or statistics. At the same time, these models often require discrete, non-negative category-indexed inputs, which raises a preprocessing issue for mixed-type datasets (Han et al., 2011; scikit-learn developers, 2025).

In this work, we present and evaluate this general NB-based federated training logic through the Categorical Naive Bayes model. The reason for this choice is that real datasets often contain mixed data types, including continuous and categorical variables, which can be represented in a unified categorical form after suitable discretization and encoding. This choice is also methodologically advantageous because the model parameters can be interpreted as frequencies associated with discrete categories and target classes, or as conditional probabilities derived from these frequencies. At the same time, Categorical Naive Bayes can only handle input data represented as discrete, non-negative category indices, which gives rise to the preprocessing issue discussed above (Han et al., 2011; scikit-learn developers, 2025).

In a Federated Learning setting, it is therefore not sufficient merely to aggregate the parameters of local NB models according to an existing method. Continuous variables must be identified within the procedure and then transformed into discrete categories that can be handled by Categorical NB. This should preferably be done in such a way that the discretization configuration is interpreted consistently across clients while raw data are not transferred to the central server. For categorical variables, it must be ensured that identical categories represent the same values independently of the client. A further difficulty is that, in practice, client data distributions are often non-IID, meaning that the distributions across clients show biased and uneven patterns. Therefore, the discretization and aggregation steps within the procedure must handle client heterogeneity and locally distorted class distributions (Dominini et al., 2025; Jimenez et al., 2024).

Our objective is to develop and evaluate a Naive Bayes-based federated training and preprocessing procedure in which both preprocessing and aggregation steps are adapted to the decentralized data environment. The concrete implementation and experimental evaluation of the procedure are presented through the Categorical Naive Bayes model, because this variant particularly clearly illustrates the discretization and categorical-representation problems arising from mixed-type data. One novelty of the proposed procedure is that it applies Federated Discretization based on locally computed impurity scores for continuous variables, ensures cross-client consistency of categorical representations through hash-based encoding, and investigates a sign-based aggregation method for updating global conditional probability distributions. The contribution of the work is to examine how the local-statistics-based training logic of the Naive Bayes model family can be adapted to a federated setting, with particular attention to preprocessing, categorical-representation consistency and aggregation. The aim of the method is not to provide formal differential privacy or cryptographic security guarantees, but to develop and evaluate a privacy-conscious and exposure-reducing approach. Accordingly, raw values are not shared, direct semantic disclosure is reduced, and the level of detail of the information transmitted during aggregation is also limited. The proposed procedure is evaluated on several public datasets using centralized reference models, hyperparameter sensitivity analyses, ablation comparisons and non-IID client partitioning settings.

The remainder of the paper is organized as follows. Section 2 summarizes the notation and abbreviations used throughout the manuscript. Section 3 reviews the literature background of Federated Learning, Naive Bayes models, privacy risks and the related research gap. Section 4 presents the proposed FLNB procedure, including federated preprocessing, deterministic hashing and DirSign aggregation. Section 5 describes the experimental protocol, while Section 6 presents and interprets the results. Finally, Section 7 summarizes the conclusions and outlines future research directions.

2. Notation and abbreviations

The main mathematical notation used in this work is summarized in Table 1.

3. Literature review

The architecture proposed in this work builds on several methodological areas, including Federated Learning, Naive Bayes models and data preprocessing. This section reviews the results and methodological foundations that are directly related to the proposed FLNB procedure.

3.1. Federated Learning

Federated Learning is a machine learning paradigm in which several clearly separated clients jointly participate in the training of a predictive model without directly transferring raw local data to a central location. The central actor is typically a server, which coordinates the training process, sends models or configurations to the clients, and then uses local updates or aggregatable statistics received from the clients to construct a global model. The main motivation of Federated Learning is to enable model development from multiple data sources in settings where sharing raw data is not feasible for legal, privacy-related, business or technological reasons (B. Liu et al., 2024; McMahan et al., 2017).

One fundamental classification of FL systems is based on how data are distributed across clients. Based on this, horizontal, vertical and hybrid Federated Learning can be distinguished. In horizontal FL, the clients have the same variable structure but store different samples. This is typical for joint model development by similar institutions or organizations, such as hospitals or financial institutions, where each institution measures similar types of variables but observes different patients or customers. In vertical FL, the clients possess different variables of the same or partially overlapping samples, while hybrid settings may involve distribution along both the sample and variable dimensions (Y. Liu et al., 2024; Yang et al., 2023). The present work focuses on horizontal Federated Learning; in the experiments, clients use the same feature space but have disjoint sample sets.

The classical federated training process is iterative and can be divided into communication rounds. First, the server initializes a global model or model configuration and sends it to the selected clients. The clients perform local training steps on their own data and then send back the locally produced model parameters or related information in the form of model updates. The server constructs a new global model from these updates using an aggregation rule, and this model is sent to the clients again in the next communication round. The process may continue for a predefined number of communication rounds or until a convergence criterion is satisfied (B. Liu et al., 2024; McMahan et al., 2017).

Fig. 1 illustrates the basic server-client communication structure of Federated Learning. The server does not collect raw data; instead, it coordinates training and aggregates updates. The clients perform training on their own local data and only send back predefined updates, parameters or aggregatable statistics.

The general server-client training cycle is summarized in Algorithm 1.

An important task on the server side is aggregation, which is one of the central methodological elements of FL and can be implemented in several ways. The simplest and one of the most widely used approaches is averaging local model parameters, known as Federated Averaging (FedAvg). In settings with heterogeneous client data, modified aggregation or optimization schemes are also often used, for example proximal regularization, adaptive server-side optimization, robust aggregation, or communication-reducing sign-based updates (Bernstein et al., 2018, 2019; Li et al., 2020; McMahan et al., 2017; Reddi et al., 2021; Torrijos

Table 1
Main mathematical notation used in the methodological description.

Notation	Meaning
General Federated Learning notation	
K	Number of clients.
k	Client index, where $k \in 1, \dots, K$.
S_t	Set of clients selected in the tth communication round.
T	Number of general Federated Learning communication rounds.
E	Number of local optimization steps or epochs in the illustrative FedAvg algorithm.
w_t	Global model parameters in the tth Federated Learning communication round.
$w^{(k)} * t+1$	Locally updated model parameters of the kth client.
n_k, n_t	Sample size of the kth client in the FedAvg notation, where $n_t = \sum * k \in S_t n_k$.
Data structure and classification task	
D	Full experimental dataset.
D_k	Local dataset of the kth client.
D_{train}	Training set, partitioned into client subsets in the federated simulation.
m, m_k	Total sample size and local sample size of the kth client in the FLNB notation.
$x^{(i)}, y^{(i)}$	Input vector and target class of the ith sample.
$\mathcal{X}$	Feature space or feature list.
x_j	jth input variable.
d	Number of input variables.
$\mathcal{Y}, C$	Set of class labels and number of classes.
Continuous feature identification and federated binning	
$u^{(k)} * j$	Number of unique values locally observed for the jth feature at the kth client.
$\tau *$ unique	Unique-value threshold for identifying numerical features to be treated as continuous.
$F^{(k)} *$ cont	Set of features identified as continuous by the kth client.
$F *$ bin	Global set of features selected for federated discretization.
$R_j^{(k)}$	Local value range of the jth feature at the kth client.
$\rho_j^{(k)}, \rho_j^{\text{glob}}$	Local and global order-of-magnitude scale estimate for initializing binning.
$B_j, B_j^{(r)}$	Candidate bin width set of the jth feature and its version in iteration r.
$s^{(k)} * j(b), s_j(b)$	Local and aggregated binning quality score for candidate bin width b.
$R *$ bin	Number of federated binning search rounds.
$b_j^{*,(r)}, b^* * j$	Bin width selected in round r, and final bin width for the jth feature.
$B *$ max	Maximum number of allowed bins per feature.
L_j, U_j	Lower and upper range bounds used for binning the jth feature.
$\tilde{x}_j$	Discretized value of the jth feature represented as a bin index.
Hashing and Categorical Naive Bayes representation	
$h_j(v)$	Deterministic hashing function mapping value v of feature x_j to an integer index.
N_{hash}	Number of hash buckets, i.e., the size of the hash space.
$\hat{x}_{k,j}^{(i)}$	Model-compatible value of the jth feature of the ith sample of the kth client after hashing or binning.
$\hat{D}_k$	Preprocessed local dataset of the kth client compatible with Categorical Naive Bayes.
C_j	Number of possible category values of the jth feature in the Categorical Naive Bayes model.
Naive Bayes model and local statistics	
α	Laplace or Lidstone smoothing parameter.
$N^{(k)} * y$	Number of samples belonging to class y at the kth client.
$N^{(k)} * j, c, y$	Local joint frequency of $x_j = c$ and y at the kth client.
$P_k(y), P_g(y)$	Local and global class prior.
$P_k(x_j = c \mid y)$	Local class-conditional probability of the kth client.
$P_g(x_j = c \mid y)$	Global class-conditional probability.
$P_g^{(t)}(x_j = c \mid y)$	Global class-conditional probability in the tth DirSign iteration.
DirSign aggregation and non-IID partitioning	
$D^{(k)} * j, c, y$	Difference between local and global class-conditional PMFs at the kth client.
$\Delta^{(k)} * j, c, y$	DirSign update direction of the kth client for feature j, category c and class y.
$\Delta_{j,c,y}$	Server-side aggregated DirSign update direction.
η	DirSign learning-rate parameter.
δ	Small numerical stabilizing constant in DirSign normalization and PMF stabilization.
$\tilde{P}^{(t+1)} * g(x_j = c \mid y)$	Temporary global PMF value before normalization after the DirSign update.
$T *$ max	Maximum number of DirSign aggregation rounds.
ϵ	DirSign convergence threshold.
$r^{(t)}$	Relative change between consecutive DirSign iterations.
$\pi_c, \pi_{c,k}$	Client proportion vector sampled for class c during Dirichlet partitioning, and its kth component.
α_{Dir}	Heterogeneity parameter of Dirichlet-based non-IID partitioning.

et al., 2025). FedAvg is well suited for illustrating the logic of general FL aggregation.

Federated Averaging is one of the most widely used aggregation methods in FL. In this approach, the server aggregates local model parameters, typically weighted in proportion to the number of samples at each client (McMahan et al., 2017). If w_t denotes the global model parameters in the tth communication round and $w_{t+1}^{(k)}$ denotes the locally updated parameters of the kth client, a typical FedAvg aggregation step is

$$w_{t+1} = \sum_{k \in S_t} \frac{n_k}{\sum_{\ell \in S_t} n_\ell} w_{t+1}^{(k)}, \tag{1}$$

where S_t is the set of clients participating in the tth round and n_k is the local sample size, or a quantity proportional to it, for the kth client. This scheme is simple and intuitive, but sample-size weighting may be sensitive to client heterogeneity and differences in client sizes.

These points show that the practical application of FL involves several system-level and methodological challenges. Examples include non-IID client data, partial client participation, communication cost, client-side computational constraints and the robustness of aggregation. The non-IID nature of client distributions is particularly important, because local models trained under different class distributions, feature distributions or measurement environments may modify the global

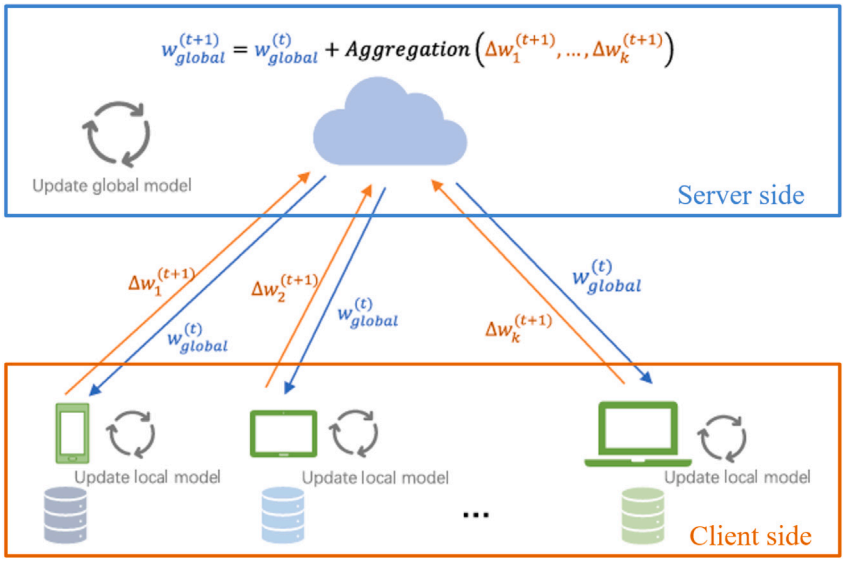

Fig. 1. General server-client architecture of federated learning (Ma et al., 2021).

Algorithm 1: General Federated Averaging (FedAvg) training cycle (McMahan et al., 2017)

Input: initial global model parameters w_0; number of communication rounds T; number of local optimization steps E
Output: final global model parameters w_T

for $t = 0, 1, \ldots, T-1$ **do**
  The server selects the participating client set $S_t \subseteq \{1, \ldots, K\}$
  The server sends the current global model w_t to each selected client
  foreach $k \in S_t$ *in parallel* **do**
    Client k locally optimizes the model on its own D_k data
    $w_{t+1}^{(k)} \leftarrow \text{LocalUpdate}(w_t, D_k, E)$
    The client sends $w_{t+1}^{(k)}$ back to the server
  $n_t \leftarrow \sum_{k \in S_t} n_k$
  $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n_t} w_{t+1}^{(k)}$
return w_T

model in different directions. This can lead to client drift, slower or unstable convergence, and the dominance of certain clients (Dominini et al., 2025; Jimenez et al., 2024).

Several federated optimization and aggregation methods have been proposed to address these challenges. FedProx aims to reduce excessive drift of local models away from the global model by adding a proximal term to the local objective function (Li et al., 2020). Adaptive federated optimization methods, such as FedAdam or FedYogi, aim to scale and stabilize the server-side update (Reddi et al., 2021). Other directions focus on communication reduction, robust aggregation or the integration of privacy-preserving mechanisms (Bernstein et al., 2018, 2019; Bonawitz et al., 2017; Yin et al., 2018).

While gradient-based models focus on aggregating local model parameters or gradient updates, NB models make the aggregation of locally computable class priors, class-conditional frequencies, moments or probability mass distributions central. These quantities can carry considerable information about the local data of a client and therefore may involve data-protection risks when communicated directly.

3.2. Naive Bayes models

Naive Bayes classifiers are generative probabilistic models based on Bayes' theorem and on the assumption that input variables are conditionally independent given the target class. For an input vector $x = (x_1, \ldots, x_d)$ and a target class y, the posterior probability is proportional to the product of the class prior and the class-conditional likelihood:

$$P(y \mid x) \propto P(y) \prod_{j=1}^{d} P(x_j \mid y). \tag{2}$$

In practice, the decision rule is usually used in logarithmic form:

$$\hat{y} = \arg\max_y \left[\log P(y) + \sum_{j=1}^{d} \log P(x_j \mid y) \right]. \tag{3}$$

The conditional independence assumption is often not strictly satisfied on real datasets. Nevertheless, Naive Bayes models provide stable, interpretable and computationally efficient baselines in many classification tasks (Domingos & Pazzani, 1997; Murphy, 2012; Zhang, 2004).

Different variants of Naive Bayes differ in the distributional assumption made for the input variables. Gaussian Naive Bayes assumes class-specific normal distributions for continuous variables, and therefore requires class-specific means, variances and sample counts for parameter estimation. Bernoulli Naive Bayes is naturally suited to binary variables, Multinomial NB to count or frequency data, and Categorical NB to discrete, multi-valued categorical variables (Mitchell, 1997; Murphy, 2012; scikit-learn developers, 2025).

Thus, applying NB models in an FL setting requires the aggregation of different types of local statistics. Table 2 summarizes the most important NB models and the local statistics that can be aggregated in a federated setting.

In the table, N_y denotes the number of samples belonging to a given target class. The quantities S_{yj} and Q_{yj} are required for moment-based class-wise parameter estimation in Gaussian NB, while M_{yj} and M_{yjv} denote class-specific occurrence counts. The table shows that federated training of the Naive Bayes model family naturally leads to the aggregation of local statistics. The present work concretizes this general logic for Categorical NB, where federated discretization, cross-client consistency of categorical representation and aggregation of conditional probability distributions appear simultaneously.

3.3. Categorical Naive Bayes

The present work concretizes the general logic of Naive Bayes-based federated training through the Categorical NB model. The main reason is that the parameters of Categorical NB can be estimated

Table 2
Naive Bayes variants and locally aggregatable statistics in a federated setting.

NB variant	Class-conditional model	Aggregatable local statistics
Gaussian NB	$x_j \mid y \sim \mathcal{N}(\mu_{yj}, \sigma^2_{yj})$	$N_y = \lvert I_y \rvert, \quad S_{yj} = \sum_{i \in I_y} x_{ij}, \quad Q_{yj} = \sum_{i \in I_y} x^2_{ij}$
Bernoulli NB	$x_j \in \{0,1\}, \ P(x_j = 1 \mid y) = \theta_{yj}$	$N_y = \lvert I_y \rvert, \quad M_{yj} = \sum_{i \in I_y} \mathbb{I}(x_{ij} = 1)$
Multinomial NB	$x \mid y \sim \text{Multinomial}(\theta_y)$	$N_y = \lvert I_y \rvert, \quad M_{yj} = \sum_{i \in I_y} x_{ij}$
Categorical NB	$x_j \in \{1, \dots, C_j\}, \ P(x_j = v \mid y) = \theta_{yjv}$	$N_y = \lvert I_y \rvert, \quad M_{yjv} = \sum_{i \in I_y} \mathbb{I}(x_{ij} = v)$

directly from class- and category-dependent occurrence frequencies, which can be computed locally and aggregated on the server side. This fits the Federated Learning setting in which raw records are not centralized and clients transmit only aggregatable statistical information or direction-like updates.

The use of Categorical NB also creates a preprocessing requirement: every input feature must be available as a discrete, non-negative category index (scikit-learn developers, 2025). This requirement can be satisfied easily for datasets that already consist of categorical or discrete variables. However, practical classification tasks often contain continuous numerical, discrete numerical and categorical variables at the same time. In such cases, Categorical NB can be used in a unified way only if continuous variables are discretized and categorical values are mapped to a cross-client consistent numerical representation. Discretization and category encoding are therefore not merely technical preprocessing steps, but part of the federated model construction (Han et al., 2011).

A further reason for choosing this model is interpretability and low computational demand. Naive Bayes models work with parameters that can be estimated in closed form, do not require complex structural learning, and can serve as stable baselines even for small or medium-sized local datasets (Domingos & Pazzani, 1997; Murphy, 2012; Zhang, 2004). In a federated setting, this is advantageous because it reduces client-side computational burden, and the information obtained from local models appears in the form of interpretable probability distributions.

Recent work on federated NB also indicates that NB models can be naturally adapted to decentralized training environments. However, these works often assume variables that are already discrete, or they focus on other aspects of parameter learning, such as discriminative NB variants (Marchioro et al., 2022; Torrijos et al., 2025). The present work investigates how an FLNB pipeline can be constructed in which the discrete representation required by Categorical NB, cross-client category consistency and the estimation of global NB parameters are all treated as part of a single federated workflow.

3.4. Privacy risks in Federated Learning

One of the most important advantages of Federated Learning is that raw data remain at the clients. However, this does not by itself constitute a formal privacy guarantee. Model updates, gradient information, aggregated statistics and preprocessing metadata transmitted to the server may still contain information about the local data distribution. This is particularly important in application domains where client data are sensitive from a personal, business or technological point of view, and where even partial information leakage can be risky (Chen et al., 2024; B. Liu et al., 2024).

The approach examined in this work does not introduce a formal differential privacy mechanism and does not implement a cryptographic secure aggregation protocol. The goal is to develop a procedure in which the level of detail and direct interpretability of information communicated to the server during federated Naive Bayes training can be reduced already at the level of model construction and preprocessing. This approach does not replace formal privacy-preserving methods; rather, it investigates model- and communication-level information reduction that can later be combined with secure aggregation or differential privacy mechanisms.

This issue is especially relevant for Naive Bayes models because their parameter estimation is often reducible to class priors, class-conditional frequencies and other locally computable sufficient statistics. These statistics are favorable from computational and communication perspectives, but in the case of small local sample sizes, rare categories or strongly skewed class distributions, they may carry information about the structure of the client data. Therefore, privacy-conscious design for federated NB models cannot be limited to the aggregation rule alone. Preprocessing, categorical representation and model-parameter communication jointly determine the level of detail and semantic interpretability of the information obtained by the server about the local data distributions.

3.5. Research gap

Based on the literature review above, Federated Learning, Naive Bayes models and privacy risks associated with federated training are each widely studied research areas. Most federated learning approaches, however, focus on gradient-based or parameter-averaging models, where the main issue is the efficient, stable and preferably secure aggregation of local model updates (B. Liu et al., 2024; McMahan et al., 2017). In contrast, federated training of Naive Bayes models represents a different type of problem because model parameters can often be estimated from locally computable class priors, class-conditional frequencies or other aggregatable statistics.

For Categorical Naive Bayes, this issue is further extended by preprocessing. The model assumes input variables represented by discrete, non-negative category indices, whereas real datasets often contain continuous, discrete numerical and categorical variables simultaneously (Murphy, 2012; scikit-learn developers, 2025). In a centralized setting, discretization, category encoding and unified variable representation can be performed using the complete training set. In a decentralized setting, however, raw data are separated by client, and direct centralization of the same preprocessing steps is undesirable and often not permitted.

It therefore remains an open question in the literature how, for mixed-type data, a Federated Learning Naive Bayes architecture can be constructed that simultaneously handles discretization, cross-client consistency of categorical representation, and the level of detail of the information communicated during aggregation. This question is also relevant from a privacy perspective, because keeping raw data locally does not by itself provide a formal privacy guarantee, and model updates, aggregated statistics and preprocessing metadata may still carry information about local data distributions (Chen et al., 2024; Huang et al., 2021).

The present work investigates how federated discretization of continuous variables, hash-based categorical representation and direction-based aggregation can be integrated into a single training workflow, and how these design choices affect predictive performance, robustness and the level of directly communicated local distributional information.

It is important to emphasize that this approach does not replace secure aggregation or differential privacy methods, and it does not claim formal cryptographic or differential privacy guarantees (Banse et al., 2024; Bonawitz et al., 2017).

4. Methodology

This section presents the proposed Federated Learning-based Naive Bayes (FLNB) methodological procedure. The aim is to construct a modular Naive Bayes-based training process in which local preprocessing, the establishment of cross-client representational consistency, and the estimation of a global probabilistic model appear as parts of a single federated training process. The present study concretizes this procedure through the Categorical Naive Bayes model, because this model requires a discrete, non-negative feature representation and therefore makes preprocessing an integral part of model construction (scikit-learn developers, 2025).

The method assumes a horizontal Federated Learning setting, where clients share the same feature space but store different samples. Raw records remain local throughout the complete process; the server uses only the aggregatable preprocessing information, local model statistics or sign-based updates defined by the protocol. The federated trainability of Naive Bayes models follows from the fact that their parameters can be estimated from class priors and class-conditional feature distributions, which can be computed locally and aggregated on the server side (Mitchell, 1997; Murphy, 2012; Torrijos et al., 2025).

For Categorical Naive Bayes, the class-conditional feature distribution is a probability mass function (PMF) defined over discrete category values. This means that, for every feature x_j and class y, the model estimates the category-conditional probabilities $P(x_j = c \mid y)$, where c denotes the possible discrete category values of the feature. Consequently, a unified and cross-client consistent integer representation of continuous and categorical variables is a core part of the method.

The FLNB procedure can be divided into three main methodological blocks. The first block is federated preprocessing, in which clients locally identify features to be treated as continuous and then perform federated binning for these features. The second block is deterministic hashing, which provides unified encoding of discretized and originally categorical features without a shared plaintext dictionary. The third block is the federated estimation of the Naive Bayes model, in which the server constructs global class priors and class-conditional PMFs from local statistics or sign-based updates. The complete process is in algorithms in Appendix A.

To make the methodological steps easier to follow, consider an example in which two clients have the same feature structure but different local samples. Let x_a denote a numerical feature taking many different values, and let x_o denote a nominal categorical feature. In the following, x_a is used to illustrate features treated as continuous, while x_o illustrates categorical features requiring hashing. The example is used only to clarify the notation and the methodological steps; the algorithm is defined independently of it for any set of clients with a common feature space. In later chapters, we will demonstrate each step through this example.

4.1. Federated Learning setting and FLNB formulation

Let the complete, theoretically defined dataset be

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}, \tag{4}$$

where $x^{(i)} \in \mathcal{X}$ is the feature vector of the ith sample, $y^{(i)} \in \mathcal{Y} = \{y_1, \ldots, y_C\}$ is the associated class label, C is the number of classes, and m is the total number of samples.

In a Federated Learning setting, the complete dataset $\mathcal{D}$ is not available at a single central location, but is distributed across K clients:

$$\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k, \qquad m_k = |\mathcal{D}_k|, \qquad \sum_{k=1}^{K} m_k = m. \tag{5}$$

Here, $\mathcal{D}_k$ is the local dataset of the kth client. The client index is denoted by k; the notation C is used only for the number of classes.

When public datasets are artificially partitioned into clients in the experimental section, this serves only to simulate a federated environment and is not part of the operational assumptions of the actual protocol.

The Naive Bayes model is based on the conditional independence assumption:

$$P(y \mid x) \propto P(y) \prod_{j=1}^{d} P(x_j \mid y), \tag{6}$$

where d is the number of features. Accordingly, the decision rule of the global Naive Bayes classifier is

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \left[\log P_g(y) + \sum_{j=1}^{d} \log P_g(x_j \mid y) \right], \tag{7}$$

where $P_g(y)$ is the global class prior and $P_g(x_j \mid y)$ is the global class-conditional distribution of the jth feature.

In the present implementation, each $P_g(x_j \mid y)$ term is a PMF defined over discrete category values. Therefore, every feature must be available as a non-negative integer category index. In the example introduced above, the numerical feature x_a must first be transformed into a bin index, while the categorical feature x_o must be transformed into a hash index before the Categorical Naive Bayes statistics can be estimated.

4.2. Threat model and scope of privacy protection

The FLNB procedure assumes a limited honest-but-curious server model. In this setting, the server follows the prescribed protocol, but may attempt to infer certain properties of local datasets from the messages it receives. The clients are assumed to be protocol-following, meaning that they execute the specified computations correctly and do not deliberately attempt to distort the global model.

The present work does not investigate malicious client behavior, Byzantine attacks, poisoning attacks, client–server collusion, secure multiparty computation protocols or formal differential privacy guarantees. Accordingly, the method should not be interpreted as a complete cryptographic privacy protocol. The privacy-related objective is narrower: avoiding the direct sharing of raw records, reducing the semantic exposure of categorical variables, and reducing the level of detail of local distributional information.

The type of information transmitted to the server in the main communication steps is summarized in the Table B.13.

4.3. Multi-phase preprocessing for Naive Bayes models

The proposed procedure applies multi-phase federated preprocessing. Its purpose is to transform different types of features stored by the clients into a unified, discrete representation that can be used directly by the Categorical Naive Bayes model. Because of the category-index requirement established in the previous subsection, preprocessing is not a centrally performed technical step before training, but an integrated part of federated training.

The preprocessing consists of three main operations. First, clients locally identify the features to be treated as continuous. Second, federated binning is performed for these features. Third, discretized and originally categorical variables are mapped into a unified integer representation by deterministic hashing. These steps correspond to lines 1–7 of Algorithm 2.

4.3.1. Continuous feature identification

The purpose of continuous feature identification is to decide which numerical features should be treated as continuous variables and which can be used directly as discrete or categorical variables. This decision is important for Categorical Naive Bayes because continuous variables must be assigned to bins, whereas variables that are already discrete or categorical can be used directly after appropriate encoding. This step is related to lines 1–2 of Algorithm 2.

The method uses a unique-value threshold τ_{unique}. For each numerical feature x_j, the kth client locally determines the number of unique values:

$$u_j^{(k)} = |\text{Unique}_k(x_j)|, \tag{8}$$

where $\text{Unique}_k(x_j)$ is the set of distinct values locally observed for the jth feature at the kth client. If

$$u_j^{(k)} > \tau_{\text{unique}}, \tag{9}$$

then the client identifies the feature as a continuous candidate. The set of features marked as continuous by the kth client is

$$F_{\text{cont}}^{(k)} = \{x_j : u_j^{(k)} > \tau_{\text{unique}}\}. \tag{10}$$

The server does not receive local unique values or raw feature values, but only the list of features identified as continuous. The globally discretized feature set is obtained as the union of the client-side markings:

$$F_{\text{bin}} = \bigcup_{k=1}^{K} F_{\text{cont}}^{(k)}. \tag{11}$$

This rule is conservative. If a feature behaves as continuous at least at one client, it is treated as a feature to be discretized in the entire federated process. This reduces the risk that a high-cardinality numerical variable is introduced into the Categorical Naive Bayes model as discrete at some clients, which could lead to a sparse and excessively large category space. In the example, x_a enters the set F_{bin} if the number of unique values exceeds the threshold τ_{unique} at at least one client; in contrast, x_o is not a numerical continuous candidate and therefore obtains a model-compatible representation later, through hashing. The threshold can be selected based on computational capacity and heuristic considerations.

4.3.2. Federated binning of continuous features

Binning continuous variables is one of the key steps for applying Categorical Naive Bayes. In centralized training, bin boundaries could be determined from the complete dataset. In a federated environment, however, this is not feasible because directly sharing raw feature values, exact distributions or exact minimum and maximum values may reveal information about the structure of local data. Therefore, the FLNB procedure decomposes binning into client-side score computation and server-side aggregation. This step corresponds to lines 3–6 of Algorithm 2.

The first part of the binning process is a scale-estimation round. Each client locally determines the order of magnitude of the value range of the feature $x_j \in F_{\text{bin}}$. The local range is

$$R_j^{(k)} = \max(x_j^{(k)}) - \min(x_j^{(k)}). \tag{12}$$

The client does not transmit the range itself, but a low-resolution order-of-magnitude representation of the range:

$$\rho_j^{(k)} = \left\lfloor \log_{10} R_j^{(k)} \right\rfloor. \tag{13}$$

From these values, the server constructs a feature-wise global scale estimate:

$$\rho_j^{\text{glob}} = \max_k \rho_j^{(k)}. \tag{14}$$

This scale information does not reveal exact local minima and maxima, but it is sufficient for the server to generate initial candidate bin width values.

For every feature $x_j \in F_{\text{bin}}$, the server defines a candidate bin width set:

$$\mathcal{B}_j = \{b_{j,1}, b_{j,2}, \ldots, b_{j,M}\}, \tag{15}$$

where $b_{j,m}$ is a possible bin width. The clients evaluate every candidate bin width locally. The score computed by the kth client is

$$s_j^{(k)}(b). \tag{16}$$

The score measures the quality of the binning with respect to the target variable. The implementation supports several metrics, such as class-frequency variance, Gini-index-based scores and entropy-based scores. Their optimization directions may differ: for a variance-based score, a larger value indicates better separation of class proportions, whereas for Gini and entropy, a smaller value indicates purer bins. Therefore, the procedure applies a maximization or minimization rule depending on the selected metric:

$$b_j^{*,(r)} = \begin{cases} \arg\max_{b \in \mathcal{B}_j^{(r)}} s_j(b), & \text{for a variance-based score,} \\ \arg\min_{b \in \mathcal{B}_j^{(r)}} s_j(b), & \text{for Gini- or entropy-based scores.} \end{cases} \tag{17}$$

Here, r denotes the current iteration of the binning search, and

$$s_j(b) = \frac{1}{K} \sum_{k=1}^{K} s_j^{(k)}(b) \tag{18}$$

is the aggregated server-side score.

The binning search is iterative. In the first round, the server examines a broader candidate bin width range, and then narrows the search interval around the best $b_j^{*,(r)}$:

$$\mathcal{B}_j^{(r+1)} = \text{Refine}\left(\mathcal{B}_j^{(r)}, b_j^{*,(r)}\right). \tag{19}$$

The process stops after the predefined number R_{bin} of binning rounds, and the final bin width is

$$b_j^* = b_j^{*,(R_{\text{bin}})}. \tag{20}$$

Using these b_j^* values, clients locally transform their continuous feature values into bin indices. If $[L_j, U_j)$ denotes the globally used feature range, the bin index of a value x_j is

$$\widetilde{x}_j = \left\lfloor \frac{x_j - L_j}{b_j^*} \right\rfloor. \tag{21}$$

In practice, this means that the raw numerical values of feature x_a are transformed into category indices at every client using the same server-selected bin width b_a^*.

During testing, a continuous feature value may fall outside the binning range determined during training. In this case, the corresponding bin index is clipped to the nearest valid bin. For previously unseen categorical values, the same deterministic hashing function is applied; therefore, the value is mapped to one of the indices in the fixed hash space.

4.3.3. Consistency-preserving hashing of categorical values

Discrete numerical, discretized continuous and originally categorical features all require a unified numerical representation. In a centralized setting, this is often done by label encoding, where a shared dictionary is created based on the category values of the complete dataset. In a federated setting, however, constructing a shared plaintext dictionary is undesirable because it may reveal which category values occur in the local data of clients. This step is related to line 7 of Algorithm 2.

The FLNB procedure therefore uses deterministic hashing-based encoding. The hash index of a category value v of feature x_j is

$$h_j(v) = \text{hash}(j, v) \bmod N_{\text{hash}}, \tag{22}$$

where N_{hash} is the size of the hash space. A concrete deterministic hash function can be used in the implementation; from the methodological

perspective, the key point is that identical inputs are mapped to identical indices at every client.

The transformed local dataset is

$$\widehat{D}_k = \{(\hat{x}_k^{(i)}, y_k^{(i)})\}_{i=1}^{m_k}, \tag{23}$$

where $\hat{x}_{k,j}^{(i)} = h(\tilde{x}_{k,j}^{(i)})$ if the feature requires hashing, and $\tilde{x}_{k,j}^{(i)}$ denotes the discretized or originally categorical feature value.

The size of the hash space is a trade-off. A larger N_{hash} reduces the probability of hash collisions, but increases the category space of the Categorical Naive Bayes model. A smaller N_{hash} yields a more compact representation, but increases the chance that different category values are mapped to the same index. The role of hashing is therefore twofold: it ensures cross-client representational consistency and avoids the construction of a shared plaintext dictionary. Thus, the possible category values of feature x_o obtain integer indices through the function $h_j(\cdot)$, without direct binning, and these indices identify the same categories across clients.

4.4. Federated categorical Naive Bayes training

After federated preprocessing, every client has a local dataset in which the features appear as discrete, non-negative category indices. This enables the estimation of local statistics for the Categorical Naive Bayes model. For training, we propose the DirSign PMF aggregation variant, which uses sign-based updates instead of full local frequency tables. These steps correspond to lines 8–15 of Algorithm 2.

4.4.1. Local class-conditional probability estimation

The kth client locally estimates class frequencies and class-conditional feature frequencies. Let $N_y^{(k)}$ denote the number of samples belonging to class y at the kth client, and let $N_{j,c,y}^{(k)}$ denote the number of samples for which the jth feature has value c while the class label is y.

The local class-conditional probability can be estimated using Laplace or Lidstone smoothing:

$$P_k(x_j = c \mid y) = \frac{N_{j,c,y}^{(k)} + \alpha}{N_y^{(k)} + \alpha C_j}, \tag{24}$$

where $\alpha > 0$ is the smoothing parameter and C_j is the number of possible category values of the jth feature. The purpose of smoothing is to avoid zero probabilities, which is particularly important for rare categories, small client-side sample sizes or non-IID partitions (Manning et al., 2008; scikit-learn developers, 2025). The local class prior is

$$P_k(y) = \frac{N_y^{(k)}}{m_k}. \tag{25}$$

For features x_a and x_o, this step is interpreted in the same way: the clients count how often a given bin index or hash index occurs within each class. From these local frequencies, the class-conditional probability $P_k(x_j = c \mid y)$ can be computed.

4.4.2. Direct count-based aggregation

The simplest way to obtain a global Categorical Naive Bayes model is to aggregate local frequencies. This direct count-based aggregation is statistically well interpretable and serves as a natural baseline compared with the sign-based aggregation variant.

The global class prior is

$$P_g(y) = \frac{\sum_{k=1}^{K} N_y^{(k)}}{\sum_{k=1}^{K} m_k}. \tag{26}$$

The global class-conditional probability is

$$P_g(x_j = c \mid y) = \frac{\sum_{k=1}^{K} N_{j,c,y}^{(k)} + \alpha}{\sum_{k=1}^{K} N_y^{(k)} + \alpha C_j}. \tag{27}$$

This solution directly corresponds to the case where local frequency tables can be summed. Its advantages are simplicity and statistical interpretability. Its drawback is that it may require more detailed local frequency information, especially with a large category space or strongly heterogeneous client distributions.

4.4.3. DirSign PMF aggregation

The goal of DirSign aggregation is to update the global class-conditional PMFs not by directly summing full local frequency tables, but by aggregating correction directions between local and global PMFs. The approach is related to the logic of sign-based federated optimization; however, in the present case, the update does not apply to neural network weights, but to the class-conditional PMFs of Categorical Naive Bayes (Bernstein et al., 2018; Guo et al., 2023). For brevity, we refer to this FedSign-inspired, direction-based PMF aggregation as DirSign aggregation.

Let $P_g^{(t)}(x_j = c \mid y)$ denote the current global class-conditional PMF in the tth iteration, and let $P_k(x_j = c \mid y)$ denote the local estimate of the kth client. The client first computes the difference between the current local and global PMFs:

$$D_{j,c,y}^{(k)} = P_k(x_j = c \mid y) - P_g^{(t)}(x_j = c \mid y). \tag{28}$$

The update direction is normalized class-wise. If class y is present at the kth client, that is, $N_y^{(k)} > 0$, then

$$\Delta_{j,\cdot,y}^{(k)} = \frac{D_{j,\cdot,y}^{(k)}}{\|D_{j,\cdot,y}^{(k)}\|_2 + \delta}, \tag{29}$$

where $\delta > 0$ is a small stabilizing constant. If the given class does not occur in the local data of the client, the client does not send an update for that class row. This is especially important in non-IID settings because it prevents missing local classes from producing artificial uniform directions in the global PMF update.

The server aggregates the class-wise local directions using support-based weighting:

$$\Delta_{j,\cdot,y} = \frac{\sum_{k:\, N_y^{(k)} > 0} N_y^{(k)} \Delta_{j,\cdot,y}^{(k)}}{\sum_{k:\, N_y^{(k)} > 0} N_y^{(k)}}. \tag{30}$$

This step ensures that a given class PMF row receives updates only from clients where the class is actually present, and that directions based on very small local support do not dominate the update disproportionately.

The global PMF is then updated along the aggregated direction:

$$\widetilde{P}_g^{(t+1)}(x_j = c \mid y) = P_g^{(t)}(x_j = c \mid y) + \eta \Delta_{j,c,y}, \tag{31}$$

where $\eta > 0$ is the DirSign learning-rate parameter. After the update, values are stabilized and normalized feature-wise and class-wise:

$$P_g^{(t+1)}(x_j = c \mid y) = \frac{\max\left(\widetilde{P}_g^{(t+1)}(x_j = c \mid y), \delta\right)}{\sum_{c'=1}^{C_j} \max\left(\widetilde{P}_g^{(t+1)}(x_j = c' \mid y), \delta\right)}. \tag{32}$$

The iteration runs until a predefined maximum number $T_{\max}$ of steps or, optionally, until a convergence threshold ϵ is reached. The advantage of DirSign aggregation is that the server does not use the direct summation of full local frequency tables, but rather correction directions relative to the current global model. This reduces the level of detail of transmitted local information, while the update is more heuristic and more sensitive to hyperparameters than direct count-based aggregation. Therefore, the two aggregation variants are examined separately in the experimental evaluation.

In the illustrative case, the DirSign step applies not to the raw values of x_a or x_o, but to their already discretized or hashed category indices. For example, if a certain hash index of x_o is more frequent in one class at a client than reflected by the current global PMF, the client computes a correction direction for the corresponding PMF row. The server modifies the global class-conditional distribution based on the directions received from multiple clients and weighted by class support.

Algorithm 2: FLNB procedure: federated preprocessing and Naive Bayes training

Input: local datasets $D_1, \dots, D_K$; feature list; class labels; τ_{unique}; N_{hash}; R_{bin}; α; η; $T_{\max}$; ϵ
Output: global Categorical Naive Bayes classifier F_g

Clients locally identify continuous candidate features; the server constructs $F_{\text{bin}} = \bigcup_k F_{\text{cont}}^{(k)}$
foreach $x_j \in F_{\text{bin}}$ **do**
  The server sends candidate bin width values to the clients
  Clients compute local binning scores and return only the scores
  The server aggregates the scores, refines the search and fixes the final binning configuration b_j^*
foreach *client* $k = 1, \dots, K$ *in parallel* **do**
  Apply the selected binning and deterministic hashing $h_j(v) = \text{hash}(j, v) \bmod N_{\text{hash}}$
  Estimate local quantities $N_y^{(k)}$, $N_{j,c,y}^{(k)}$, $P_k(y)$ and $P_k(x_j = c \mid y)$
The server may construct a count-based reference model from local count statistics
Initialize global class-conditional PMFs for the proposed DirSign branch
for $t = 0, 1, \dots, T_{\max} - 1$ **do**
  foreach *client* $k = 1, \dots, K$ *in parallel* **do**
    Compute class-row-normalized directions $\Delta_{j,\cdot,y}^{(k)}$ only for locally present classes
    Send the directions and class supports to the server, without local count tables
  The server weights and aggregates directions by class support
  The server updates the global PMFs, clips them from below by δ, and normalizes each feature-class row to sum to one
  if *the relative PMF change is below* ϵ **then**
    break
Construct F_g from the final global priors and PMFs
return F_g

5. Experimental evaluation

The purpose of the experiments is to examine the predictive performance and stability of the proposed FLNB procedure under different client partitioning, aggregation and related hyperparameter settings. We evaluated how federated preprocessing, deterministic hashing for cross-client representational consistency, and sign-based aggregation affect model performance.

The experiments examined how the performance of the FLNB procedure compares with centralized and local Naive Bayes models. We also investigated the effectiveness of federated binning, deterministic hashing and DirSign aggregation, as well as the sensitivity of the method's hyperparameters under IID and non-IID data settings.

5.1. Datasets

For the experimental evaluation, four publicly available structured classification datasets were used: Adult Income, Breast Cancer Wisconsin Diagnostic, Credit Score Classification and Forest Cover Type. In selecting the datasets, an important criterion was to include benchmarks with different sizes, class structures and variable types. This makes it possible to evaluate the method not on a single special data structure, but across several classification settings.

The Adult Income dataset originates from the 1994 US census data and is used to predict whether a person's annual income exceeds USD 50,000 (Becker & Kohavi, 1996). The task is a binary classification problem containing both numerical and categorical variables, such as age, education, occupation, marital status and weekly working hours. Due to its size and mixed variable types, the dataset is suitable for examining how federated binning and hashing influence the performance of the Categorical Naive Bayes model.

The Breast Cancer Wisconsin Diagnostic dataset represents a medical diagnostic classification task. The aim is to classify samples as benign or malignant tumors. The dataset is relatively small, but contains purely numerical morphological features, such as geometric and textural indicators related to cell nuclei (Street et al., 1993). It is therefore particularly suitable for examining how the discretization of continuous variables affects the performance of the Categorical Naive Bayes model.

The Credit Score Classification dataset is a banking and financial multi-class classification task. The goal is to predict customer creditworthiness based on financial, behavioral and demographic variables (Paris, 2022). The target variable contains three categories: Good, Standard and Poor. The larger sample size, mixed variable types and multi-class target variable of this dataset make it suitable for evaluating the proposed federated Naive Bayes-based workflow.

The Forest Cover Type dataset is a multi-class classification task based on geographical and cartographic variables. The aim is to predict the dominant forest cover type based on the characteristics of 30×30 m area units (Blackard, 1998). The dataset is large, contains seven classes, and shows substantial class imbalance. This is particularly useful for justifying balanced-accuracy-based evaluation, since simple accuracy is less informative about performance on minority classes in such settings.

The main characteristics of the datasets are summarized in Table 3. In the class-distribution column, the majority-class proportion or a short characterization of the class structure is reported. The final values are checked using the unified EDA pipeline, especially when records or features are removed during preprocessing.

5.2. Data splitting and client partitioning

For every benchmark, the complete dataset is split into training, validation and test subsets. The training and validation parts are used for model selection and hyperparameter tuning, while the test set is used exclusively for final performance evaluation. The split is stratified by class in order to preserve class proportions as far as possible in the different subsets.

To simulate the federated environment, the training data are divided into K disjoint client subsets:

$$\mathcal{D}_{\text{train}} = \bigsqcup_{k=1}^{K} \mathcal{D}_k. \tag{33}$$

Both IID and non-IID client partitioning are examined in the experiments. In the IID setting, samples are assigned to clients using stratified random partitioning. In the non-IID setting, Dirichlet-based partitioning is used, which makes client-wise class proportions different.

During Dirichlet partitioning, a client proportion vector is sampled for every class c:

$$\boldsymbol{\pi}_c = (\pi_{c,1}, \pi_{c,2}, \dots, \pi_{c,K}) \sim \text{Dirichlet}(\alpha_{\text{Dir}}, \alpha_{\text{Dir}}, \dots, \alpha_{\text{Dir}}), \tag{34}$$

where $\pi_{c,k}$ specifies the proportion of samples from class c assigned to client k. The samples belonging to class c are then distributed among clients according to the vector $\boldsymbol{\pi}_c$.

The parameter α_{Dir} controls the heterogeneity of client distributions. Larger α_{Dir} values yield proportions closer to the uniform distribution, so the class compositions of clients become more similar. Smaller α_{Dir} values produce sparser and more extreme proportion vectors, meaning that samples of certain classes are more likely to concentrate at only a few clients. This results in a stronger non-IID setting.

Therefore, several α_{Dir} values are used in the non-IID experiments. Larger values represent milder, while smaller values represent stronger client heterogeneity. In the experimental setup, for example, $\alpha_{\text{Dir}} = 0.5$ corresponds to moderately heterogeneous partitioning, while $\alpha_{\text{Dir}} = 0.1$ corresponds to strongly heterogeneous partitioning.

The randomized partitions are repeated with multiple seeds to ensure that the results do not reflect the characteristics of a single

Table 3
Main characteristics of the datasets used in the experimental evaluation.

Dataset	Samples	Features	Classes	Variables	Class distribution
Adult	32,561	14	2	6 num./8 cat.	≤50K: 75.9%; >50K: 24.1%
Breast	569	30	2	30 num./0 cat.	B: 62.7%; M: 37.3%
Credit	100,000	23	3	17 num./6 cat.	Standard: 53.2%; Poor: 29.0%; Good: 17.8%
Cover	581,012	54	7	54 num./0 cat.	7 classes; two majority classes: 85.3%; rarest: 0.5%

Table 4
Compared Naive Bayes-based methods.

Method	Short description
Centralized-CNB-MDLP	Centralized CNB with supervised MDLP discretization.
Centralized-CNB-FLPrep	Centralized CNB with the same preprocessing as FLNB.
Local-only CNB	Separate local CNB trained at each client, without server-side aggregation.
FLNB-Count	Count-based reference aggregation. It uses detailed local count statistics and is therefore not the proposed privacy-oriented branch.
FLNB-DirSign	Proposed direction-based aggregation that updates global PMFs from local correction directions.

client partition. Every compared method uses the same train-validation-test split and the same client partitions so that the comparison reflects methodological differences rather than variance arising from different data splits.

5.3. Compared Naive Bayes-based methods

The proposed FLNB procedure is evaluated using several Naive Bayes-based baselines and variants. The purpose of selecting these baselines is to measure separately the effects of centralized training, purely local training, federated count-based aggregation and DirSign-based aggregation (see Table 4).

The Centralized-CNB-MDLP and Centralized-CNB-FLPrep models serve as centralized references. The Local-only CNB baseline shows what performance can be achieved without aggregation, using only client-side local data. FLNB-Count is a performance reference that shows what result can be obtained when the server is allowed to use detailed local frequency information. This variant is used not as a privacy-oriented solution, but as a control comparison. The FLNB-DirSign variant examines the performance and stability trade-off of sign- and direction-based PMF updates that reduce the level of detail of directly transmitted local distributional information.

5.4. Hyperparameter tuning and sensitivity analysis

Validation-based search and sensitivity analysis are performed for the main hyperparameters. The purpose of validation-based model selection is to ensure that final test results do not depend on a single manually selected parameter configuration. The hyperparameters examined are reported in Table B.10.

A detailed summary of the investigated hyperparameters and their roles is provided in Appendix B. In the main text, smoothing, DirSign learning rate, binning complexity and hash-space size receive separate interpretation because these most directly influenced balanced-accuracy values.

Model selection is performed on the validation set. Balanced accuracy is used as the primary selection metric because, as the average of class-wise recall values, it is less dominated by the majority class. This is especially important in settings where the class distribution is imbalanced or locally distorted by non-IID client partitioning.

After selecting the best validation configuration, the final model is refitted using the training and validation data and then evaluated on the test set. In the sensitivity analysis, particular attention is given to the effects of τ_{unique}, N_{hash}, η, ϵ, $T_{\max}$, and the Dirichlet partitioning parameter α_{Dir}. These parameters directly influence feature-role identification, hashing, DirSign convergence and client heterogeneity.

5.5. Component-level analyses

The purpose of the component-level analyses is to examine how the main preprocessing, hashing, aggregation and partitioning choices affect final performance and stability.

The first analysis examines the role of federated binning. In this variant, federated binning is replaced by a simpler or non-iterative setting, and the impact of client-side score-based bin width selection on performance is evaluated.

The second analysis examines the role of deterministic hashing. The purpose is to analyze how the size of the hash space affects the category space, possible collisions and model performance.

The third analysis focuses on the aggregation rule. Here, the performance of direct count-based aggregation and DirSign PMF aggregation is compared. This analysis directly shows the trade-off introduced by sign-based updating relative to the statistically more direct count aggregation.

The fourth analysis examines the effect of the client distribution. Methods are evaluated under IID and different strengths of non-IID partitions to show which components are most sensitive to local distortions of the class distribution.

5.6. Evaluation protocol

Model performance is compared primarily using balanced accuracy. This metric is particularly justified in the present study because the datasets include several imbalanced or multi-class tasks, and non-IID client partitioning can further distort local class distributions. Accuracy, macro-F1 and weighted-F1 values are also computed as supplementary metrics, but model selection and the main comparison are based on balanced accuracy.

The validation set is used only for hyperparameter tuning and model selection. The final method comparison is based on results measured on the test set. Every method uses the same train-validation-test split, the same client partitions and the same random seeds, so that the comparison reflects methodological differences rather than variance caused by the data split.

In the main result tables, accuracy, macro-F1 and weighted-F1 values are also reported in addition to balanced accuracy. The latter metrics are not primary model-selection criteria; they support a more detailed interpretation of predictive behavior.

5.7. Statistical comparison

Balanced-accuracy values obtained over multiple datasets and several random seeds are used to evaluate differences between methods. The Friedman test is applied for the joint comparison of multiple methods. If the Friedman test indicates a significant difference, a post-hoc

Table 5
Performance of Naive Bayes-based methods under IID client partitions. Detailed F1 metrics are reported in the supplementary result tables.

Dataset	Method	Balanced accuracy	Accuracy
Adult	Centralized-CNB-MDLP	0.823 ± 0.010	0.839 ± 0.004
Adult	Centralized-CNB-FLPrep	0.817 ± 0.010	0.832 ± 0.005
Adult	Local-only CNB	0.807 ± 0.009	0.831 ± 0.004
Adult	FLNB-Count	0.817 ± 0.010	0.832 ± 0.005
Adult	FLNB-DirSign	0.775 ± 0.010	0.837 ± 0.004
Breast	Centralized-CNB-MDLP	0.936 ± 0.014	0.942 ± 0.013
Breast	Centralized-CNB-FLPrep	0.942 ± 0.003	0.947 ± 0.009
Breast	Local-only CNB	0.902 ± 0.012	0.920 ± 0.008
Breast	FLNB-Count	0.942 ± 0.003	0.947 ± 0.009
Breast	FLNB-DirSign	0.940 ± 0.012	0.947 ± 0.009
Credit	Centralized-CNB-MDLP	0.721 ± 0.002	0.677 ± 0.001
Credit	Centralized-CNB-FLPrep	0.700 ± 0.003	0.650 ± 0.001
Credit	Local-only CNB	0.685 ± 0.003	0.642 ± 0.001
Credit	FLNB-Count	0.700 ± 0.003	0.650 ± 0.001
Credit	FLNB-DirSign	0.699 ± 0.003	0.652 ± 0.000
Cover	Centralized-CNB-MDLP	0.494 ± 0.003	0.661 ± 0.001
Cover	Centralized-CNB-FLPrep	0.636 ± 0.005	0.684 ± 0.002
Cover	Local-only CNB	0.600 ± 0.005	0.687 ± 0.002
Cover	FLNB-Count	0.636 ± 0.005	0.684 ± 0.002
Cover	FLNB-DirSign	0.628 ± 0.005	0.684 ± 0.002

analysis is performed to examine rank differences between methods. For the most important pairwise comparisons, especially the comparison between FLNB-DirSign and FLNB-Count, the Wilcoxon signed-rank test is used.

The role of the statistical tests is supplementary: the main conclusions are not drawn solely from p-values, but are interpreted based on balanced accuracy, macro-F1, the heterogeneity of client partitions and the behavior of the aggregation mechanisms.

6. Results and interpretation

This section presents the experimental results of the proposed FLNB procedure. The interpretation is organized around five main questions. First, the cleaned characteristics of the datasets are summarized. Second, the model comparison obtained under IID client partitions is presented. Third, the difference between the aggregation reference and DirSign aggregation is analyzed separately. Fourth, the effects of the most important preprocessing and model parameters are examined. Finally, the effect of non-IID client partitioning is evaluated.

6.1. Summary characteristics of the datasets

The main characteristics of the datasets used in the experiments were summarized earlier in Table 3. After cleaning, the Breast Cancer Wisconsin dataset contains 30 predictive features, while the Credit Score Classification dataset retains 23 features after handling unrealistic numerical values and incorrect formats. When interpreting the results, it is particularly important to distinguish binary and multi-class tasks, mixed numerical-categorical feature structures, and the strong class imbalance of Forest Cover Type.

6.2. Overall performance under IID client partitions

The main method comparison is first performed under IID client partitions. This setting provides a controlled reference for assessing how different model variants behave when client-side class distributions do not differ in an extreme way. The comparison includes centralized reference models, local client models and two federated aggregation variants.

Table 5 summarizes the test results obtained in the IID setting. The Centralized-CNB-FLPrep model uses the same preprocessing logic as the FLNB procedure, but with centralized training. This reference separates the effect of FL-compatible preprocessing from the effect of federated aggregation. FLNB-Count is the reference based on aggregating local frequency tables, while FLNB-DirSign is the direction-based aggregation variant.

Table 6
Comparison of DirSign aggregation and the count-based reference in the IID setting.

Dataset	Method	Balanced accuracy	Accuracy
Adult	FLNB-Count reference	0.817 ± 0.010	0.832 ± 0.005
Adult	FLNB-DirSign	0.775 ± 0.010	0.837 ± 0.004
Breast	FLNB-Count reference	0.942 ± 0.003	0.947 ± 0.009
Breast	FLNB-DirSign	0.940 ± 0.012	0.947 ± 0.009
Credit	FLNB-Count reference	0.700 ± 0.003	0.650 ± 0.001
Credit	FLNB-DirSign	0.699 ± 0.003	0.652 ± 0.000
Cover	FLNB-Count reference	0.636 ± 0.005	0.684 ± 0.002
Cover	FLNB-DirSign	0.628 ± 0.005	0.684 ± 0.002

The FLNB-Count results serve as a control reference. The fact that this variant practically reproduces the performance of the Centralized-CNB-FLPrep model indicates that preprocessing and the federated experimental partitioning are consistent. However, this variant is not regarded as the proposed privacy-conscious aggregation approach, because it uses detailed local count statistics.

Under IID settings, FLNB-DirSign approaches this statistical reference on several datasets, particularly on Breast Cancer and Credit Score. Larger balanced-accuracy losses appear for Adult Income and Forest Cover Type, suggesting that the direction-based update may be more sensitive to data structure, client partitioning and feature representation. The main results are illustrated in Fig. 2.

6.3. Comparison of aggregation variants

In this subsection, FLNB-DirSign aggregation is compared with a reference that uses full local frequency information. FLNB-Count is a statistically direct comparison baseline because it constructs global class priors and class-conditional distributions by aggregating local class and feature-category frequencies. However, it requires detailed local distributional information and is therefore not the proposed exposure-reducing aggregation mechanism. In contrast, FLNB-DirSign aggregates the direction of the difference between local and global class-conditional distributions. The approach is related to sign-based federated optimization and communication-reducing aggregation methods (Bernstein et al., 2018; Guo et al., 2023).

Table 6 compares the performance of the proposed DirSign aggregation and the count-based reference.

The smallest difference between FLNB-DirSign and the reference is observed on the Credit Score dataset, where the two aggregation variants are practically identical in terms of balanced accuracy. The difference is also small on the Breast Cancer dataset. For Adult Income, a moderate balanced-accuracy loss appears, while for Forest Cover Type the loss is larger, although accuracy remains close to the reference in several cases. This suggests that the direction-based update mainly affects the balance of class-wise recall, which is directly reflected in the balanced-accuracy metric (see Fig. 3).

6.4. Hyperparameter sensitivity

The purpose of the following analyses is to reveal how sensitive the proposed method is to the most important preprocessing and aggregation hyperparameters. The detailed curves are shown in Figs. 4–11, while Table 7 summarizes the best one-factor screening results. This analysis is not a full multidimensional hyperparameter optimization, but a targeted sensitivity analysis.

The screening results show that DirSign aggregation can approach the aggregation reference in several cases when properly parameterized. For Adult Income, the best DirSign setting reaches a balanced

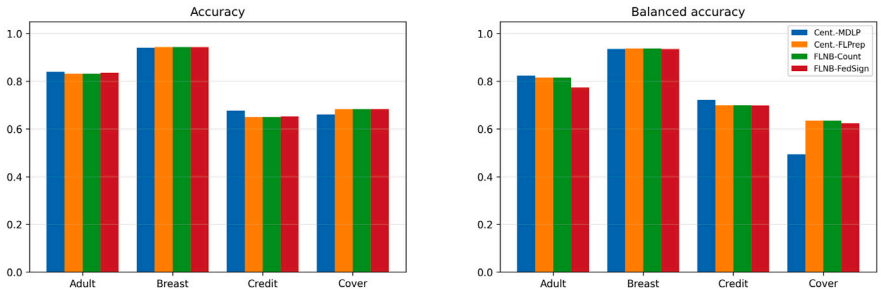

Fig. 2. Comparison of centralized and federated Naive Bayes-based methods based on accuracy and balanced accuracy under IID client partitions.

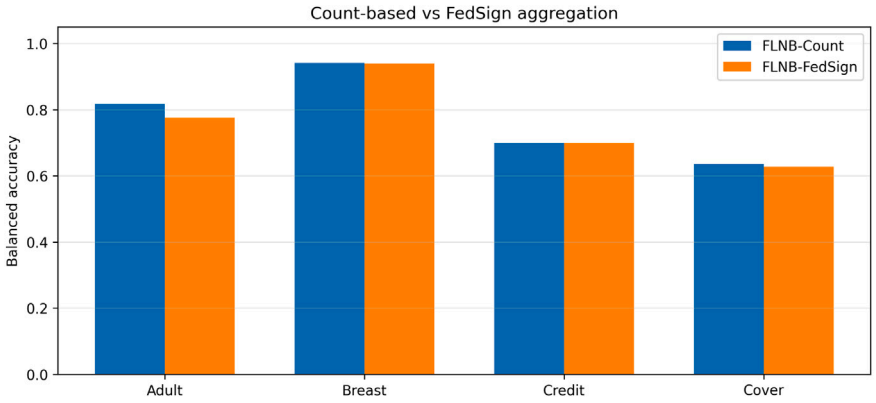

Fig. 3. Balanced-accuracy comparison of DirSign aggregation with the count-based reference under IID client partitions.

Table 7
Best one-factor screening results for the federated aggregation variants.

Dataset	Method	Best investigated setting	Balanced accuracy	Accuracy
Adult	FLNB-Count	$\tau_{\text{unique}} = 100$	0.821	0.840
Adult	FLNB-DirSign	$T_{\max} = 100$	0.809	0.830
Breast	FLNB-Count	$R_{\text{bin}} = 1$	0.950	0.956
Breast	FLNB-DirSign	$R_{\text{bin}} = 1$	0.957	0.965
Credit	FLNB-Count	$N_{\text{hash}} = 10\,000$	0.715	0.670
Credit	FLNB-DirSign	$N_{\text{hash}} = 10\,000$	0.709	0.665
Cover	FLNB-Count	Binning metric = entropy	0.635	0.683
Cover	FLNB-DirSign	$B_{\max} = 500$	0.626	0.683

accuracy of 0.809; for Breast Cancer, 0.957; for Credit Score, 0.709; and for Forest Cover Type, 0.626. This confirms that direction-based aggregation can be competitive, but is sensitive to update dynamics, category-space size and the amount of smoothing.

6.4.1. Smoothing parameter

In the Categorical Naive Bayes model, the smoothing parameter α controls the handling of zero or rare category-class combinations. With small α, the model remains more sensitive to locally observed frequencies, whereas with larger α, the class-conditional distributions are smoothed more strongly. Smoothing is particularly important in the case of a large category space, rare bins and non-IID client distributions (Manning et al., 2008; Murphy, 2012).

Fig. 4 shows the effect of the parameter α. The results indicate that excessive smoothing causes performance degradation on several datasets. For example, on Forest Cover Type, FLNB-Count achieves a balanced accuracy of 0.633 at $\alpha = 0.01$, while it decreases to 0.344 at $\alpha = 2.0$. On the same dataset, FLNB-DirSign decreases from 0.575 to 0.195. This indicates that excessive smoothing can wash out information associated with rare classes and fine category values.

6.4.2. DirSign learning rate

One of the most important parameters of the DirSign update is the learning rate η. Fig. 5 shows that too small an η leads to slow and under-corrected updates, while too large an η may cause unstable or excessive updates. For example, on Forest Cover Type, balanced accuracy is 0.255 at $\eta = 0.0001$, 0.623 at $\eta = 0.01$, 0.606 at $\eta = 0.05$, and 0.599 at $\eta = 0.1$. For Credit Score, the best range appears around $\eta = 0.005$–0.01.

6.4.3. Binning and hashing parameters

The performance of the Categorical Naive Bayes model strongly depends on how much class-relevant information is preserved by the discretization of continuous variables. Too coarse binning may lead to information loss, while too fine binning may result in rare categories and unstable class-conditional estimates. The effects of binning rounds, the unique-value threshold and the maximum number of bins are summarized in Figs. 6, 7 and 8.

In deterministic hashing, the parameter N_{hash} controls the size of the hash space. A larger N_{hash} reduces the probability of collisions but increases the size of the category space. A smaller N_{hash} provides a more compact representation but increases the chance that different category values are mapped to the same index. Based on Fig. 9, a larger hash space is favorable for Credit Score Classification, where FLNB-Count reaches a balanced accuracy of 0.715 and FLNB-DirSign reaches 0.710 at $N_{\text{hash}} = 10\,000$.

6.4.4. Aggregation rounds and convergence threshold

For the DirSign update, the maximum number of aggregation rounds and the convergence threshold directly affect how far the global class-conditional distributions can stabilize. For Adult Income, increasing the maximum number of aggregation rounds improved the DirSign result in the examined screening, with the best one-factor setting reported in Table 7. Figs. 10 and 11 show the effects of these parameters.

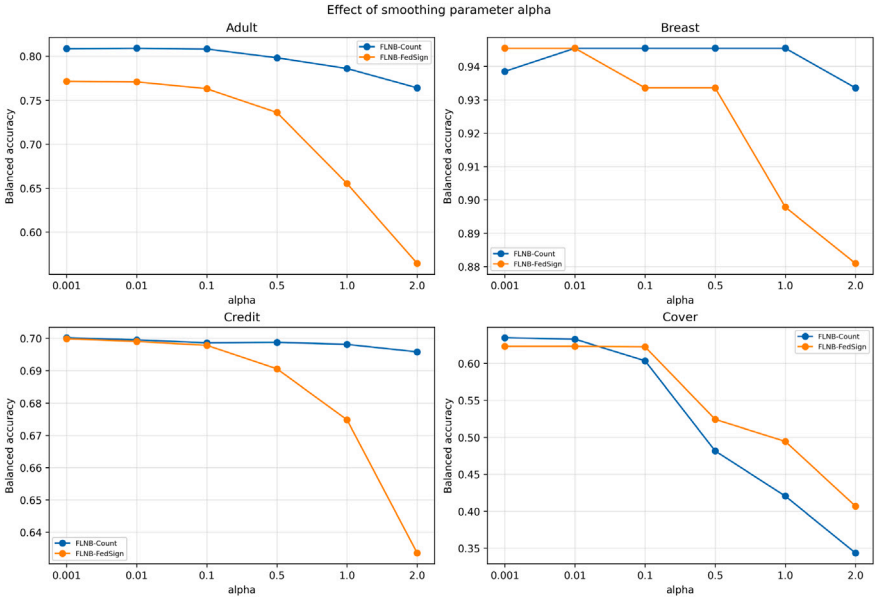

Fig. 4. Effect of the smoothing parameter α on balanced-accuracy values.

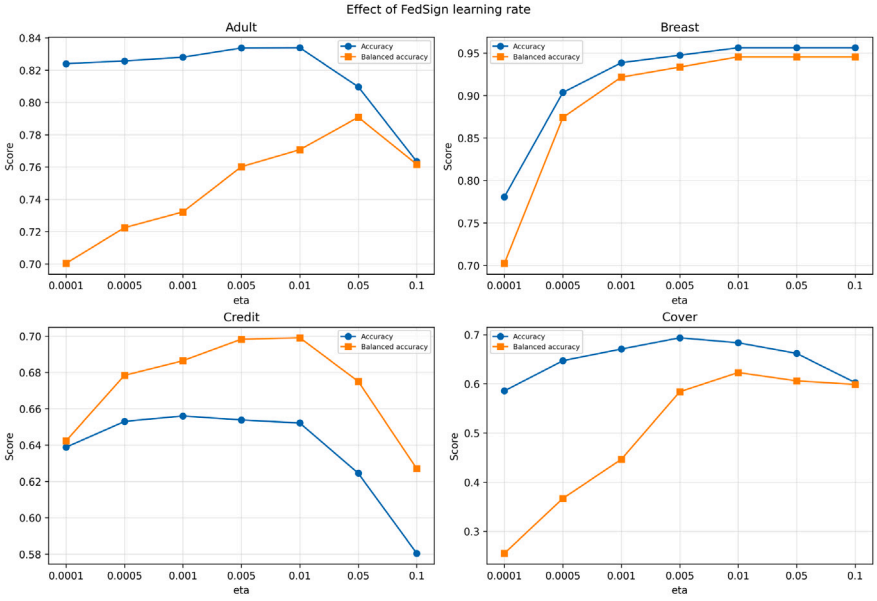

Fig. 5. Effect of the learning-rate parameter of the DirSign update.

6.5. Robustness under non-IID client partitions

One central challenge in Federated Learning is that client-side data are often not identically distributed. Non-IID distributions may cause client drift, biased local estimates and unstable aggregation (Li et al., 2020; McMahan et al., 2017). To examine this issue, Dirichlet-based client partitioning was applied, where smaller α_{Dir} values result in stronger class-distribution heterogeneity.

Table 8 summarizes the FLNB-Count and FLNB-DirSign results obtained under IID and Dirichlet-based partitions.

The non-IID results indicate that the reference using full local count information remains stable on all datasets. This is expected because this variant directly sums local frequencies and is therefore less sensitive to which clients contain the samples of each class. In contrast, the behavior of DirSign aggregation is dataset-dependent. For Breast Cancer, performance remains close to the IID results even in a non-IID environment. For Credit Score, only a small degradation is observed

Table 8
Effect of client partitioning on the balanced-accuracy values of the count-based reference and FLNB-DirSign.

Dataset	Partition	FLNB-Count ref.	FLNB-DirSign	Difference
Adult	IID	0.817 ± 0.010	0.775 ± 0.010	0.042
Adult	Dir. 0.5	0.816 ± 0.009	0.773 ± 0.007	0.043
Adult	Dir. 0.1	0.814 ± 0.012	0.772 ± 0.010	0.042
Breast	IID	0.942 ± 0.003	0.940 ± 0.012	0.002
Breast	Dir. 0.5	0.929 ± 0.019	0.928 ± 0.018	0.001
Breast	Dir. 0.1	0.942 ± 0.003	0.940 ± 0.012	0.002
Credit	IID	0.700 ± 0.003	0.699 ± 0.003	0.001
Credit	Dir. 0.5	0.700 ± 0.003	0.699 ± 0.003	0.001
Credit	Dir. 0.1	0.700 ± 0.002	0.699 ± 0.002	0.001
Cover	IID	0.636 ± 0.005	0.628 ± 0.005	0.008
Cover	Dir. 0.5	0.634 ± 0.009	0.622 ± 0.009	0.012
Cover	Dir. 0.1	0.633 ± 0.004	0.621 ± 0.004	0.012

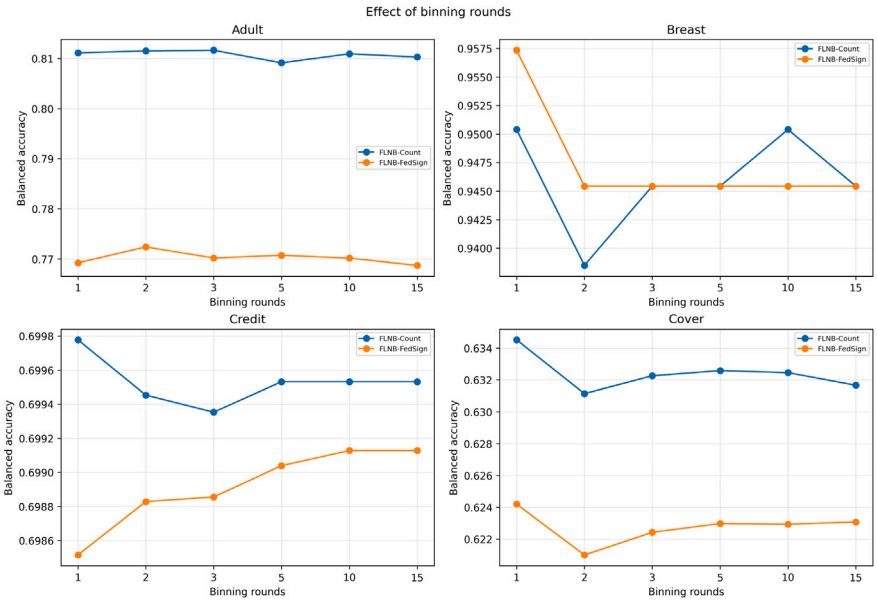

Fig. 6. Effect of the binning rounds parameter on balanced-accuracy values.

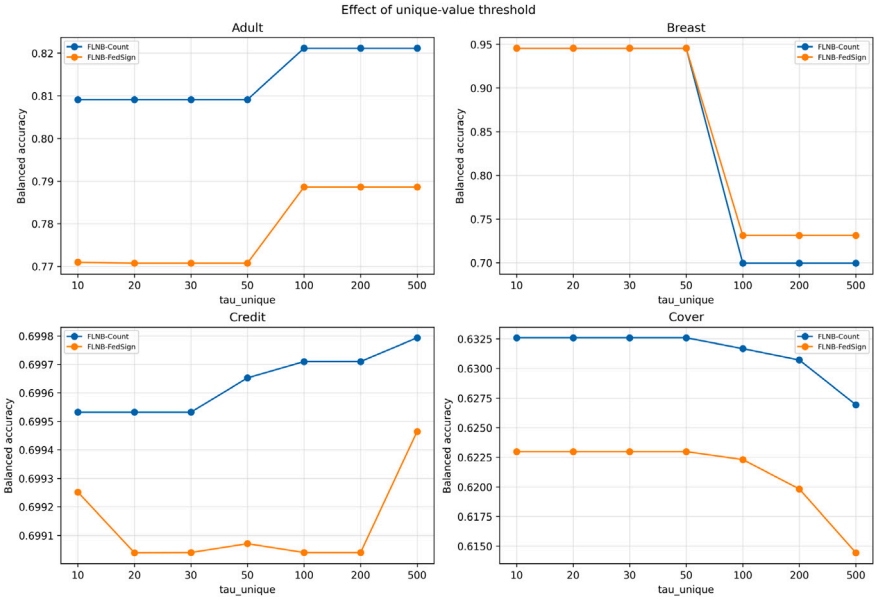

Fig. 7. Effect of the unique-value threshold τ_{unique} on balanced-accuracy values.

under the milder Dirichlet 0.5 setting, and the DirSign results remain close to the IID setting under both Dirichlet 0.5 and Dirichlet 0.1 partitioning.

For the Forest Cover Type dataset, the latest results show that DirSign aggregation does not collapse in the non-IID environment. Under IID partitioning, the balanced accuracy of FLNB-DirSign is 0.628, while it is 0.622 under Dirichlet 0.5 and 0.621 under Dirichlet 0.1. This indicates that class-row-wise direction-based aggregation remains stable in the examined non-IID settings, although it still shows a small performance loss compared with the reference using full local count information (see Fig. 12).

6.6. Statistical comparison

A statistical comparison was also performed based on the results obtained over several datasets and multiple random seeds. The Friedman test indicated a significant difference between the ranks of the examined methods, with $p = 8.68 \cdot 10^{-17}$, $n = 36$. Based on average ranks, Centralized-CNB-MDLP achieved the best rank with a value of 2.236, followed by Centralized-CNB-FLPrep and FLNB-Count with identical average ranks of 2.278. The average rank of FLNB-DirSign was 3.375, while that of Local-only CNB was 4.833.

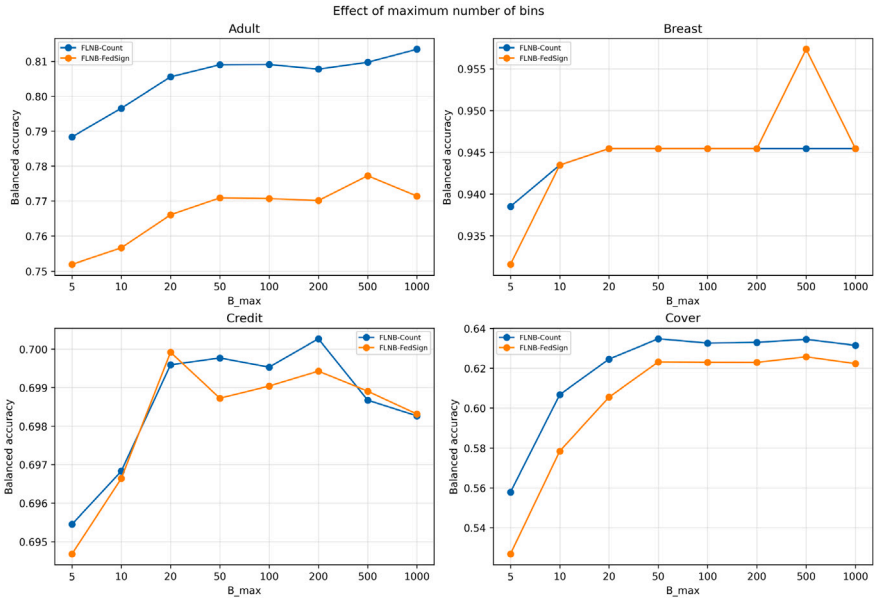

Fig. 8. Effect of the maximum allowed number of bins on balanced-accuracy values.

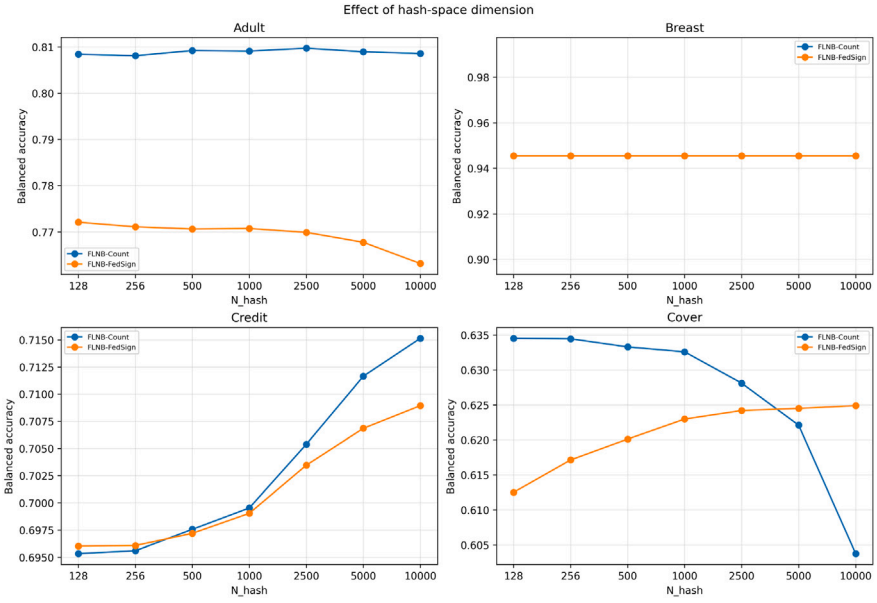

Fig. 9. Effect of hash-space size on balanced-accuracy values.

The Nemenyi critical-difference comparison indicates no significant difference between FLNB-Count and Centralized-CNB-FLPrep. This is consistent with the fact that the two methods provide practically the same predictive performance under identical preprocessing. The rank difference between FLNB-DirSign and the count-based reference was significant in the full comparison including all partitions. This difference should be interpreted together with the practical effect size, because under IID settings the difference is small on several datasets, and in the latest non-IID results no degenerate collapse appears for Forest Cover Type either (see Table 9).

The statistical tests show that the reference using full local frequency information obtains a more favorable rank overall. This result is expected because this reference uses more information than DirSign. From a practical perspective, therefore, the important point is not the rank difference alone, but the fact that under IID settings DirSign aggregation approaches this reference with a small performance loss

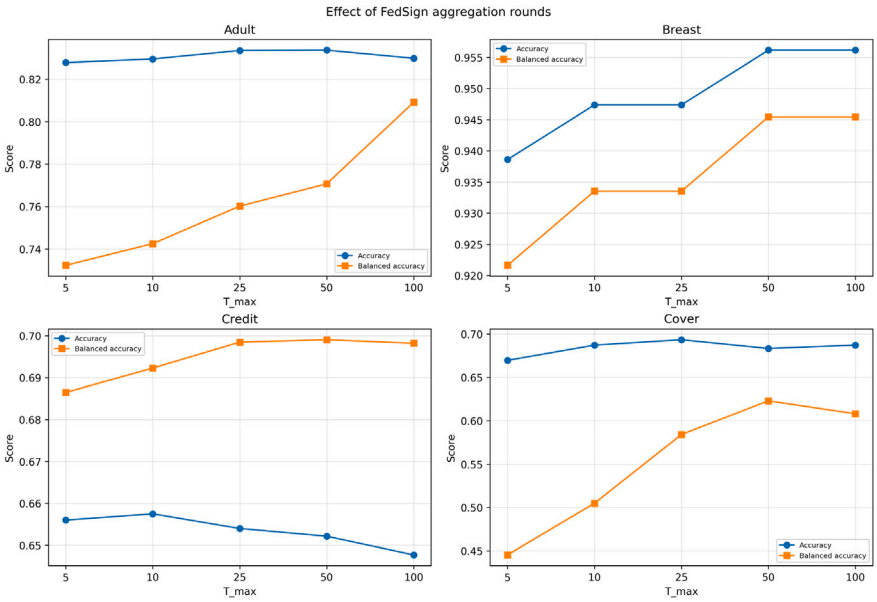

Fig. 10. Effect of the maximum number of DirSign aggregation rounds.

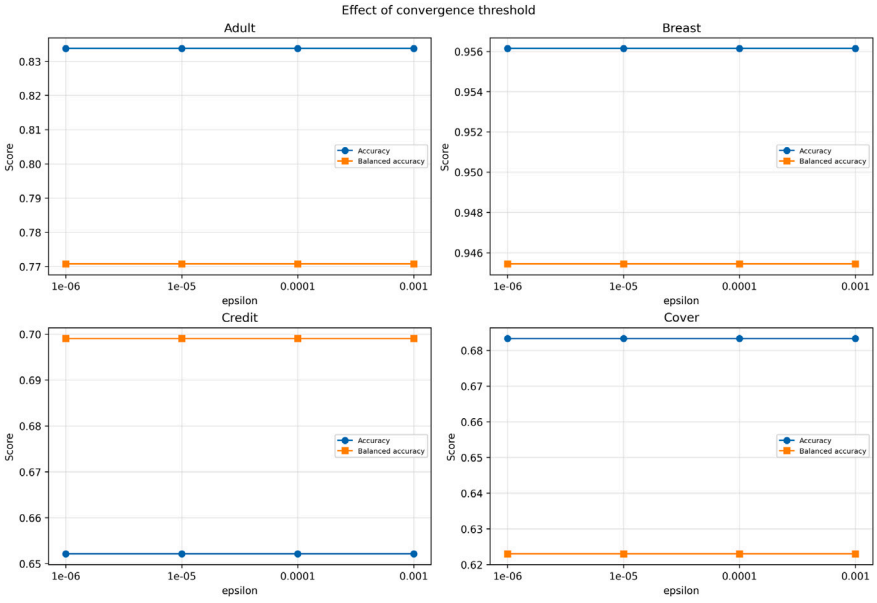

Fig. 11. Effect of the DirSign aggregation convergence threshold.

on several datasets, while stronger robustness improvements are still needed for highly heterogeneous multi-class tasks.

6.7. Summary of results

Three main conclusions can be drawn from the results. First, under IID conditions, DirSign aggregation approaches the non-privacy-oriented federated reference that uses full local frequency information on several benchmarks. For Adult Income, Breast Cancer Wisconsin and Credit Score Classification, the balanced-accuracy difference is small or moderate, while for Forest Cover Type the difference is larger, but the method does not collapse in the IID environment.

Table 9
Statistical comparison of the methods based on balanced accuracy.

Test	Comparison	Statistic	p-value
Friedman	All methods	81.427	$8.68 \cdot 10^{-17}$
Wilcoxon	FLNB-DirSign vs. FLNB-Count	40.000	$1.72 \cdot 10^{-5}$
Wilcoxon	FLNB-DirSign vs. Centralized-CNB-FLPrep	40.000	$1.72 \cdot 10^{-5}$
Wilcoxon	FLNB-DirSign vs. Local-only CNB	25.000	$1.30 \cdot 10^{-6}$

Second, DirSign aggregation is more sensitive to hyperparameters, especially the choice of learning rate, smoothing and representation parameters. Therefore, validation-based parameter selection is necessary in practical applications.

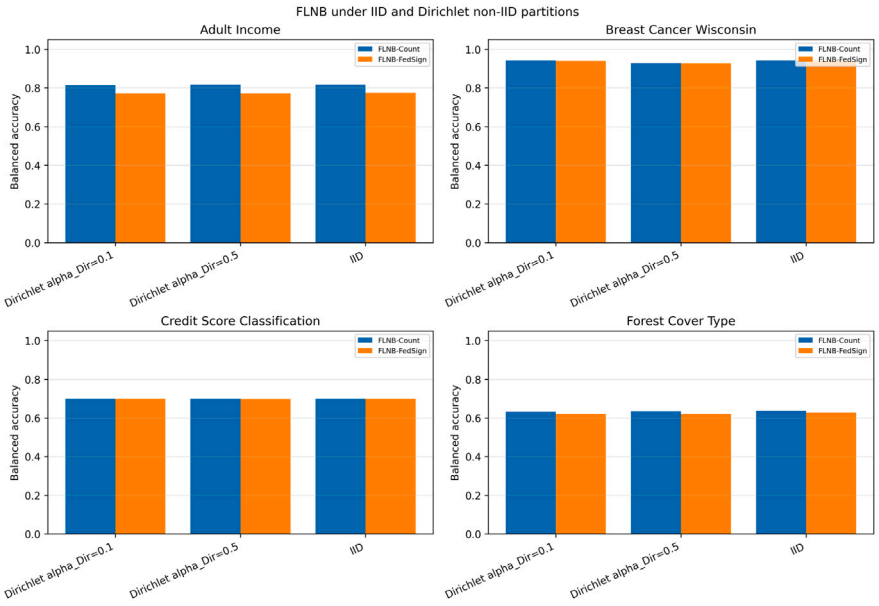

Fig. 12. Effect of Dirichlet-based non-IID client partitioning on balanced-accuracy values of the federated models.

Third, based on the non-IID experiments, DirSign aggregation remained stable in the examined settings, including the Forest Cover Type dataset. Overall, the stability of the FLNB procedure is supported by federated preprocessing, cross-client representational consistency and the aggregatable statistical structure of Categorical Naive Bayes, while DirSign aggregation is the proposed lower-information-content aggregation alternative that requires appropriate parameterization.

7. Conclusions and future directions

This study presented a Naive Bayes-based federated training procedure for structured classification tasks. The starting point of the work was that the parameters of Naive Bayes models can be estimated from local class and feature frequencies or class-conditional distributions; therefore, with suitable preprocessing and aggregation mechanisms, they can be naturally adapted to horizontal Federated Learning settings. The purpose of the proposed FLNB procedure was to construct an interpretable, communication-efficient and Naive Bayes-compatible federated training process in which raw records are not centralized.

The main contribution of the method is that it treats Naive Bayes training not merely as local model fitting, but as a complete federated preprocessing and aggregation process. Continuous variables were handled using federated binning, cross-client representational consistency of categorical and discretized variables was ensured by deterministic hashing, and the global Categorical Naive Bayes model was constructed from local statistics or direction-based PMF updates. This structure enables the model to learn from local client information without collecting raw samples at a central location.

Based on the experimental results, the FLNB procedure could be applied to several datasets with different sizes and class structures. The different data structures of Adult Income, Breast Cancer Wisconsin, Credit Score Classification and Forest Cover Type made it possible to examine how the method behaves in binary and multi-class, more balanced and strongly imbalanced classification tasks. The results showed that FLNB performance is not determined by a single component, but by the combined effects of discretization, hashing, smoothing and aggregation dynamics.

The objective of the FLNB-DirSign variant was to update global class-conditional distributions not by directly summing full local frequency tables, but based on correction directions between local and global PMFs. Under IID client partitions, this aggregation strategy worked on several datasets with only small or moderate performance loss relative to the reference using full local count information. This suggests that direction-based PMF updates may carry sufficient information to estimate the global Categorical Naive Bayes model for certain data structures, while not requiring direct aggregation of full local frequency tables.

The sensitivity analyses also showed that the performance of FLNB-DirSign strongly depends on the hyperparameters. The learning rate, smoothing parameter, binning complexity and hash-space size all influenced final predictive performance. This is particularly important for Categorical Naive Bayes because overly fine discretization may produce rare category-class combinations, while overly strong smoothing may wash out class-specific information. From the perspective of practical application, validation-based hyperparameter selection is therefore an essential part of the method.

The non-IID client partitioning results showed that FLNB-DirSign also remained stable under the examined Dirichlet-based heterogeneous settings. Client heterogeneity remains an important methodological challenge, but the present results indicate that the proposed class-row-wise DirSign aggregation did not show degenerate behavior on the Forest Cover Type dataset either. This is consistent with the general observation that client distribution heterogeneity must be explicitly examined and handled in Federated Learning (Li et al., 2020; McMahan et al., 2017).

The privacy interpretation of the method requires careful wording. The FLNB procedure reduces the need to centralize raw records, and deterministic hashing avoids the direct sharing of a common plaintext category dictionary. However, this does not constitute a formal privacy guarantee. The present study does not claim differential privacy, secure multiparty computation or cryptographic protection. The privacy-related contribution of the method is rather exposure-reduction: reducing the amount and semantic interpretability of directly transmitted local information. Formal privacy guarantees would

require the future integration of secure aggregation, differential privacy or other cryptographic protocols (Bonawitz et al., 2017; Dwork & Roth, 2014).

A further limitation of the study is that it focuses on structured classification data. Image or raw-text benchmarks do not fit directly into the current Categorical Naive Bayes-based workflow because they typically require representation learning or a separate feature-extraction step. Similarly, the present procedure is not directly applicable to regression tasks because Naive Bayes is based on a classification decision rule and a discrete target variable. These extensions would require separate methodological development.

Future work can proceed in several directions. First, adaptive hyperparameter-selection strategies should be developed to select binning, hashing, smoothing and DirSign update parameters based on client-side or server-side validation information. Second, the robustness of aggregation could be further improved through class-support-based weighting, adaptive learning rates or confidence-thresholding mechanisms, especially in non-IID multi-class tasks with rare classes. Third, the privacy-related components of the method could be made more formal by integrating differential privacy or secure aggregation. Fourth, the FLNB procedure could be extended to other Naive Bayes variants, such as Gaussian, Complement or Bernoulli Naive Bayes, provided that the parameters of the given model can be estimated in a federated way.

Overall, the results show that a federated preprocessing and aggregation process designed for Naive Bayes models is a viable approach for structured classification tasks. The key lesson of the method is that the simplicity, additive statistical structure and interpretability of Naive Bayes can be effectively exploited in a federated environment, provided that cross-client representational consistency, discretization and aggregation dynamics are handled appropriately.

CRediT authorship contribution statement

Zsolt András Romhányi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing. **Alex Kummer:** Supervision, Methodology, Validation, Writing – review & editing.

Code availability

The code supporting this study is publicly available at https://github.com/DataCentricSE/FederatedLearning_Bayesian_Classification.git.

Acknowledgments

Project no. 2024-1.1.1-KKV_FÓKUSZ-2024-00033 is co-funded by the NRDI Office and the National Innovation Agency with the support of the Hungarian State. This project was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences (B0/00439/25/6). All authors have read and approved the submitted version of the manuscript.

Appendix A. The complete FLNB algorithms

Algorithm 2 in the main text provides a compact summary of the complete procedure. The following three algorithms present the main steps of the FLNB pipeline in a more detailed, implementation-oriented form. Part I covers client-side profiling and the selection of features to be discretized, Part II covers federated binning, hashing and the computation of local Naive Bayes statistics, and Part III covers global aggregation and the construction of the final Categorical Naive Bayes model.

Algorithm 3: FLNB - Part I: initialization, feature selection and federated binning optimization

PHASE 1: Initialization and data distribution
Server:
Load configuration parameters: K, τ_{unique}, R_{bin}, $B_{\max}$, N_{hash}, α, η, $T_{\max}$, and ϵ.
Fix the common feature list $\mathcal{X} = \{x_1, \ldots, x_d\}$ and the class-label set $\mathcal{Y}$.
In the experimental simulation, split the complete dataset into train-validation-test subsets and partition the training set into K disjoint client subsets: $\mathcal{D}_{\text{train}} = \bigcup_{k=1}^{K} \mathcal{D}_k$.
Remove identifier-like columns and initialize the clients with their local datasets $\mathcal{D}_k$.
PHASE 2: Federated feature profiling and feature selection
Server:
Send $\mathcal{X}$, $\mathcal{Y}$, and τ_{unique} to every client.
Each client k in parallel:
Separate local input features and the target variable.
Verify that the local data follow the common feature order.
Determine the local sample size m_k and the class supports $N_y^{(k)}$.
Initialize the local continuous-feature set $F_{\text{cont}}^{(k)} \leftarrow \emptyset$.
foreach *numerical feature* $x_j \in \mathcal{X}$ **do**
  Determine the number of local unique values: $u_j^{(k)} = |\text{Unique}_k(x_j)|$.
  if $u_j^{(k)} > \tau_{\text{unique}}$ **then**
    Add x_j to $F_{\text{cont}}^{(k)}$ and compute the local scale information $\rho_j^{(k)}$.
  else
    Mark x_j as a discrete numerical feature.
Mark non-numerical features as categorical features.
Send $F_{\text{cont}}^{(k)}$, the corresponding $\rho_j^{(k)}$ values, and the class supports $N_y^{(k)}$ to the server.
Do not send raw feature values, plaintext category values, or record-level data.
Server:
Receive the feature markings from all clients.
Determine the global set of features to be discretized: $F_{\text{bin}} = \bigcup_{k=1}^{K} F_{\text{cont}}^{(k)}$.
For every feature $x_j \in F_{\text{bin}}$, aggregate scale information into ρ_j^{glob}.
PHASE 3: Federated binning optimization
Round 0: scale information and initial candidate set
Server:
Based on ρ_j^{glob}, construct an initial candidate bin-width set for every $x_j \in F_{\text{bin}}$: $\mathcal{B}_j^{(1)} = \{b_{j,1}, \ldots, b_{j,M}\}$.
Send F_{bin} and the candidate bin-width sets to the clients.
Rounds 1 to R_{bin}: bin-width search
for $r = 1, \ldots, R_{\text{bin}}$ **do**
  Server: send the current candidate sets $\mathcal{B}_j^{(r)}$ to the clients.
  Each client k in parallel:
  foreach $x_j \in F_{\text{bin}}$ *and* $b \in \mathcal{B}_j^{(r)}$ **do**
    Locally form bin indices based on the current candidate b.
    Compute the selected binning score $s_j^{(k)}(b)$.
    Apply a penalized score for unstable, overly sparse, or $B_{\max}$-violating binning.
  Send the score values $\{s_j^{(k)}(b)\}$ to the server.
  Do not send raw values, local bin boundaries, or record-level bin indices.
  Server:
  Aggregate client-side scores: $s_j(b) = K^{-1} \sum_{k=1}^{K} s_j^{(k)}(b)$.
  Select the best candidate $b_j^{*,(r)}$ according to the optimization direction of the score.
  Refine the next search range $\mathcal{B}_j^{(r+1)}$ around the selected candidate.
Server:
Fix the final binning configuration $b_j^* = b_j^{*,(R_{\text{bin}})}$ for every $x_j \in F_{\text{bin}}$.
Send the final binning configuration to every client.
Continues in Algorithm 4.

Algorithm 4: FLNB - Part II: final preprocessing, hashing and local Naive Bayes statistics

PHASE 4: Final local preprocessing
Server:
Define feature roles: $F_{\text{binned}} = F_{\text{bin}}$ and $F_{\text{hash}} = \mathcal{X} \setminus F_{\text{binned}}$.
Send the final binning configuration, hash-space size N_{hash}, and feature roles to every client.
Each client k **in parallel:**
Initialize the preprocessed local dataset $\widehat{D}_k \leftarrow \emptyset$.
foreach *local sample* $(x_k^{(i)}, y_k^{(i)}) \in D_k$ **do**
  foreach *feature* $x_j \in \mathcal{X}$ **do**
    if $x_j \in F_{\text{binned}}$ **then**
      Form a bin index using the final binning configuration b_j^* and clip it to the valid range.
    else
      Apply deterministic hashing: $\hat{x}_{k,j}^{(i)} = \text{hash}(j, x_{k,j}^{(i)}) \bmod N_{\text{hash}}$.
  Add $(\hat{x}_k^{(i)}, y_k^{(i)})$ to $\widehat{D}_k$.
The client does not create a common plaintext category dictionary and does not send plaintext category values to the server.
PHASE 5: Computation of local Categorical Naive Bayes statistics
Each client k **in parallel:**
Determine local class supports $N_y^{(k)}$ for every class $y \in \mathcal{Y}$.
Determine feature-wise category counts C_j.
foreach *feature* $x_j \in \mathcal{X}$, *class* $y \in \mathcal{Y}$, *and category* $c \in \{0, \dots, C_j - 1\}$ **do**
  Compute the local feature-class count $N_{j,c,y}^{(k)}$.
  if $N_y^{(k)} > 0$ **then**
    Estimate the local PMF with smoothing: $P_k(x_j = c \mid y) = \frac{N_{j,c,y}^{(k)} + \alpha}{N_y^{(k)} + \alpha C_j}$.
  else
    Do not construct a DirSign update for the given class row.
Compute the local class priors $P_k(y)$.
Store the local PMFs for DirSign aggregation.
Local count tables may be used only for count-based reference evaluation.
PHASE 6: Server-side preparation for aggregation
Server:
Receive the local PMFs or update interface required for DirSign aggregation.
Receive the class supports $N_y^{(k)}$.
In the reference evaluation, also receive the local count statistics.
Determine the aggregated supports required for global class priors: $N_y = \sum_{k=1}^{K} N_y^{(k)}$.
Continues in Algorithm 5.

Algorithm 5: FLNB - Part III: reference aggregation, DirSign aggregation and global classifier

PHASE 7: Global class priors
Server:
Aggregate class supports: $N_y = \sum_{k=1}^{K} N_y^{(k)}$.
Compute global priors: $P_g(y) = \frac{N_y}{\sum_{y' \in \mathcal{Y}} N_{y'}}$.
PHASE 8A: Count-based reference aggregation
Server:
Use this branch only as a performance reference, not as the proposed exposure-reducing aggregation.
foreach *feature* x_j, *class* y, *and category* c **do**
  Sum local counts: $N_{j,c,y} = \sum_{k=1}^{K} N_{j,c,y}^{(k)}$.
  Estimate the reference global PMF: $P_g(x_j = c \mid y) = \frac{N_{j,c,y} + \alpha}{N_y + \alpha C_j}$.
The reference branch directly uses detailed local count information and is therefore less favorable from a privacy perspective.
PHASE 8B: DirSign aggregation
Server initialization:
For every feature x_j, class y, and category c, initialize $P_g^{(0)}(x_j = c \mid y) = \frac{1}{C_j}$.
Set the DirSign parameters η, $T_{\max}$, ϵ, and δ.
for $t = 0, \dots, T_{\max} - 1$ **do**
  Server: send the current $P_g^{(t)}$ PMFs to the clients.
  Each client k **in parallel:**
  foreach *feature* x_j *and class* y **do**
    if $N_y^{(k)} > 0$ **then**
      Compute the local-global PMF difference: $D_{j,\cdot,y}^{(k)} = P_k(x_j \mid y) - P_g^{(t)}(x_j \mid y)$.
      Form the normalized direction: $\Delta_{j,\cdot,y}^{(k)} = \frac{D_{j,\cdot,y}^{(k)}}{\|D_{j,\cdot,y}^{(k)}\|_2 + \delta}$.
    else
      Do not send an update for the given class row.
  Send the directions $\Delta_{j,\cdot,y}^{(k)}$ and supports $N_y^{(k)}$ to the server.
  Do not send local count tables for DirSign aggregation.
  Server: support-weighted direction aggregation:
  $\Delta_{j,\cdot,y} = \frac{\sum_{k:\, N_y^{(k)} > 0} N_y^{(k)} \Delta_{j,\cdot,y}^{(k)}}{\sum_{k:\, N_y^{(k)} > 0} N_y^{(k)}}$.
  Server: PMF update:
  $\widetilde{P}_g^{(t+1)}(x_j \mid y) = P_g^{(t)}(x_j \mid y) + \eta \Delta_{j,\cdot,y}$.
  Apply a lower bound with δ, and normalize every feature-class PMF row.
  Convergence check:
  $r^{(t)} = \frac{\sum_{j,y} \|P_g^{(t+1)}(x_j|y) - P_g^{(t)}(x_j|y)\|_2}{\sum_{j,y} \|P_g^{(t)}(x_j|y)\|_2 + \delta}$.
  if $r^{(t)} < \epsilon$ **then**
    Stop the DirSign iteration.
PHASE 9: Global Categorical Naive Bayes classifier
Server:
Build the global CNB model from the final $P_g(y)$ and $P_g(x_j = c \mid y)$ values.
Use the prediction rule: $F_g(x) = \arg\max_{y \in \mathcal{Y}} \left[\log P_g(y) + \sum_{j=1}^{d} \log P_g(x_j \mid y)\right]$.
Evaluate the global model on the central test set using the metrics defined in the experimental protocol.

Table B.10
Default values of the main experimental hyperparameters.

Hyperparameter	Default value or examined role
K	8 clients
Train-validation-test split	64%/16%/20%
τ_{unique}	30
R_{bin}	5
B_{max}	Adult/Credit/Cover: 100; Breast: 50
N_{hash}	1000
α	0.01
η	0.01
T_{max}	50
ϵ	10^{-6}
α_{Dir}	IID, 0.5 and 0.1 non-IID experiments

Table B.11
Supplementary IID model comparison with macro-F1 and weighted-F1 metrics.

Dataset	Method	Balanced acc.	Accuracy	Macro-F1	Weighted-F1
Adult	Centralized-CNB-MDLP	0.823 ± 0.010	0.839 ± 0.004	0.796 ± 0.007	0.845 ± 0.005
Adult	Centralized-CNB-FLPrep	0.817 ± 0.010	0.832 ± 0.005	0.789 ± 0.008	0.839 ± 0.006
Adult	Local-only CNB	0.807 ± 0.009	0.831 ± 0.004	0.784 ± 0.006	0.837 ± 0.004
Adult	FLNB-Count ref.	0.817 ± 0.010	0.832 ± 0.005	0.789 ± 0.008	0.839 ± 0.006
Adult	FLNB-DirSign	0.775 ± 0.010	0.837 ± 0.004	0.776 ± 0.008	0.837 ± 0.005
Breast	Centralized-CNB-MDLP	0.936 ± 0.014	0.942 ± 0.013	0.937 ± 0.014	0.941 ± 0.013
Breast	Centralized-CNB-FLPrep	0.942 ± 0.003	0.947 ± 0.009	0.943 ± 0.009	0.947 ± 0.008
Breast	Local-only CNB	0.902 ± 0.012	0.920 ± 0.008	0.911 ± 0.010	0.918 ± 0.009
Breast	FLNB-Count ref.	0.942 ± 0.003	0.947 ± 0.009	0.943 ± 0.009	0.947 ± 0.008
Breast	FLNB-DirSign	0.940 ± 0.012	0.947 ± 0.009	0.943 ± 0.010	0.947 ± 0.009
Credit	Centralized-CNB-MDLP	0.721 ± 0.002	0.677 ± 0.001	0.670 ± 0.001	0.682 ± 0.002
Credit	Centralized-CNB-FLPrep	0.700 ± 0.003	0.650 ± 0.001	0.644 ± 0.002	0.654 ± 0.001
Credit	Local-only CNB	0.685 ± 0.003	0.642 ± 0.001	0.636 ± 0.001	0.646 ± 0.001
Credit	FLNB-Count ref.	0.700 ± 0.003	0.650 ± 0.001	0.644 ± 0.002	0.654 ± 0.001
Credit	FLNB-DirSign	0.699 ± 0.003	0.652 ± 0.000	0.646 ± 0.001	0.657 ± 0.000
Cover	Centralized-CNB-MDLP	0.494 ± 0.003	0.661 ± 0.001	0.490 ± 0.004	0.661 ± 0.001
Cover	Centralized-CNB-FLPrep	0.636 ± 0.005	0.684 ± 0.002	0.576 ± 0.006	0.688 ± 0.002
Cover	Local-only CNB	0.600 ± 0.005	0.687 ± 0.002	0.574 ± 0.005	0.690 ± 0.002
Cover	FLNB-Count ref.	0.636 ± 0.005	0.684 ± 0.002	0.576 ± 0.006	0.688 ± 0.002
Cover	FLNB-DirSign	0.628 ± 0.005	0.684 ± 0.002	0.579 ± 0.006	0.688 ± 0.002

Table B.12
Best one-factor screening results with supplementary metrics.

Dataset	Method	Best investigated setting	Balanced acc.	Accuracy	Macro-F1	Weighted-F1
Adult	FLNB-Count ref.	$\tau_{\text{unique}} = 100$	0.821	0.840	0.797	0.845
Adult	FLNB-DirSign	$T_{\text{max}} = 100$	0.809	0.830	0.784	0.836
Breast	FLNB-Count ref.	$R_{\text{bin}} = 1$	0.950	0.956	0.953	0.956
Breast	FLNB-DirSign	$R_{\text{bin}} = 1$	0.957	0.965	0.962	0.965
Credit	FLNB-Count ref.	$N_{\text{hash}} = 10\,000$	0.715	0.670	0.663	0.675
Credit	FLNB-DirSign	$N_{\text{hash}} = 10\,000$	0.709	0.665	0.659	0.670
Cover	FLNB-Count ref.	Binning metric = entropy	0.635	0.683	0.574	0.687
Cover	FLNB-DirSign	$B_{\text{max}} = 500$	0.626	0.683	0.579	0.687

Table B.13
Main communication steps of the FLNB procedure and the type of information transmitted to the server.

Component	Information transmitted to the server	Information not transmitted
Feature profiling	List of features marked as continuous, low-resolution scale information, class support	Raw feature values, local records, plaintext category values
Federated binning	Local scores associated with candidate bin width values	Local raw values, local bin boundaries, record-level bin indices
Deterministic hashing	Hash-space configuration and the applied deterministic mapping rule	Shared plaintext category dictionary, local category lists
DirSign aggregation	Normalized PMF correction directions and class supports	Full local feature-class count tables in the proposed aggregation branch
Count-based reference	Local feature-class count statistics	Not a privacy-oriented branch; included only as a performance reference

Appendix B. Additional sensitivity figures and tables

See Tables B.10–B.13.

Data availability

The data used in this study are publicly available from the cited sources.

References

Banse, A., Kreischer, J., & i Jürgens, X. O. (2024). Federated learning with differential privacy. arXiv preprint arXiv:2402.02230 URL https://arxiv.org/abs/2402.02230.

Becker, B., & Kohavi, R. (1996). Adult [dataset]. http://dx.doi.org/10.24432/C5XW20, URL https://archive.ics.uci.edu/dataset/2/adult.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., & Anandkumar, A. (2018). SignSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th international conference on machine learning (ICML 2018)* (pp. 560–569). PMLR.

Bernstein, J., Zhao, J., Azizzadenesheli, K., & Anandkumar, A. (2019). SignSGD with majority vote is communication efficient and fault tolerant. In *International conference on learning representations*. arXiv:1810.05291 URL https://openreview.net/forum?id=BJxhijAcY7.

Blackard, J. A. (1998). *Covertype*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C50K5N.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 1175–1191). ACM, http://dx.doi.org/10.1145/3133956.3133982.

Carletti, V., Foggia, P., Mazzocca, C., Parrella, G., & Vento, M. (2025). SoK: Gradient inversion attacks in federated learning. In *34th USeNIX security symposium (USeNIX security 25)* (pp. 6439–6459). USENIX Association, URL https://www.usenix.org/conference/usenixsecurity25/presentation/carletti.

Chen, J., Yan, H., Liu, Z., Zhang, M., Xiong, H., & Yu, S. (2024). When federated learning meets privacy-preserving computation. *ACM Computing Surveys*, *56*(12), 1–36. http://dx.doi.org/10.1145/3679013.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under Zero-One loss. *Machine Learning*, *29*, 103–130. http://dx.doi.org/10.1023/A:1007413511361.

Dominini, D., Marchesi, A., Marcelli, A., Peverelli, V., Resta, M., & Resta, G. (2025). ProFed: A benchmark for proximity-based non-IID federated learning. In *ICLR tiny papers (openReview)*. OpenReview preprint/extended abstract.

Dwork, C., & Roth, A. (2014). *Foundations and trends in theoretical computer science*: *vol. 9*, *The algorithmic foundations of differential privacy* (3–4), (pp. 211–407). Now Publishers, ISBN: 978-1-60198-818-8, http://dx.doi.org/10.1561/0400000042.

Guo, Z., Xu, L., & Zhu, L. (2023). FedSIGN: A sign-based federated learning framework with privacy and robustness guarantees. *Computers & Security*, *135*, Article 103474. http://dx.doi.org/10.1016/j.cose.2023.103474.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann, ISBN: 978-0-12-381480-7.

Huang, Y., Gupta, S., Song, Z., Li, K., & Arora, S. (2021). Evaluating gradient inversion attacks and defenses in federated learning. In *Advances in neural information processing systems*. http://dx.doi.org/10.48550/arXiv.2112.00059, URL https://arxiv.org/abs/2112.00059.

Jimenez, G. D. M., Solans, D., Heikkilä, M., Vitaletti, A., Kourtellis, N., Anagnostopoulos, A., & Chatzigiannakis, I. (2024). Non-IID data in federated learning: A survey with taxonomy, metrics, methods, frameworks and future directions. ArXiv Preprint arXiv:2411.12377.

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. In *Proceedings of machine learning and systems* (pp. 429–450). URL https://proceedings.mlsys.org/.

Liu, Y., Kang, Y., Xing, C., Chen, T., & Yang, Q. (2024). Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, *36*(7), 3615–3634. http://dx.doi.org/10.1109/TKDE.2024.3352628.

Liu, B., Lv, N., Guo, Y., & Li, Y. (2024). Recent advances on federated learning: A systematic survey. *Neurocomputing*, *597*, Article 128019. http://dx.doi.org/10.1016/j.neucom.2024.128019.

Ma, J., Naas, S.-A., Sigg, S., & Lyu, X. (2021). Privacy-preserving federated learning based on multi-key homomorphic encryption. http://dx.doi.org/10.48550/arXiv.2104.06824, arXiv:2104.06824 URL https://arxiv.org/abs/2104.06824.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge, UK: Cambridge University Press, ISBN: 978-0-521-86571-5, http://dx.doi.org/10.1017/CBO9780511809071.

Marchioro, T., Bistarelli, A. U., & Gasparini, S. (2022). Differentially private naive Bayes classification for federated systems. *IEEE Access*, *10*, 56643–56655. http://dx.doi.org/10.1109/ACCESS.2022.3169902.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-Efficient learning of deep networks from decentralized data. In *Proceedings of machine learning research*: *vol. 54*, *Proceedings of the 20th international conference on artificial intelligence and statistics* (pp. 1273–1282). PMLR, URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, MA: MIT Press.

Paris, R. (2022). *Credit score classification [dataset]*. Kaggle, URL https://www.kaggle.com/datasets/parisrohan/credit-score-classification. (Accessed 06 January 2026).

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2021). Adaptive federated optimization. In *International conference on learning representations*. URL https://openreview.net/forum?id=LkFG3lB13U5, Also available as arXiv:2003.00295.

scikit-learn developers (2025). CategoricalNB-scikit-learn 1.7.2 API reference. URL https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html, scikit-learn.

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE 1993 international symposium on electronic imaging: science and technology* (pp. 861–870). http://dx.doi.org/10.1117/12.148698.

Torrijos, P., Alfaro, J. C., Gámez, J. A., & Puerta, J. M. (2025). Federated learning with discriminative Naive Bayes classifier. In *Lecture notes in computer science*: *vol. 15347*, *Intelligent data engineering and automated learning – IDEAL 2024* (pp. 328–339). Springer, http://dx.doi.org/10.1007/978-3-031-77738-7_27.

Yang, L., Chai, D., Zhang, J., Jin, Y., Wang, L., Liu, H., Tian, H., Xu, Q., & Chen, K. (2023). A survey on vertical federated learning: From a layered perspective. http://dx.doi.org/10.48550/arXiv.2304.01829, arXiv:2304.01829 URL https://arxiv.org/abs/2304.01829.

Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *vol. 80*, In *Proceedings of the 35th international conference on machine learning* (pp. 5650–5659). PMLR.

Zhang, H. (2004). The optimality of Naive Bayes. In *Proceedings of the seventeenth international florida artificial intelligence research society conference* (pp. 562–567). AAAI Press, URL https://aaai.org/papers/flairs-2004-097/.

Zhang, X., Chen, Y., & Li, J. (2025). A review of research on secure aggregation for federated learning. *Future Internet*, *17*(7), 308. http://dx.doi.org/10.3390/fi17070308.