Cost-Efficient Feature Selection for Horizontal Federated Learning

Sourasekhar Banerjee[1], Devvjiit Bhuyan[1], Erik Elmroth[1], and Monowar Bhuyan[1]

[1]Affiliation not available

September 11, 2024

Cost-Efficient Feature Selection for Horizontal Federated Learning

Sourasekhar Banerjee, *Student Member, IEEE*, Devvjiit Bhuyan, Erik Elmroth *Member, IEEE*, and Monowar Bhuyan, *Senior Member, IEEE*

***Abstract*—Horizontal Federated Learning exhibits substantial similarities in feature space across distinct clients. However, not all features contribute significantly to the training of the global model. Moreover, the curse of dimensionality delays the training. Therefore, reducing irrelevant and redundant features from the feature space makes training faster and inexpensive. This work aims to identify the common feature subset from the clients in federated settings. We introduce a hybrid approach called Fed-MOFS[1], utilizing Mutual Information and Clustering for local feature selection at each client. Unlike the Fed-FiS, which uses a scoring function for global feature ranking, Fed-MOFS employs multi-objective optimization to prioritize features based on their higher relevance and lower redundancy. This paper compares the performance of Fed-MOFS [2] with conventional and federated feature selection methods. Moreover, we tested the scalability, stability, and efficacy of both Fed-FiS and Fed-MOFS across diverse datasets. We also assessed how feature selection influenced model convergence and explored its impact in scenarios with data heterogeneity. Our results show that Fed-MOFS enhances global model performance with a 50% reduction in feature space and is at least twice as fast as the FSHFL method. The computational complexity for both approaches is O(d^2), which is lower than the state-of-the-art.**

***Impact Statement*—This work aims to identify common feature subsets in federated settings for Horizontal Federated Learning (HFL). We extended Fed-FiS and proposed a new hybrid federated feature selection approach, Fed-MOFS that ranks features based on high relevance and low redundancy criteria. Both approaches improve global model performance by reducing 50% size of the features and are $2\times$ faster than FSHFL based on wall-clock running time analysis. The computational complexity is also lower than the state-of-the-art federated feature selection methods. Fed-MOFS and Fed-FiS are scalable and compatible with non-IID data. Also, feature selection does not hamper the convergence of the global model.**

***Index Terms*—Clustering, Horizontal Federated Learning, Feature Selection, Mutual Information, and Multi-objective Optimization.**

The manuscript is submitted for review on January 20, 2024. Resubmitted on June 13, 2024, Accepted July 17, 2024.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. The computations were enabled by the supercomputing resource Berzelius provided by National Super computer Centre at Linkoping University and the Knut and Alice Wallenberg Foundation.

Sourasekhar Banerjee is with the Department of Computing Science, Umeå University, Umeå, SE-90187, Sweden (e-mail: sourasb@cs.umu.se).

Devvjiit Bhuyan is with the Department of Electronics and Communication Engineering, Tezpur University, Assam-784028, India (e-mail: ecb19050@tezu.ac.in).

Erik Elmroth is with the Department of Computing Science, Umeå University, Umeå, SE-90187, Sweden (e-mail: elmroth@cs.umu.se).

Monowar Bhuyan is with the Department of Computing Science, Umeå University, Umeå, SE-90187, Sweden (e-mail: monowar@cs.umu.se).

This paragraph will include the Associate Editor who handled your paper.

[1]This manuscript is an extension of Banerjee et al. [1]

[2]We share our code, data, and supplementary copy through https://github.com/DevBhuyan/Horz-FL/blob/main/README.md.

I. Introduction

FEDERATED LEARNING (FL) is a novel machine learning paradigm that facilitates collaborative model training among multiple data owners (a.k.a. clients) [2], [3]. This occurs through the iterative exchange of model parameters via an FL server, all while maintaining the privacy of individual clients without sharing local data. According to the intersection or distribution of data among clients in terms of sample space or feature space, federated learning can be classified into three main categories: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL) [4]. HFL operates with data, sharing a uniform feature space across all clients. Whereas, VFL leverages dissimilar data with distinct feature spaces to train a global model collaboratively. Conversely, FTL uses a pre-trained model initially trained on similar data to solve different problems. HFL scenarios frequently occur in practical use cases, such as in internet-of-vehicles [5], smart grid analysis [6], e-commerce [7], health-care [8], etc. The efficacy of local models is influenced by the quality of local features possessed by clients, consequently impacting the overall performance of the global model. Clients may have irrelevant or noisy features for the learning task, or they may have an excessive number of redundant features, leading to a significant degradation in the performance of the global model. Therefore, the absence of feature selection results in poor model performance and extends the duration of model training. Therefore, it is crucial to identify appropriate feature sets that overlap among the clients, as this can lead to reduced training time and energy consumption, which results in reduced communication rounds without compromising the global model's performance.

Feature Selection (FS) is a crucial preprocessing technique in a centralized Machine Learning (ML) framework. It has been extensively studied and proven

valuable in data mining, knowledge discovery, and ML. The primary goal of FS is to identify the most pertinent, trustworthy, and non-redundant features from extensive datasets. This process enhances model performance and facilitates cost-effective learning of models. The choice of features is also very important when working with high-dimensional datasets, where the number of features is much higher than the number of samples. This makes it hard to find the right features for making cost-effective learning of models. In the context of deep learning, identifying meaningful features within a data set is significant. Representation learning algorithms extract valuable patterns from raw data, generating representations that simplify processing. These representations can be crafted for interpretability, to reveal hidden features, or utilized in transfer learning applications. In contrast to dimension reduction techniques such as principal component analysis (PCA) [9], feature selection does not modify the original features. Instead, it identifies and chooses a subset of the most valuable features from the dataset during runtime. The feature selection process involves evaluating the importance of each feature using various methods, such as correlation [10], mutual information [11], chi-square [12], etc. After identifying crucial features, they are employed to construct precise, effective, and budget-friendly ML models for prediction or classification. Additionally, feature selection enhances interpretability by pinpointing the key variables influencing the model's predictions. Information-theoretic measures have been widely utilized and established as a paradigm for filter-based feature selection. Specifically, Mutual Information-based Feature Selection (MIFS) empowers the feature selection method by removing redundant and irrelevant features without impacting the classifier's reachable performance. Traditional MIFS approaches [11], [13], [14], such as MIFS-ND are designed for centralized systems where data is available in a centralized server.

Why does feature selection benefit in federated learning? When each client independently performs feature selection and builds individual local models based on their private data without exchanging the selected feature set with the server, it impacts the global model updates and performance. This results in objective shifts by cause of feature selection bias [15] and statistical heterogeneity among clients. Furthermore, if clients opt for distinct feature subsets, it poses challenges in ensuring homogeneous model training across all clients in HFL settings. These issues inspired us to develop a feature selection algorithm tailored to the context of HFL.

In our previous work, Fed-FiS [1] evaluates the importance of the features by a score function and generates global ranks of each feature from higher to lower scores. Here, we introduce Fed-MOFS, a new federated feature ranking and selection method based on multi-objective optimization. The resulting method optimizes the relevance and redundancy of the feature set simultaneously. It produces a non-dominated solution set or Pareto fronts from where we get the ranking of the features. The local feature selection of Fed-FiS and Fed-MOFS remain the same, but the global feature selection and ranking differ.

Why does global feature ranking of individual features affect FL performance?
All clients share a common set of features, but the importance of these features varies among clients. Determining a unified and relevant set of features across all clients is challenging, as the features selected locally by each client may differ. For instance, in Table III, feature f_4 is crucial for clients Cl_2 and Cl_3 but not for Cl_1. If we choose features that are locally selected by all clients in common, we must omit f_4. In the worst-case, there might be no overlap in locally selected features among clients, making it difficult to train a global model. To address this challenge, we consider two approaches: Fed-FiS (refer to Table IV) and Fed-MOFS (refer to Table V). These approaches provide global rankings for each feature. For example, f_4 obtains global rankings of 3, i.e., the third important feature using Fed-FiS and 4, which means the fourth important feature using Fed-MOFS. Thus, calculating the global ranking for each feature helps to achieve a fair assessment, ensuring a balanced representation of the importance of the feature in the training process.

The main contributions of this work are summarized as follows.

- We introduce Fed-MOFS, an approach for global ranking and feature selection based on multi-objective optimization for horizontal federated learning. It employs mutual information and 1-D clustering to choose a local set of features. Furthermore, it utilizes Pareto optimization to create a global ranking of these features and selects features from the Pareto fronts (see Section IV-C).
- We derived the computational complexity of both the Fed-FiS and Fed-MOFS algorithms (see Section IV-D) and compared them with the current state-of-the-art horizontal federated learning algorithms (see Table I) that explicitly demonstrate the computational cost, benefits and drawbacks.
- We conducted a comprehensive empirical study comparing the performance, scalability, efficiency, stability, and convergence of both Fed-FiS and Fed-MOFS across various datasets, including NSL-KDD99, Wine, Vowel, Vehicle, Segmentation, WDBC, Ionosphere, Hill-Valley, ISOLET, Diabetes, IoT, Anonymized Credit Card (ACC), Boston housing prices, California house pricing, and synthetic data. This analysis utilized conventional feature selection techniques within federated settings, such as RFE and ANOVA, as well as federated feature selection methods like FSHFL [16] and Fed-mRMR [17]. For classification tasks, we trained Federated Forest [18] and

federated averaging [3] on Deep Neural Networks (DNNs), and for regression tasks, we employed ridge regression, all following the application of various feature selection methods in federated settings (see Section V).

II. Related Work

This section provides a comprehensive overview of three key related research areas: centralized feature selection, distributed feature selection, and horizontal federated feature selection.

A. *Centralized feature selection*

In centralized settings, computation happens on a single system. Data is collected and resides in a centralized server; hence, it is easy to access and build models for a specific task. Feature selection is comparatively easy as the server has complete information about the data. Regarding the availability of class information, feature selection methods are categorized as supervised [19], semi-supervised [20], and unsupervised [21] approaches. Also, depending on how feature selection methods interact with the classifier, the existing works are categorized as filter [22], wrapper [23], embedded [24], and hybrid [25]. **Filter** methods are statistical approaches that select features based on their relevance to the class without considering the employed model. Examples include correlation-based [10], mutual information-based [11], and chi-squared [12] based feature selection methods. **Wrapper** methods involve training a model using a subset of features and evaluating the performance of the model [26], [27]. Examples of wrapper methods include forward selection, backward elimination, and recursive feature elimination. **Embedded** methods incorporate feature selection within the model-building process. For example, lasso regression [28], ridge regression [29], and elastic net [30]. **Hybrid** methods combine multiple feature selection techniques, such as filter and wrapper, to improve the model's overall performance.

The feature selection helps to describe data better for extracting valuable knowledge from high dimensional data [13] and solve problems that occur due to the higher dimensionality of data. This reduces the model's computation cost by reducing feature space and improves overall learning performance by selecting relevant and less redundant features. Mutual Information (MI) based feature selection is a filter method that chooses strongly correlated features with labels and has minimal redundancy among feature sets. The MI method adopts the entropy difference to calculate the information a feature contributes. MIM [31] rapidly selects label-related features. However, the consideration of feature redundancy is absent. MIFS [32] proposes feature redundancy as a metric for assessing the quality of features. Methods in [11], [33]–[35] also consider relevance and redundancy relationship to select features. The application of feature selection is wide and beneficial. Some real-life applications of feature selection are: healthcare analytics [36]–[39], image analysis [40] and recognition [41], credit scoring [42]–[44], marketing analysis [45]–[48], anomaly detection [49]–[53], bio-informatics [54]–[56], etc.

These feature selection methods are not directly applicable to federated learning because, in federated settings, not all clients have complete information about the data.

B. *Distributed feature selection*

Although centralized feature selection approaches are fast and effective. They struggle to achieve satisfactory performance when dealing with big data, which is characterized not only by its large volume but also by its diverse and intricate nature. Therefore, a specific distributed feature selection method is necessary. In [57], the authors compare distributed vs centralized feature selection. Several rounds of feature selection are performed on horizontal as well as vertical partitions of data. Finally, the outputs of every round are combined and produce a single subset of relevant features using different data complexity measures [58]. An adaptive aggregation (ADAGES) flexible distributed feature selection method is proposed in [59]. A distributed fuzzy rough set (DFRS) based feature selection method to enable fuzzy rough set for big data analysis is presented in [60]. A distributed quadratic programming-based feature selection is reported in [61]. In [62], the authors formalized the feature selection problem as a diversity maximization problem by proposing an MI-based metric distance on features. They focused on vertically distributed feature selection that can deal with redundancy.

FL has additional characteristics, including imbalanced and massively distributed IID (see Definition 2) and non-IID (see Definition 3) data, and clients with limited computational capacity, despite the fact that distributed learning and federated learning appear to be similar. However, the idea behind FL is that data should remain private to the clients.

C. *Horizontal federated feature selection*

Federated feature selection is challenging because the distribution of the whole dataset is unknown for each client. Federated feature selection is first introduced in [1], where authors formulated the feature selection problem for horizontal federated learning. The authors proposed a hybrid feature selection method using mutual information and clustering. Fed-FiS is stable while data distribution is IID, i.e., every client has information on features and classes but is not stable for vertical or hybrid distributions. Cassara et al. [63] proposed federated feature selection for cyber-physical systems. Their approach involves a mutual

information-based feature selection algorithm run by the autonomous vehicles (clients) and Bayes' theorem-based aggregation executed on the server. Hu et al. [64] proposed a federated feature selection algorithm using evolutionary computing techniques. Zhang et al. [16] proposed an unsupervised feature selection method for horizontal federated learning where clients share a common feature space but have different class labels. In [65], suggested a greedy algorithm for feature selection. In Table I, we report a comparative study by considering multiple key factors between our method and the current state-of-the-art horizontal federated feature selection methods [1], [16], [17], [63], [65].

III. Problem Statement

Consider a HFL system consists of q clients $(\forall_{i=1}^{q} Cl_i)$ and a server. We assume that $q \geq 2$, if $q = 1$, it is considered a centralised system with full dataset information. Suppose the dataset D contains samples $S \in \mathbb{R}^{n \times d}$, the features set $F \in \mathbb{R}^{d \times 1}$, and class $C \in \{0,1,\ldots,k\}^{n \times 1}$. D is distributed across q clients such that each client contains the features set F. Sample set $S_{Cl_i} \in \mathbb{R}^{m \times d}$, where $\cup_{i=1}^{q} S_{Cl_i} = S$, and $\cap_{i=1}^{q} S_{Cl_i} = \emptyset$ and class $C_{Cl_i} \in \{0,\ldots,k\}^{m \times 1}$, where $m < n$. In HFL, all clients have partial class information, which creates statistical heterogeneity. Our objective is to uncover relevant features subset (F'') and obtain stable and generalizable global model performance.

IV. Proposed Approach

Our approaches comprise two components: (1) local feature selection performed independently by clients using mutual information and clustering, and (2) global feature selection achieved through a global score function for Fed-FiS and multi-objective optimization for Fed-MOFS.

A. Data division

For a given dataset $D(F,S) \in \mathbb{R}^{n \times d}$, consists of feature set $F = \{f_1, f_2, \ldots, f_d\}^T$, $F \in \mathbb{R}^{d \times 1}$, and sample set $S = \{s_1, s_2, \ldots, s_n\}$, $S \in \mathbb{R}^{n \times d}$. The dataset is distributed across q clients in a horizontal (Figure 1) manner. In horizontal federated learning, all clients have access to the same set of features but different samples. The dataset is considered to be Independent and Identically Distributed (IID) if each client possesses complete information about all the classes. On the other hand, if clients only have partial information about the classes, the data distribution is referred to as Non-Independent and Identically Distributed (non-IID). We introduced the following foundational definitions to understand the federated settings.

Definition 1. ***Horizontal:*** *For a given dataset $D(F,S) \in \mathbb{R}^{n \times d}$, consists of feature set F, and sample set S, distributed across q clients. Then all clients have similar feature sets but different samples, i.e., $\bigcap_{i=1}^{q} F_{Cl_i} = F$ but $\bigcap_{i=1}^{q} S_{Cl_i} = \emptyset$, where Cl_i is the i^{th} client, F_{Cl_i} and the S_{Cl_i} are the feature and sample set of Cl_i, respectively.*

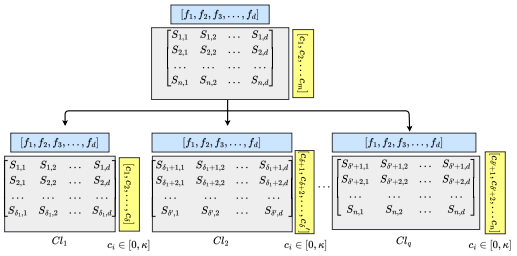


Figure 1: Horizontal data division in HFL

Lemma 1. *All clients share common feature set, $F_{Cl_i} \cup F_{Cl_j} = F$.*

Proof: Lets consider clients Cl_i and Cl_j, if $F_{Cl_i} \bigcap F_{Cl_j} = \emptyset$, then there is no common features between clients Cl_i and Cl_j. According to Definition 1, $F_{Cl_i} \bigcap F_{Cl_j} \neq \emptyset$. By contradiction, we proved that the Lemma 1 is true. ■

Definition 2. ***IID:*** *For a given dataset, $D(F,S) \in \mathbb{R}^{n \times d}$, $C = \{c_1, c_2, \ldots, c_\kappa\}$ consisting of κ classes, it would be called IID if the probability distribution of each class c_i is independent of other class $P(c_i|c_1, c_2, \ldots, c_\kappa) = \mathscr{P}(c_i)$ and all classes are drawn from the same underlying probability distribution $\mathscr{P}(c)$, $\mathscr{P}(c_1 = c, c_2 = c, \ldots c_\kappa = c) = \mathscr{P}(c)$.*

Definition 3. ***Non-IID:*** *For a given dataset, $D(F,S) \in \mathbb{R}^{n \times d}$, $C = \{c_1, c_2, \ldots, c_\kappa\}$ consisting of κ classes, it would be called non-IID if the probability distribution of each class c_i depends on the values or presence of other classes within dataset $\mathscr{P}(c_i|c_1, c_2, \ldots, c_\kappa) \neq \mathscr{P}(c_i)$, and the classes are drawn from different underlying probability distributions, across different clients, $\mathscr{P}(c_1) \neq \mathscr{P}(c_2) \neq \ldots \neq \mathscr{P}(c_\kappa)$.*

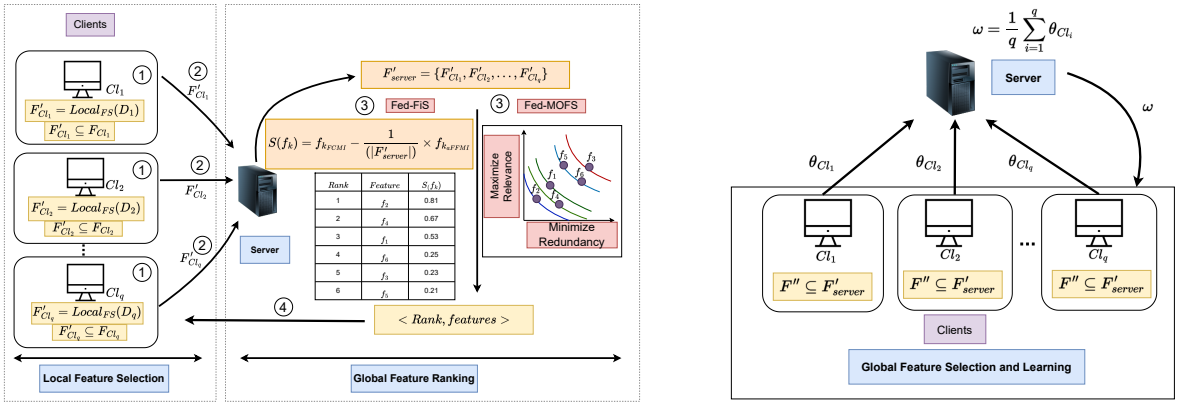
B. Framework

A conventional HFL consists of q clients $(Cl_1, Cl_2, \ldots, Cl_q)$ and a server. Figure 2a illustrates the proposed framework in four steps, as follows.

1) Each client Cl_i has its private feature set F_{Cl_i} and runs the procedure $Local_{FS}(D_i)$ independently to generate local feature subset F'_{Cl_i} $(F'_{Cl_i} \subseteq F_{Cl_i})$.
2) Each client Cl_i sends the FCMI (see Definition 4) and aFFMI (see Definition 5 and 6) value of each $f_i^{Cl_i} \in F'_{Cl_i}$ to the server. Server averages these scores of similar features and generate a list of unique feature set F'_{server}.
3) Server applies Fed-FiS or Fed-MOFS to generate the global ranks of the locally selected features.
4) Server sends the global ranks of each feature to the clients.

TABLE I: Comparisons among prior works on feature selection in FL settings. None of the current methods fully meet the specified requirements (C1)–(C5)

Federated Feature selection algorithms	Category (C1)	Feature Relevance (C2)	Feature Redundancy (C3)	Global Feature Selection using Multi-objective optimization (C4)	Computational Complexity (C5)	Benefits	Drawbacks
CE-based FFS [63] [2022]	Filter	✓	✓	×	$O(n*d)$	• Robustness against non-IID data	• Complexity in the aggregation function
Greedy-Feature Selection [65] [2021]	Wrapper	×	×	×	$O(nd^2)$	• Adapt to various attack types by selecting different feature sets for each type of attack • The algorithm is straightforward and can effectively identify non-contributing or redundant features	• Suboptimal feature set due to greedy nature • The outcome heavily depends on the initial set of features
FSHFL [16] [2023]	Filter	✓	×	×	$O(d^2 \log d)$	• Improvement in performance • Reduced training time and energy consumption • Privacy benefit for federated learning	• High in complexity • Hierarchical clustering is time-consuming, therefore not scalable
Fed-mRmR [17] [2024]	Filter	✓	✓	×	$O(nd^2)$	• Lossless feature selection • Capability of finding features from non-IID distributions	• Scalability issues during handling large datasets • High in computation complexity
Fed-FiS [1] [2021]	Hybrid	✓	✓	×	$O(d^2)$	• Improved handling of heterogeneity • Reduced computation and communication cost • Stable and relevant feature selection	• Highly dependent on mutual information measures • Potential computational overhead on devices
Fed-MOFS This work	Hybrid	✓	✓	✓	$O(d^2)$	• stable across IID and non-IID data distribution • Scalable across multiple clients • Stable and relevant feature selection	• Highly dependent on mutual information measures • Potential computational overhead on devices



(a) Global feature ranking

(b) Feature selection and learning

Figure 2: Proposed framework

TABLE II: Notations

Symbol	Description	Symbol	Description
D	Dataset, $D \in \mathbb{R}^{n \times d}$.	F	Feature space $F = \{f_1, f_2, \ldots, f_d\}^\intercal$, $F \in \mathbb{R}^{d \times 1}$.
n	Number of samples	F'_{server}	The global feature set that contains unique features only.
d	Number of features	F''	The feature subset according to the rank of each feature.
S	Sample space, $S \in \mathbb{R}^{n \times d}$.	F'_{Cl_i}	Locally selected features at client Cl_i.
D_i	Dataset of client Cl_i	$\tau_k^{Cl_i}$	Feature triplet for k^{th} feature of Cl_i
β	Number of clusters or cluster centroids	ζ	Current cluster or cluster centroid
δ	Performance threshold of global model	ψ	Performance of global model.
C	Class, $C \in \{0, \ldots, \kappa\}^{n \times 1}$.	q	Number of clients.
θ_{Cl_i}	Local model of the i^{th} client.	ω	Global model.
Cl_i	i^{th} client from a pool of q clients.	$\tau_k^{Cl_i}$	Triplet of the k^{th} locally selected feature of the i^{th} client.
F_{Cl_i}	Feature set available at the i^{th} client.	S_{Cl_i}	Sample set available at the i^{th} client.
$FCMI$	Feature Class Mutual Information.	$aFFMI$	Averaged Feature Feature Mutual Information.
$f_{k_{FCMI}}^{Cl_i}$	FCMI of the k^{th} feature of the i^{th} client.	$f_{k_{aFFMI}}^{Cl_i}$	aFFMI of the k^{th} feature of the i^{th} client.
$f_k^{Cl_i}$	i^{th} feature of the client Cl_i.	C_{Cl_i}	Set of target class available at the i^{th} client.
$\lvert F_{Cl_i} \rvert$	Number of features at client Cl_i.	$\lvert S_{Cl_i} \rvert$	Number of samples at client Cl_i.
$F_{Cl_i}^{FCMI}$	Set of features with FCMI values for the i^{th} client	$F_{Cl_i}^{aFFMI}$	Set of features with aFFMI values for the i^{th} client
$F'_{Cl_i FCMI}$	Set of features of Cl_i that belong to the clusters that have maximum centroid value.	$F'_{Cl_i aFFMI}$	Set of features of Cl_i that belong to the clusters that have minimum centroid value.
$f_{k_{FCMI}}$	Average of FCMI value of feature f_k across all clients.	$f_{k_{aFFMI}}$	Average of aFFMI value of feature f_k across all clients.
T	Global rounds	ε	Number of features selected from the ranked features for training.
Δ	Fraction of clients participated $\Delta \in (0, 1)$	γ	Non-IID factor $\gamma \in [0, 1]$

In Figure 2b, after having the rank of each feature, all clients start FL with a feature subset ($F'' \subseteq F'_{server}$).

C. Algorithms

The federated feature selection approaches are described with four algorithms. Algorithm 1 for local feature selection, Algorithm 2 and 3 described global feature ranking using Fed-FiS and Fed-MOFS, respectively, and finally, Algorithm 4 for global feature selection.

1) ***Local feature selection:*** We adopted Mutual Information (MI) to measure the certainty of a feature variable with a target variable, which could be another feature or a class [14], [66]. The MI-based feature selection approach depends on the relevance of each feature, measured by Feature-Class Mutual Information (FCMI, see Definition 4), as well as the redundancy, measured by, Feature-Feature Mutual Information (FFMI, see Definition 5) of each feature. The average of all FFMI values of features is called aFFMI (see Definition 6).

Definition 4. ***FCMI:*** *Given a feature $f_k^{Cl_i}$ at client Cl_i, and class C, then FCMI of feature $f_k^{Cl_i}$ and C can be*

computed as:

$$FCMI(f_k^{Cl_i},C) \triangleq \sum_{f_k^{Cl_i},C} \mathscr{P}(f_k^{Cl_i},C) log \frac{\mathscr{P}(f_k^{Cl_i},C)}{\mathscr{P}(f_k^{Cl_i})\mathscr{P}(C)} \tag{1}$$

where $f_k^{Cl_i}$ is the k^{th} feature of the i^{th} client (Cl_i). $\mathscr{P}(f_k^{Cl_i})$ and $\mathscr{P}(C)$ are the marginal and $\mathscr{P}(f_k^{Cl_i},C)$ is the joint probability distribution for $f_k^{Cl_i}$ and C.

Definition 5. ***FFMI:*** *Given two features $f_k^{Cl_i}$ and $f_j^{Cl_i}$, at client Cl_i, then the FFMI of features $f_k^{Cl_i}$ and $f_j^{Cl_i}$ can be estimated as:*

$$\begin{aligned} FFMI(f_k^{Cl_i},f_j^{Cl_i}) \triangleq MI(f_k^{Cl_i};f_j^{Cl_i}) = H(f_k^{Cl_i}) + \\ H(f_j^{Cl_i}) - H(f_k^{Cl_i},f_j^{Cl_i}) \end{aligned} \tag{2}$$

where $f_k^{Cl_i}$ and $f_j^{Cl_i}$ are k^{th} and j^{th} feature of the i^{th} client (Cl_i), respectively, and $f_k^{Cl_i} \neq f_j^{Cl_i}$. $H(f_k^{Cl_i})$ and $H(f_j^{Cl_i})$ are marginal entropy. $H(f_k^{Cl_i},f_j^{Cl_i})$ is the joint entropy of the $f_k^{Cl_i}$ and $f_j^{Cl_i}$, and $f_k^{Cl_i} \neq f_j^{Cl_i}$.

Definition 6. ***aFFMI:*** *Given a set of d features at client Cl_i, then averaged FFMI (aFFMI) value of $f_k^{Cl_i}$ can be calculated as:*

$$aFFMI(f_k^{Cl_i}) = \frac{1}{d-1} \sum_{j=1, f_j^{Cl_i} \in F_{Cl_i} \setminus f_k^{Cl_i}}^{d-1} FFMI(f_k^{Cl_i},f_j^{Cl_i}) \tag{3}$$

where $f_k^{Cl_i}$ and $f_j^{Cl_i}$ are k^{th} and j^{th} feature of the i^{th} client (Cl_i), respectively, and $f_k^{Cl_i} \neq f_j^{Cl_i}$.

Algorithm 1 Local feature selection

Input: $F_{Cl_i} = \{f_1^{Cl_i}, f_2^{Cl_i}, \ldots, f_d^{Cl_i}\}$ is the original feature set and d is the dimension of the data for the i^{th} client Cl_i
Output: F'_{Cl_i} is the selected features for client i

1: **procedure** $Local_{FS}(D_i)$
2: **for** $f_k^{Cl_i} \in F_{Cl_i}$ **do**
3: $f_k^{FCMI} = FCMI(f_k^{Cl_i},C)$ using Equation (1). ▷ return FCMI score of a feature
4: $f_k^{aFFMI} = aFFMI(f_k^{Cl_i})$ using Equation (2) and Equation (3). ▷ return averaged FFMI score of a feature
5: $F_{Cl_i}^{FCMI} = \{f_k^{FCMI} | \forall_{k=1}^{d} f_k^{FCMI} \in [0,1]\}$
6: $F_{Cl_i}^{aFFMI} = \{f_k^{aFFMI} | \forall_{k=1}^{d} f_k^{aFFMI} \in [0,1]\}$
7: $F'_{Cl_i FCMI}$ = CLUSTER($F_{Cl_i}^{FCMI}$) ▷ return cluster of features with high FCMI values
8: $F'_{Cl_i aFFMI}$ = CLUSTER($F_{Cl_i}^{aFFMI}$) ▷ return cluster of features with low aFFMI values
9: $F'_{Cl_i} = F'_{Cl_i FCMI} \cup F'_{Cl_i aFFMI}$
10: return F'_{Cl_i}
11: **procedure** CLUSTER($F_{Cl_i}^x$) ▷ here, $F_{Cl_i}^x$ is $F_{Cl_i}^{FCMI}$ or $F_{Cl_i}^{aFFMI}$
12: Initialize β random cluster centroid.
13: **repeat**
14: $\forall_{k=1}^{|F_{Cl_i}^x|} f_k \in F_{Cl_i}^x$
15: minimum $\leftarrow$ 0
16: cluster_member $\leftarrow$ 0
17: $\forall_{\zeta=1}^{\beta} centroid\ \zeta \in \beta$
18: dist $\leftarrow$ Distance(f_k, ζ)
19: **if then** $dist < minimum$
20: minimum $\leftarrow$ dist
21: cluster_member $\leftarrow$ ζ
22: recalculate centroid(ζ)
23: **until** Converge
24: return $F'_{Cl_i{}^x}$ ▷ here, $F'_{Cl_i{}^x}$ is $F'_{Cl_i FCMI}$ or $F'_{Cl_i aFFMI}$

Algorithm 1 computes the FCMI and aFFMI values of all features (Line 2 to Line 4) at each client using the Equation (1), Equation (2), and Equation (3), respectively. $F_{Cl_i}^{FCMI}$ and $F_{Cl_i}^{aFFMI}$ are two one-dimensional vectors that contain FCMI and aFFMI values of all features at client Cl_i. The size of the vector depends on the number of features present at the client, but it is bound to d. The FCMI and aFFMI values are within the range of 0 to 1 (Line 5 and Line 6). The FCMI value close to zero indicates the low relevance of that feature. The aFFMI value close to one indicates the high redundancy of that feature. Here, we aim to find the optimal feature set by (1) maximizing the relevance and (2) minimizing the redundancy. Based on the FCMI and aFFMI values, we compute CLUSTER($F_{Cl_i}^{FCMI}$) and CLUSTER($F_{Cl_i}^{aFFMI}$) (Line 7 and Line 8) using the procedure CLUSTER (Line 11 to Line 24) to generate feature clusters with higher FCMI and lower aFFMI values. If there are β clusters of features, then the objective can be written as follows.

$$F'_{Cl_i{}^{FCMI}} = \arg\max_{\forall i \in \beta} Centroid(cluster_i) \tag{4}$$

$$F'_{Cl_i{}^{aFFMI}} = \arg\min_{\forall i \in \beta} Centroid(cluster_i) \tag{5}$$

where $Centroid(cluster_i)$ returns the centroid value of the i^{th} cluster, and $|\ cluster_i\ |$ is the cardinality of $cluster_i$.

In Equation (4), select the cluster with the maximum centroid($F'_{Cl_i{}^{FCMI}}$). It contains the features with utmost relevance and $F'_{Cl_i{}^{FCMI}} \subseteq F_{Cl_i}$. Similarly, Equation (5) returns the cluster with the minimum centroid ($F'_{Cl_i{}^{aFFMI}}$). It contains features with minimum redundancy and $F'_{Cl_i{}^{aFFMI}} \subseteq F_{Cl_i}$. Union of the output of Line 7 and Line 8 produces the final local feature subset (Line 9). The example of local feature selection is in illustration IV-C1.

Illustration: In Table III, five features, f_1, f_2, f_3, f_4, and f_5 have been distributed across three clients, Cl_1, Cl_2, and, Cl_3 horizontally. The FCMI and aFFMI scores of each feature are estimated on each client independently, and then the clustering method is employed on the FCMI and aFFMI scores separately and divided into two clusters, C1 and C2. The value of cluster center of $C1 > C2$. Therefore, a union between C1 from Clustered FCMI and C2 from Clustered aFFMI has to be performed to get the locally selected features.

*2) **Fed-FiS**:* In Algorithm 2, each client, Cl_i sends triplets of locally selected features to the server. The definition of the feature triplet is given below.

Definition 7. ***Feature triplet*** ***($\tau_k^{Cl_i}$):*** *$\forall_{i=1}^{q} Cl_i$, a feature $f_k^{Cl_i} \in F'_{Cl_i}$ then feature triplet of $f_k^{Cl_i}$ can be defined as $\tau_k^{Cl_i} =< f_k^{Cl_i}, f_{k_{FCMI}}^{Cl_i}, f_{k_{aFFMI}}^{Cl_i} >$*

TABLE III: Local feature selection

Client(s)	Feature(s)	FCMI	aFFMI	Clustered FCMI	Clusered aFFMI	Locally selected features
Cl_1	f_1	0.67	0.31	**C1:[** f_2,f_1,f_3**]** C2:[f_4,f_5]	C1:[f_2,f_4,f_5] **C2:[** f_1,f_3**]**	f_1,f_2, f_3
	f_2	0.91	0.43			
	f_3	0.57	0.21			
	f_4	0.27	0.61			
	f_5	0.17	0.51			
Cl_2	f_1	0.57	0.32	**C1:[** f_1,f_2,f_3,f_4**]** C2:[f_5]	C1:[f_4, f_5] **C2:[** f_2, f_1, f_3**]**	f_1,f_2, f_3, f_4
	f_2	0.81	0.21			
	f_3	0.37	0.42			
	f_4	0.47	0.53			
	f_5	0.17	0.66			
Cl_3	f_1	0.73	0.31	**C1:[** f_1,f_2,f_4**]** C2:[f_3,f_5]	C1:[f_4] **C2:[** f_2,f_3,f_1, f_5**]**	f_1,f_2 f_3, f_4 f_5
	f_2	0.82	0.17			
	f_3	0.46	0.23			
	f_4	0.51	0.41			
	f_5	0.34	0.31			

where $\tau_k^{Cl_i}$ is the triplet of the k^{th} locally selected feature of the i^{th} client Cl_i. $f_k^{Cl_i}$ is the unique identifier of the feature. $f_{k_{FCMI}}^{Cl_i}$, and $f_{k_{aFFMI}}^{Cl_i}$ are the FCMI and aFFMI score of $f_k^{Cl_i}$ respectively.

Each device sends a vector of triplets to the server. So a triplet vector F'_{Cl_i} of device Cl_i can be defined as $F'_{Cl_i} = \{\tau_1^{Cl_i}, \tau_2^{Cl_i}, \ldots, \tau_k^{Cl_i}\}$, where $k \leq d$. The server receives feature triplets from q clients (Line 2). Multiple clients can share a single feature f_k. Therefore, F_{server} may have multiple similar features with different FCMI and aFFMI scores. Server averages the FCMI and aFFMI values of similar features using the Equation (6), and Equation (7) (Line 3).

$$f_{k_{FCMI}} = \frac{\sum_{i=1}^{j} f_{k_{FCMI}}^{Cl_i}}{j}, where\, j \in [1,q] \tag{6}$$

$$f_{k_{aFFMI}} = \frac{\sum_{i=1}^{j} f_{k_{aFFMI}}^{Cl_i}}{j}, where\, j \in [1,q] \tag{7}$$

Server generates the unique feature list F'_{server} (Line 4). For all features, the server computes a score $S(f_k)$ using Equation (8) (Line 5).

$$S(f_k) = f_{k_{FCMI}} - \frac{1}{(\mid F'_{server} \mid -1)} \times f_{k_{aFFMI}}, \tag{8}$$

where $\mid F'_{server} \mid > 1$ *and* $S(f_k) \in [1,-1]$

Finally, the server sends the score of each feature to the clients (Line 6). We give an example of the working of Fed-FiS in illustration IV-C2.

Algorithm 2 Fed-FiS (Score-based global feature ranking)

Input: $F_{Server} = \{F'_{Cl_1}, F'_{Cl_2} \ldots F'_{Cl_q}\}$ ▷ collection of feature triplets from q clients
Output: $< rank, f_k >$ ▷ global rank of features

1: **procedure** $Fed-FiS(F_{Server})$
2: server obtained $F_{server} = \{F'_{Cl_i} | \forall_{i=1}^{q}, F'_{Cl_i} \in Cl_i\}$.
3: obtain global feature triplet by performing average over FCMI (Equation (6)) and aFFMI (Equation (7)) scores individually.
4: obtain $\{F'_{server} | \forall f_k \in F'_{server}$ *are unique*$\}$
5: compute $S(f_k), \forall\ f_k \in F'_{server}$ using Equation (8)
6: $\forall_{i=1}^{q} Cl_i$ send $< S(f_k), f_k >$ to all Cl_i iff $f_k \in F'_{server}$

Illustration: The server initially computed the average FCMI and averaged aFFMI values for a set of five features, as presented in Table IV. Subsequently, the feature scores were determined using Equation (8). These scores were then utilized to assign ranks to the features in descending order, starting from the highest score, $S(f_k)$. Therefore, The features are ranked in the following order from highest to lowest: f_2, f_1, f_4, f_3, and f_5.

TABLE IV: Illustration of feature ranking with score function in Fed-FiS

Feature(s)	Averaged FCMI	Averaged aFFMI	$S(f_k)$	Rank
f_1	0.66	0.31	0.58	2
f_2	0.85	0.27	0.78	1
f_3	0.47	0.29	0.398	4
f_4	0.52	0.47	0.403	3
f_5	0.34	0.31	0.26	5

Remark 1. *Fed-FiS employs $S(f_k)$ to determine the disparity between the average relevance and redundancy of a feature. When $S(f_k) \to 1$, i.e., this disparity is closer to 1 then the feature is significantly relevant and carries beneficial impact for learning. Conversely, when $S(f_k) \to -1$, it suggests that the feature is less relevant, making it less crucial for learning and potentially causing a negative impact. $S(f_k) = 0$ indicates that the feature possesses equal levels of relevance and redundancy. While it may have a positive impact on learning, it doesn't guarantee the absence of any negative effects.*

3) ***Fed-MOFS****:* Algorithm 3 is also a feature ranking algorithm similar to Fed-FiS, but here server applies a multi-objective optimization to find the dominant features based on the two objective functions, (1) maximizing the average FCMI score, and (2) minimizing the average aFFMI score (Line 6). This multi-objective optimization produces Pareto fronts from where we get the ranking of the features (Line 7). Finally, the server sends the feature with its rank to the clients (Line 8). The example of the working of Fed-MOFS is in illustration IV-C3.

Algorithm 3 Fed-MOFS (Multi-objective optimization based global feature ranking)

Input: $F_{Server} = \{F'_{Cl_1}, F'_{Cl_2} \ldots F'_{Cl_q}\}$ ▷ features from q clients
Output: $< rank, f_k >$ ▷ global rank of features

1: **procedure** $Fed-MOFS(F_{Server})$
2: server obtained $F_{server} = \{F'_{Cl_i} | \forall_{i=1}^{q}, F'_{Cl_i} \in Cl_i\}$.
3: obtains global feature triplet by performing average over aFFMI and FCMI scores individually.
4: obtain $\{F'_{server} | \forall f_k \in F'_{server}$ *are unique*$\}$
5: Optimize the following functions to find Pareto optimality:
6: (1) $\max_{\forall\ f_k \in F'_{server}} f_{k_{FCMI}}$ (2) $\min_{\forall\ f_k \in F'_{server}} f_{k_{aFFMI}}$
7: Get the ranking of the features from the Pareto fronts.
8: $\forall_{i=1}^{q} Cl_i$ send $< rank, f_k >$ to all Cl_i iff $f_k \in F'_{server}$

Illustration: In Table V, the averaged FCMI and averaged aFFMI values for features f_1 to f_5 are pre-

sented, which were collected from clients. The server then performs a multi-objective optimization, aiming to maximize the FCMI scores while minimizing the aFFMI scores in order to determine the domination and dominated count for each feature. The difference between the domination and dominated count is used to rank the features. According to Table V, feature f_2 has domination count 4, i.e., its averaged FCMI is higher than f_1, f_3, f_4, and f_5. It also has a dominated count 0 because its average aFFMI is the lowest among all features. Therefore, feature f_2 is not dominated by other features. Consequently, the difference between domination count and dominated count ($c_{dom} - f_{dom}$) is 4, which is the highest among all features. Hence, f_2 is the most important feature, followed by f_2, f_1, f_4, and f_5. The difference between domination and dominated count for f_4 and f_5 are the same. But the domination count of f_4 is better than f_5. Hence, the rank of f_4 is higher than f_5 as it has more relevance.

TABLE V: Fed-MOFS - illustration of feature ranking through multi-objective optimization

Feature(s)	Averaged FCMI	Averaged aFFMI	Domination count (c_{dom})	Dominated count (f_{dom})	$c_{dom} - f_{dom}$	Rank
f_1	0.66	0.31	3	2	1	2
f_2	0.85	0.27	4	0	4	1
f_3	0.47	0.29	1	1	0	3
f_4	0.52	0.47	2	4	-2	4
f_5	0.34	0.31	0	2	-2	5

Remark 2. *Fed-MOFS utilizes a multi-objective optimization approximation algorithm to determine feature rankings. Therefore, Fed-MOFS never guarantees to find the exact optimal solutions but provides a set of non-dominated solutions (Pareto-optimal). But by prioritizing the domination count over the dominated count, we can get the unique rank of each feature.*

4) ***Global feature selection****:* Algorithm 4 describes the procedure for global feature selection after acquiring feature rankings from Fed-FiS or Fed-MOFS. Every client selects a common set of features ($F'' \subseteq F'_{Server}$) according to their individual rankings (Line 3), and then clients train the local model (θ_{Cl_i}) (Line 6). The server collects the local updates from the clients and computes the global model (ω) (Line 7). After T global rounds, the performance (ψ) of global model (ω) is evaluated (Line 8) and checks whether ψ achieves a specified threshold (δ) (Line 9). If $\psi \geq \delta$, then, F'' is the selected feature subset (Line 10); otherwise, an additional set of ε features from the next-ranked features is to be incorporated (Line 12).

Algorithm 4 Global feature selection

Input: F'_{Server} ▷ features according to their global ranks
Output: F'' ▷ global feature subset

1: **procedure** $Global_{FS}(F'_{Server})$
2: **while** $\psi \leq \delta$ **do**
3: All Cl_i selects a common feature subset $F'' \subseteq F'_{server}$ based on their global ranks.
4: **for** $t = 1 \ldots T$ **do** ▷ T is the global rounds
5: **for** All clients in parallel **do**
6: θ_{Cl_i} = Local_update($D(F'', S_{Cl_i}, C_{Cl_i})$)
7: ω = Global_update($\theta_1, \theta_2, \ldots, \theta_q$)
8: Compute accuracy of the global model (ψ)
9: **if** $\psi \geq \delta$ **then**
10: F'' is the selected features.
11: **else**
12: $| F'' | = | F'' | + \varepsilon$ where $| F'' | + \varepsilon \leq |F_{Server}|$

D. Computational complexity

The computational complexity of Algorithm 1 depends on the dimensionality of the input data and clustering algorithm. For a given data $D_i \in \mathbb{R}^{n \times d}$ at client Cl_i, the computational complexity of calculating FCMI and aFFMI is $O(d^2)$. Clustering is a np-hard problem, but taking heuristics can make the time complexity of it to linear. FCMI and aFFMI are two one-dimensional vectors of size d, so the time complexity for clustering is $(O(d \cdot \beta \cdot \gamma))$ where β is the number of clusters, and γ is the number of iterations. So the computation complexity of Algorithm 1 is $O(d^2) + O(d \cdot \beta \cdot \gamma)$.

In Algorithm 2 The computation complexity to calculate $S(f_k)$ (see Equation (8)) is $O(| F'_{server} |)$ where $| F'_{server} | \leq d$, so in worst case the computation complexity would be $O(d)$.

In Algorithm 3, the computation of F'_{server} requires $O((d.q)^2)$ because in worst-case in each client, the local feature subset contains all features. The global feature selection with multi-objective optimization uses NSGA-II [67] algorithm so the time complexity of it is $O(M. | F'_{server} |^2)$, where $| F'_{server} | \leq d$ and M is the number of objectives. Here, we optimize two objectives, so the worst-case time complexity is $O(d^2)$.

The computational complexity of the Algorithm 4 depends on the specific learning problem. In the case of strongly convex and smooth problems, the convergence of FedAvg on non-IID dataset is $O\left(\frac{1}{T}\right)$ [68], where T is the global rounds. If we have a dataset with d features, and selected features are ε, then the Algorithm 4 needs to be run a maximum of $\frac{d}{\varepsilon}$ times. Consequently, for strongly convex and smooth problems, the computational complexity of Algorithm 4 is $\frac{d}{\varepsilon} \cdot O\left(\frac{1}{T}\right)$.

E. Communication cost

To perform Algorithm 1, 2, and 3, both the client and server require a single communication round. A federated learning approach is utilised in the case of Algorithm 4. Consequently, the maximum number of required communication rounds can be expressed as $\frac{d}{\varepsilon} \cdot T$. Therefore, total communication cost $\leq 1 + \frac{d}{\varepsilon} \cdot T$.

V. Evaluation

We evaluated the performance of Fed-FiS and Fed-MOFS across multiple datasets. The details of the datasets and data statistics are given in Table VI. A detailed discussion of the datasets is in the supplementary copy. We evaluated both of our algorithms on both IID and non-IID data division.

TABLE VI: Datasets

Datasets	Instances	Features	Classes	Type
NSL-KDD99 [69]	125973	41	2	Intrusion detection
ACC [70]	284807	30	2	Credit card fraud transaction detection
Wine [71]	178	13	3	Wine quality
Vowel [72]	990	13	11	Speech recognition
Vehicle [73]	846	9	4	Vehicle ownership
Segmentation [74]	10695	10	4	Customer segmentation
WDBC [75]	569	30	2	Breast cancer
Ionosphere [76]	351	34	2	Radar scans
Hill-Valley [77]	606	100	2	Terrain
ISOLET [78]	7797	617	26	Speech data
Diabetes [79]	768	8	2	Diabetes diagnostics
IoT [80]	503910	28	17	Smart home data
Boston [81]	506	14	-	Regression dataset. Median Price of owner-occupied homes
California [82]	20640	9	-	Regression dataset, Predicting median house values in California
Synthetic	100000	200	25	Synthetic data

A. Experimental setup

The experimental setup is described in Table VII. Each client is responsible for selecting local features and the local training of the model. Meanwhile, the server estimates the global feature scores and ranks them. Moreover, server performs aggregation of the local models. Our local feature selection approach is similar to [1]. We employed the k-means clustering algorithm and set the number of clusters to 2 based on the highest silhouette score of the clusters. NSGA-II was employed to maximize relevance and minimize redundancy. Subsequently, we implemented federated forest and FedAvg algorithms, utilizing feature selection results from Fed-FiS and Fed-MOFS. In FedAvg, we are training a deep neural network (DNN). The description of the DNN is given in the supplementary copy. Additionally, we performed feature selection at the client level using RFE and ANOVA-based methods. This means each client independently used RFE and ANOVA for feature selection and sent the information regarding the selected features to the server. The server performs an intersection to find common features, and clients perform federated learning on the selected features. Furthermore, we compared our feature selection techniques with FSHFL [16], and Fed-mRMR [17], the state-of-the-art federated feature selection method. We ran all experiments 10 times and carried the mean of the results.

TABLE VII: Parameters description in federated feature selection

Parameter(s)	Value	Description
clients	5 to 100	Local feature selection, Learning local model
Server	1	Global feature selection, Learning global model
client's participation	10 to 100%	Partial or full participation of each client
Clustering algorithm	k-means	Cluster with nearest mean
Multi-objective optimization	NSGA-II	Non-dominated Sorting Genetic Algorithm
Feature selection state-of-the-art	4	ANOVA, RFE, FSHFL, Fed-mRMR
Learning algorithm	2	Federated Forest [18], FedAvg [3]
Performance metrics	7	Accuracy, Precision, Recall, and F1-Score, RMSE, MAE, Categorical cross-entropy loss.

B. Results and analysis

Here, we assessed the outcomes based on various aspects, including performance, scalability, stability, efficiency, and convergence of global model.

*1) **Performance evaluation**:* We evaluated the performance of Fed-FiS and Fed-MOFS for both IID and non-IID data division on multiple datasets representing classification and regression tasks.

a) **Performance evaluation with IID data divisions**: In the following experiments in Table VIII and IX, we considered the data with random distribution among 5 clients and ensured that each client has information from all classes (IID). After conducting training with Federated Forest (Table VIII) and training a deep neural network using federated averaging (Table IX) on both full feature sets and reduced feature sets produced by feature selection algorithms (RFE, ANOVA, FSHFL, Fed-mRMR, Fed-FiS, and Fed-MOFS), we observed in Table VIII, Fed-MOFS offers better accuracy and F1-Score than the state-of-the-art in 7 out of 12 datasets. Meanwhile, Fed-FiS performed better or equivalent when compared to Fed-MOFS in 4 datasets. For the Ionosphere dataset, FSHFL and Fed-mRMR both outperformed Fed-FiS and Fed-MOFS by 1%. In the Hill-Valley dataset, the performance of Fed-FiS and Fed-MOFS was similar, but their accuracy and F1-Scores were slightly lower (1% and 2%, respectively) compared to RFE. On the segmentation dataset, achieving the best performance requires the inclusion of all features. Fed-FiS and Fed-MOFS utilizes 78% and 89% of the feature space, respectively to achieves 48% accuracy and F1-Score, which is just 1% lower than the optimal result. In Table IX, we observed that Fed-MOFS outperformed the state-of-the-art in 9 datasets in terms of accuracy and F1-Scores. Fed-FiS outperformed state-of-the-art on 6 and 4 datasets in terms of accuracy and F1-Score respectively. For WDBC and ISOLET, both Fed-FiS and Fed-MOFS give similar accuracy and F1-Score. For WDBC and ISOLET, both Fed-FiS and Fed-MOFS produce the same F1-Score. In the supplementary copy, we provide a set of experiments using Random Forest where we found that Fed-MOFS performed well in 7 datasets in terms of validation accuracy and F1-Score. Fed-MOFS is pretty consistent in 9 datasets, excluding Vehicle, Segmentation and Diabetes datasets. We evaluated Fed-FiS, Fed-MOFS, and Fed-mRMR on regression tasks (Table X) using the Boston house price and California house price prediction dataset. Our results showed that when the feature subset contained 71% of the total features, Fed-FiS slightly outperformed Fed-MOFS on the Boston dataset. Conversely, Fed-MOFS performed equivalently with Fed-FiS on the California dataset by selecting 77% of the features. Both methods performed well compared to Fed-mRMR when selecting the same amount of features, while training a ridge regression model in a federated setup on both datasets.

TABLE VIII: Performance of federated forest on selected features (*mean* / *ratio of feature selected*)

Dataset	All Features		RFE		ANOVA		FSHFL		Fed-mRMR		Fed-FiS		Fed-MOFS	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Ionosphere	0.93/1.0	0.92	0.91/0.97	0.91	0.92/0.91	0.91	**0.93/0.54**	**0.93**	0.94/0.87	**0.94**	0.92/0.15	0.92	0.93/0.76	0.92
WDBC	0.95/1.0	0.95	0.95/0.52	0.95	0.96/0.52	0.95	0.95/0.26	0.94	0.94/0.45	0.94	**0.96/0.23**	0.95	0.96/0.39	**0.96**
Wine	0.97/1.0	0.97	0.98/0.92	0.98	0.98/0.85	0.97	0.86/0.46	0.83	0.98/0.77	0.98	**0.98/0.54**	**0.98**	0.98/0.77	0.98
Hill-Valley	0.52/1.0	0.52	**0.55/0.05**	**0.54**	0.52/0.90	0.52	0.49/0.21	0.49	0.51/0.35	0.51	0.53/0.25	0.52	0.53/0.10	0.52
Vowel	0.91/1.0	0.9	0.90/0.92	0.9	0.83/0.50	0.82	0.79/0.58	0.77	0.80/0.91	0.8	**0.91/0.92**	**0.91**	**0.91/0.83**	0.9
Vehicle	0.83/1.0	0.83	0.81/0.75	0.81	0.80/0.88	0.8	0.67/0.62	0.66	0.76/0.88	0.76	0.83/0.87	0.83	**0.84/0.87**	**0.84**
ACC	0.99/1.0	0.99	0.99/0.67	0.99	0.99/0.83	0.99	0.99/0.67	0.99	0.99/0.68	0.99	0.99/0.70	0.99	**0.99/0.63**	**0.99**
Segmentation	**0.49/1.0**	0.49	0.48/0.89	0.48	0.47/0.89	0.47	0.41/0.67	0.41	0.41/0.78	0.41	0.48/0.78	0.48	0.48/0.89	0.48
ISOLET	0.92/1.0	0.92	0.89/0.91	0.88	0.90/0.91	0.91	0.88/0.38	0.88	0.89/0.78	0.9	**0.92/0.78**	**0.92**	**0.92/0.78**	**0.92**
IoT	0.98/1.0	0.98	0.98/0.46	0.98	0.96/0.68	0.96	0.89/0.64	0.89	0.99/0.52	0.99	**0.98/0.25**	**0.98**	**0.98/0.25**	**0.98**
Diabetes	0.79/1.0	0.78	0.79/0.75	0.79	0.78/0.88	0.78	0.76/0.50	0.76	0.68/0.75	0.68	0.79/0.75	0.79	**0.80/0.37**	**0.8**
NSL-KDD99	0.99/1.0	0.99	0.99/0.78	0.99	0.99/0.90	0.99	0.99/0.71	0.99	0.99/0.78	0.99	0.99/0.84	0.99	**0.99/0.63**	**0.99**

TABLE IX: Performance of FedAvg on selected features (*mean* / *ratio of feature selected*)

Dataset	All Features		RFE		ANOVA		FSHFL		Fed-mRMR		Fed-FiS		Fed-MOFS	
	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy
Ionosphere	0.87/1	0.88	0.87/0.33	0.9	0.87/0.36	0.86	0.92/0.57	0.91	0.91/0.82	0.91	0.89/0.15	0.9	**0.94/0.21**	**0.91**
WDBC	0.94/1	0.94	0.94/0.32	0.95	0.95/0.48	0.95	0.92/0.25	0.93	0.95/0.57	0.95	**0.95/0.16**	**0.95**	**0.95/0.16**	**0.95**
Wine	0.93/1	0.92	0.87/0.53	0.94	0.93/0.61	0.96	0.93/0.38	0.93	0.96/0.84	0.96	0.91/0.46	**0.96**	0.93/0.38	0.94
Hill-Valley	0.5/1	0.51	0.53/0.1	0.5	0.5/0.35	0.51	0.51/0.21	0.5	0.50/0.35	0.44	0.51/0.1	0.51	**0.53/0.15**	**0.52**
Vowel	0.8/1	0.79	0.74/0.66	0.75	0.77/0.91	0.77	0.78/0.58	0.76	0.82/0.84	0.82	0.81/0.66	**0.83**	**0.82/0.66**	0.82
Vehicle	0.65/1	0.65	0.63/0.5	0.66	0.64/0.87	0.66	0.42/0.62	0.53	0.67/0.89	0.67	0.64/0.5	**0.67**	**0.67/0.67**	**0.67**
ACC	0.99/1	0.99	0.99/0.66	0.99	0.99/0.83	0.99	0.99/0.66	0.99	0.99/0.67	0.99	0.99/0.63	0.99	**0.99/0.46**	**0.99**
Segmentation	0.48/1	0.46	0.48/0.88	0.46	0.42/0.66	0.45	0.21/0.66	0.38	0.44/0.9	0.45	**0.48/0.66**	0.46	0.47/0.55	**0.46**
ISOLET	0.95/1	0.95	0.94/0.90	0.94	0.90/0.90	0.9	0.89/0.37	0.89	0.95/0.78	0.95	**0.95/0.77**	**0.95**	**0.95/0.77**	**0.95**
IoT	0.85/1	0.84	0.84/0.78	0.85	0.85/0.89	0.85	0.83/0.64	0.83	0.85/0.89	0.86	**0.87/0.71**	**0.88**	0.84/0.53	0.84
Diabetes	0.75/1	0.75	0.74/0.5	0.74	0.74/0.5	0.75	0.69/0.5	0.7	0.65/0.62	0.65	0.74/0.5	0.74	**0.76/0.5**	**0.77**
NSL-KDD99	0.99/1	0.99	0.99/0.76	0.99	0.99/0.81	0.99	0.99/0.71	0.99	0.99/0.85	0.99	0.99/0.68	0.99	**0.99/0.63**	**0.99**

TABLE X: Performance of FedAvg on the selected features for regression tasks

Dataset/ratio of feature selected	Fed-mRMR		Fed-MOFS		Fed-FiS	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Boston/0.71	9.04	6.1	8.98	6.08	**8.96**	**6.03**
California/0.77	0.83	0.60	**0.79**	**0.57**	**0.79**	**0.57**

b) **Performance evaluation with non-IID data divisions**: Here, we focused on the performance of Fed-FiS and Fed-MOFS on different non-IID setups given in Table XI. The table represents, each client has information from at most how many classes. We represent γ as a non-IID factor. For example, $\gamma = 0.2$ means each client has data from 2 out of 10 classes, while $\gamma = 0.8$ indicates that each client has data from 8 out of 10 classes. As γ approaches 1, the data distribution becomes more IID, with $\gamma = 1$ signifying that all clients have information about all classes, which is IID. Additionally, for each configuration, there are no overlapping samples among clients.

TABLE XI: Non-IID factors (γ) based on class information

Dataset	γ			
	0.2	0.5	0.8	1.0
	Number of classes per client			
IoT	4	9	14	17
ISOLET	5	13	21	26
Synthetic	8	13	20	25

- **Performance comparison with the state-of-the-art:** We compared the performance of Fed-FiS and Fed-MOFS with Fed-mRMR in Table XII for 5 different datasets with $\gamma = 0.8$ and 100 clients. In all experiments, we fixed the proportion of features selected for each dataset. For instance, 67% of features were chosen for the vehicle dataset, 80% for the segmentation dataset, 60% for the IoT dataset, 39% for the ISOLET dataset, and 60% for the synthetic dataset. We observed from the experiments in 3 out of the 5 datasets, Fed-MOFS outperformed the Fed-FiS and Fed-mRMR. Specifically, Fed-FiS showed superior results on the IoT datasets. For the segmentation dataset, Fed-mRMR slightly surpasses (1%) both Fed-FiS and Fed-MOFS in performance. The difference in results occurs due to the heterogeneity in the data.

TABLE XII: Performance of different federated feature selection algorithms with non-iid factor γ=0.8

Dataset/ratio of feature selected	Fed-mRMR		Fed-MOFS		Fed-FiS	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Vehicle/0.67	0.57	0.61	**0.71**	**0.75**	0.7	0.74
Segmentation/0.8	**0.46**	**0.49**	0.45	0.47	0.45	0.48
IoT/0.6	0.86	0.85	0.91	0.9	**0.92**	**0.91**
ISOLET/0.39	0.92	0.91	**0.95**	**0.95**	0.94	0.94
synthetic/0.6	0.94	0.94	0.98	0.98	0.97	0.97

- **Comparative analysis of Fed-FiS and Fed-MOFS for different non-IID factors (γ) and the selected feature subset:** To compare across models and datasets, we trained two global models us-

ing the federated learning algorithms FedAvg and Federated Forest on the features selected by Fed-MOFS and Fed-FiS, results are reported in Figure 3. During the training process, 100 clients participated fully for both the IoT and synthetic datasets, while 75 clients were involved for the ISOLET dataset, maintaining full participation throughout the entire learning period. Following observations are made from there, in the ISOLET dataset (Figure 3i to 3l), Fed-MOFS outperformed Fed-FiS. For a fixed γ, Fed-MOFS achieves better accuracy with fewer features compared to Fed-FiS. However, in the IoT dataset (Figure 3e) with $\gamma = 0.5$, Fed-FiS selected better features than Fed-MOFS. In the synthetic dataset (Figure 3a, Figure 3c), the features selected by Fed-MOFS offered better performance than those selected by Fed-FiS. These observations indicate that the performance of Fed-FiS and Fed-MOFS varies across different datasets and depends on the learning algorithm employed.

- **Relation between γ and cardinality of feature subset:** In Figure 3, when we keep γ fixed and increase the feature subset (20%, 40%, 60%, 80%), the performance of learning improves as the number of features increases. Similarly, the model's performance is enhanced when we keep cardinality of feature subset fixed and increase γ from 0.2 to 0.8. A similar trend is observed for Fed-FiS. Thus, very low values of γ and feature subset fail to obtain optimal results. To enhance performance, we need to raise either γ or cardinality of the feature subsets, or both.

*2) **Scalability**:* To evaluate the scalability of Fed-FiS and Fed-MOFS, we carried out experiments with 100 clients on IoT and 75 clients on ISOLET datasets, as depicted in Figure 4. We set γ to 0.2 and varied client participation using the scale $\Delta \in (0.1, 0.5, 0.8, 1.0)$, where $\Delta = 1.0$ indicates full client participation and $\Delta = 0.1$ implies 10% client participation. Our results showed that on the IoT dataset (Figure 4a), Fed-MOFS performed better with increased client participation, achieving optimal performance at $\Delta = 0.5$. Conversely, as shown in Figure 4b, Fed-FiS achieved the best results at $\Delta = 0.8$. For the experiments on ISOLET using Fed-MOFS (Figure 4c) and Fed-FiS (Figure 4d), increasing the value of Δ impacts the model's performance. Fed-MOFS achieved the highest accuracy and precision at $\Delta = 0.5$, and the highest recall and F1-score at $\Delta = 0.8$. For Fed-FiS, the model performed equivalently at $\Delta = 0.8$ and $\Delta = 1.0$, respectively.

From these observations, we can conclude that for Fed-MOFS, involving 50% participant in each global round is sufficient for achieving a generalized model. Similarly, for Fed-FiS, satisfactory performance is achieved with 80% client participation per global iteration. Therefore, both Fed-MOFS and Fed-FiS are effective with partial client participation.

a) **Effect of non-IID Factor (γ) on Partial Participation Factor (Δ)**: In these experiments, we fixed the client participation factor at $\Delta = 0.1$, meaning only 10% of the total clients participate in each global iteration. For varying values of γ (0.2, 0.5, and 0.8), increasing γ led to improved performance for both Fed-FiS and Fed-MOFS. Notably, there were no significant changes in performance for both the ISOLET (Figure 5a) and IoT (Figure 5b) datasets when γ increased from 0.5 to 0.8. For the ISOLET dataset (Figure 5a), Fed-MOFS outperformed Fed-FiS, while for the IoT dataset (Figure 5b), Fed-FiS performed better than Fed-MOFS.

When comparing Fed-MOFS and Fed-FiS based on client participation, Fed-FiS outperformed Fed-MOFS on IoT data (Figure 4a and Figure 4b). However, for the ISOLET datasets, Fed-MOFS (Figure 4c) showed better performance compared to Fed-FiS (Figure 4d). These observations suggest that the performance of the feature selection algorithm depends on the dataset, the non-IID factor (γ), and client participation (Δ).

3) Stability: To assess stability, we conducted a comparison between Fed-FiS and Fed-MOFS against traditional feature selection methods like RFE and ANOVA. We evaluated their stability by examining the validation accuracy (Figure 6) of the global model using varying numbers of features. From Figure 6a, we observed that the ranking of features by Fed-MOFS is better than others. Based on the outcomes displayed in Figure 6b, it can be observed that both Fed-FiS and Fed-MOFS have effectively identified the top 5 significant features. As a result, their model's accuracy has demonstrated a remarkable improvement compared to other models despite reducing the feature space by over 50%. This indicates that no crucial information has been lost during the reduction process.

4) Efficiency: We performed a comparative analysis of the efficiency of Fed-FiS, Fed-MOFS, and FSHFL in terms of the time (wall-clock time) required for feature selection. As shown in Figure 7, our findings revealed that both Fed-FiS and Fed-MOFS demonstrate nearly identical performance in selecting the feature subset, while FSHFL takes considerably longer to achieve the same task. Particularly, in certain datasets, the difference in feature selection time is quite significant. For instance, in the ACC dataset, Fed-FiS outperformed FSHFL by 26.68 seconds, and Fed-MOFS was 26.54 seconds faster. Similarly, for IoT datasets, Fed-FiS surpassed FSHFL by 15.04 seconds, and Fed-MOFS is 14.55 seconds faster than FSHFL. Additionally, when considering NSL-KDD99 datasets, Fed-FiS outperformed FSHFL by 23.1 seconds, and Fed-MOFS is 23 seconds faster than FSHFL.

Table XIII compares the validation accuracy vs. model size for Federated Forest. The number of estimators in random forests is predefined and remains constant; hence, to get a better understanding of the complexity of a forest, we focus on the maximum

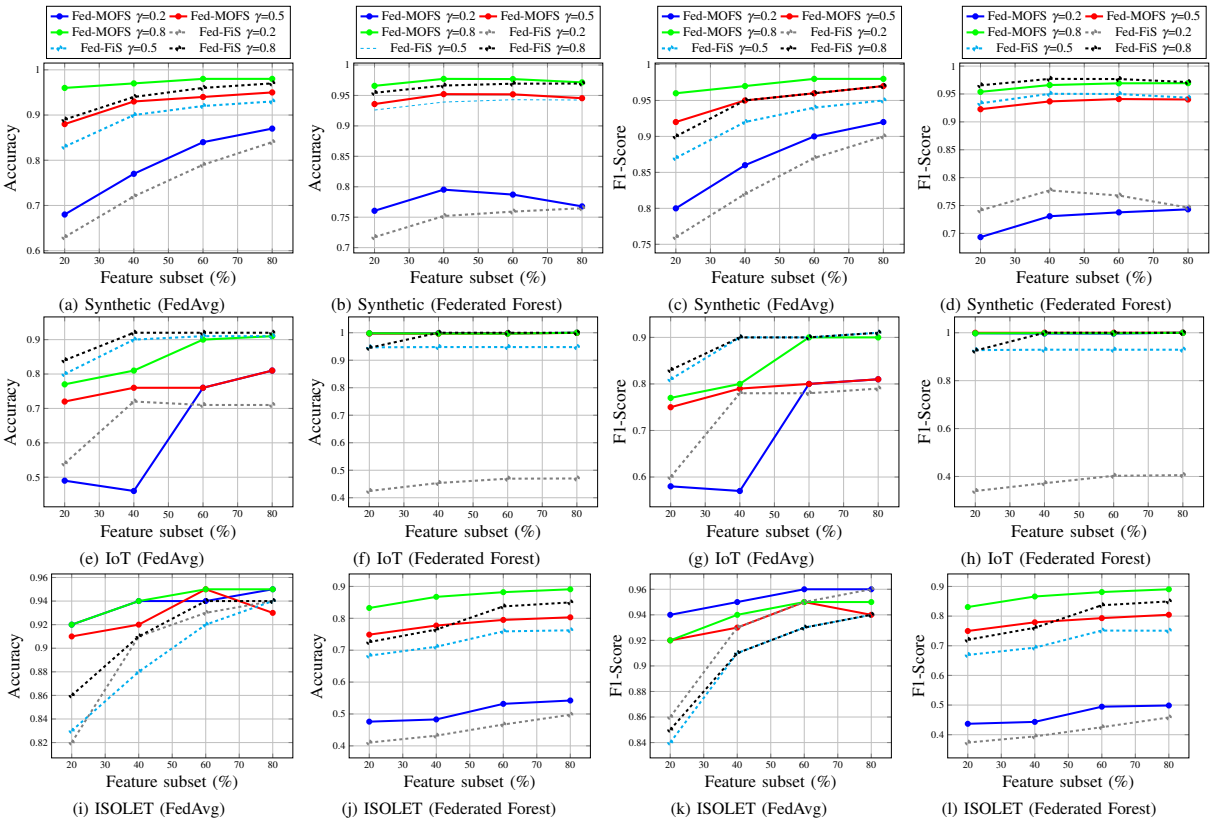


Figure 3: Performance comparison of Fed-FiS and Fed-MOFS on synthetic (Figure 3a, 3b, 3c, 3d), IoT (Figure 3e, 3f, 3g, 3h), and ISOLET (Figure 3i, 3j, 3k, 3l) datasets across different non-iid settings.

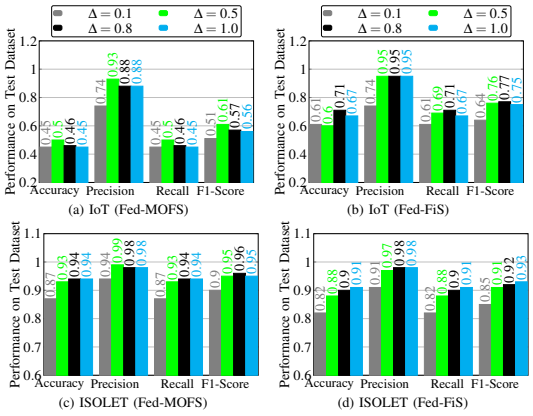


Figure 4: Performance of FedAvg with feature selection using Fed-MOFS on IoT (Figure 4a), and ISOLET (Figure 4c) datasets, compared to Fed-FiS on IoT (Figure 4b), and ISOLET (Figure 4d) datasets, for varying levels of client participation ($\Delta \in 0.1, 0.5, 0.8, 1.0$). The total number of clients is 100, with a non-IID factor (γ) of 0.2. The number of selected features is 17 for IoT (Figure 4a, 4b), and 240 for ISOLET (Figure 4c, 4d) datasets, using a DNN as the learning model.

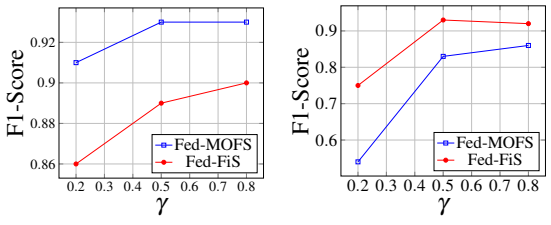


Figure 5: Performance comparison of Fed-FiS and Fed-MOFS for a fixed client participation ($\Delta = 0.1$) but varying non-IID factor $\gamma \in$ (0.2, 0.5, and 0.8)

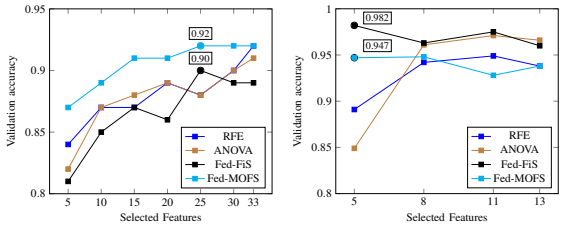


Figure 6: Validation accuracy vs. number of selected global features

depth of trees in a forest. We computed the ratio of accuracy and depth of the forest. For 4 (Ionosphere, WDBC, Segmentation and NSL-KDD99) out of 6 datasets, Fed-MOFS produced a higher ratio than Fed-FiS and no feature selection (No-FS). And in the other two datasets (Hill-Valley and Vehicle) Fed-FiS produced a higher ratio than No-FS.

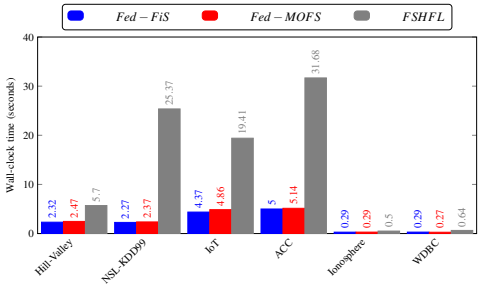
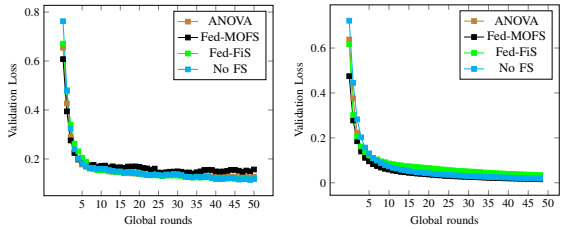


Figure 7: Wall-clock running time of federated feature selection algorithms

TABLE XIII: Comparison between the performance of federated forest and model size

Dataset	Algorithm	Accuracy/depth	Dataset	Accuracy/Depth
	No-FS	0.091		0.136
Ionosphere	**Fed-FiS**	0.092	**WDBC**	0.134
	Fed-MOFS	**0.105**		**0.162**
	No-FS	0.019		0.031
Hill-Valley	**Fed-FiS**	**0.021**	**Vehicle**	**0.033**
	Fed-MOFS	0.02		0.028
	No-FS	0.014		0.033
Segmentation	**Fed-FiS**	0.013	**NSL-KDD99**	0.030
	Fed-MOFS	**0.015**		**0.034**

5) Convergence: In the comparison depicted in Figure 8, we examined the convergence behavior of FedAvg on both NSL-KDD99 and WDBC datasets. Feature selection was carried out using three different methods: Fed-FiS, Fed-MOFS, and ANOVA. Additionally, we included experiments with the full dataset, where no feature selection was applied. Our observations indicate that the convergence pattern remains consistent whether feature selection is performed or not prior to the learning process. Thus, we can conclude that the inclusion of feature selection does not hinder the convergence of the learning model.



(a) NSL-KDD99 (b) WDBC

Figure 8: Global rounds vs. validation loss

VI. Concluding Remarks

In this paper, we proposed Fed-MOFS and performed extensive studies on Fed-FiS and Fed-MOFS. Both are feature selection methods designed exclusively for horizontal federated learning. We made an extensive empirical, computational and communication complexity analysis of Fed-FiS and Fed-MOFS. In terms of performance, Fed-MOFS and Fed-FiS together outperformed the state-of-the-art on both classification and regression tasks. In terms of stability, Fed-FiS and Fed-MOFS show high performance while reducing more than 50% of the feature space. Regarding efficiency, Fed-FiS and Fed-MOFS are at least 2× faster than FSHFL. We also observed feature selection doesn't influence the convergence of the model. Furthermore, we observed that both Fed-FiS and Fed-MOFS perform significantly well in the presence of non-IID data and partial client participation.

Federated feature selection has several potential applications, such as anomaly detection in human activity recognition, financial fraud detection, etc. In the future, we intend to implement federated feature selection that is capable of uncovering anomalies in human activity recognition with federated learning.

References

[1] S. Banerjee, E. Elmroth, and M. Bhuyan, "Fed-FiS: A novel information-theoretic federated feature selection for learning stability," in *Neural Information Processing: 28th International Conference (ICONIP), Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V*, pp. 480–487, Springer, 2021.

[2] S. Banerjee, X.-S. Vu, and M. Bhuyan, "Optimized and adaptive federated learning for straggler-resilient device selection," in *International Joint Conference on Neural Networks*, pp. 1–9, IEEE, 2022.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[5] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in IoV," *IEEE Trans. on Network and Service Management*, vol. 19, no. 3, pp. 3062–3075, 2022.

[6] C. Ren, T. Wang, H. Yu, Y. Xu, and Z. Y. Dong, "EFedDSA: An Efficient Differential Privacy-based Horizontal Federated Learning Approach for Smart Grid Dynamic Security Assessment," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2023.

[7] J. Li, T. Cui, K. Yang, R. Yuan, L. He, and M. Li, "Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development," *Sustainability*, vol. 13, no. 23, p. 13050, 2021.

[8] S. Banerjee, R. Misra, M. Prasad, E. Elmroth, and M. Bhuyan, "Multi-diseases classification from chest-x-ray: A federated deep learning approach," in *Advances in Artificial Intelligence: 33rd Australasian Joint Conference, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33*, pp. 3–15, Springer, 2020.

[9] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[10] M. A. Hall, *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[11] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371–6385, 2014.

[12] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "SCADA intrusion detection scheme exploiting the fusion of modified decision tree and Chi-square feature selection," *Internet of Things*, vol. 21, p. 100676, 2023.

[13] A. Arauzo-Azofra, J. L. Aznarte, and J. M. Benítez, "Empirical study of feature selection methods based on individual feature evaluation for classification problems," *Expert systems with applications*, vol. 38, no. 7, pp. 8170–8177, 2011.

[14] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[15] J. Krawczuk and T. Łukaszuk, "The feature selection bias problem in relation to high-dimensional gene data," *Artificial Intelligence in Medicine*, vol. 66, pp. 63–71, 2016.

[16] X. Zhang, A. Mavromatis, A. Vafeas, R. Nejabati, and D. Simeonidou, "Federated feature selection for horizontal federated learning in iot networks," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 10095–10112, 2023.

[17] J. Hermo, V. Bolón-Canedo, and S. Ladra, "Fed-mrmr: A lossless federated feature selection method," *Information Sciences*, vol. 669, p. 120609, 2024.

[18] Liu, Yang and Liu, Yingting and Liu, Zhijie and Liang, Yuxuan and Meng, Chuishi and Zhang, Junbo and Zheng, Yu, "Federated forest," *IEEE Transactions on Big Data*, vol. 8, no. 3, pp. 843–854, 2020.

[19] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Proceedings of the 24th international conference on Machine learning*, pp. 823–830, 2007.

[20] V. Feofanov, E. Devijver, and M.-R. Amini, "Wrapper feature selection with partially labeled data," *Applied Intelligence*, vol. 52, no. 11, pp. 12316–12329, 2022.

[21] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 907–948, 2020.

[22] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[23] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[24] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, 2019.

[25] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, 2011.

[26] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," in *International Conference on Engineering & MIS*, pp. 1–5, IEEE, 2016.

[27] J. Apolloni, G. Leguizamón, and E. Alba, "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments," *Applied Soft Computing*, vol. 38, pp. 922–932, 2016.

[28] R. Muthukrishnan and R. Rohini, "Lasso: A feature selection technique in predictive modeling for machine learning," in *IEEE international conference on advances in computer applications*, pp. 18–20, IEEE, 2016.

[29] S. Zhang, D. Cheng, R. Hu, and Z. Deng, "Supervised feature selection algorithm via discriminative ridge regression," *World Wide Web*, vol. 21, pp. 1545–1562, 2018.

[30] F. Amini and G. Hu, "A two-layer feature selection method using genetic algorithm and elastic net," *Expert Systems with Applications*, vol. 166, p. 114072, 2021.

[31] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Speech and Natural Language: Proceedings of a Workshop*, (Harriman, New York, February 23-26), 1992.

[32] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.

[33] D. Lin and X. Tang, "Conditional infomax learning: An integrated framework for feature extraction and fusion," in *Proc. of the 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Part I 9*, pp. 68–82, Springer, 2006.

[34] H. Yang and J. Moody, "Feature selection based on joint mutual information," in *Proceedings of international ICSC symposium on advances in intelligent data analysis*, vol. 23, Citeseer, 1999.

[35] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, 2015.

[36] M. S. Pathan, A. Nag, M. M. Pathan, and S. Dev, "Analyzing the impact of feature selection on the accuracy of heart disease prediction," *Healthcare Analytics*, vol. 2, p. 100060, 2022.

[37] C.-W. Chen, Y.-H. Tsai, F.-R. Chang, and W.-C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Systems*, vol. 37, no. 5, p. e12553, 2020.

[38] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," *Computers in biology and medicine*, vol. 112, p. 103375, 2019.

[39] S. M. Nagarajan, V. Muthukumaran, R. Murugesan, R. B. Joseph, M. Meram, and A. Prathik, "Innovative feature selection and classification model for heart disease prediction," *Journal of Reliable Intelligent Environments*, vol. 8, no. 4, pp. 333–343, 2022.

[40] V. Bolon-Canedo and B. Remeseiro, "Feature selection in image analysis: a survey," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2905–2931, 2020.

[41] F. Özyurt, "Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures," *The Journal of Supercomputing*, vol. 76, no. 11, pp. 8413–8431, 2020.

[42] N. Kozodoi, S. Lessmann, K. Papakonstantinou, Y. Gatsoulis, and B. Baesens, "A multi-objective approach for profit-driven feature selection in credit scoring," *Decision support systems*, vol. 120, pp. 106–117, 2019.

[43] S. K. Trivedi, "A study on credit scoring modeling with different feature selection and machine learning approaches," *Technology in Society*, vol. 63, p. 101413, 2020.

[44] N. Khalili and M. A. Rastegar, "Optimal cost-sensitive credit scoring using a new hybrid performance metric," *Expert Systems with Applications*, vol. 213, p. 119232, 2023.

[45] S. Ben Jabeur, N. Stef, and P. Carmona, "Bankruptcy prediction using the xgboost algorithm and variable importance feature engineering," *Computational Economics*, vol. 61, no. 2, pp. 715–741, 2023.

[46] H. Qian, B. Wang, M. Yuan, S. Gao, and Y. Song, "Financial distress prediction using a corrected feature selection measure and gradient boosted decision tree," *Expert Systems with Applications*, vol. 190, p. 116202, 2022.

[47] B. Yuan, S. Ge, and W. Xing, "A federated learning framework for healthcare iot devices," *arXiv preprint arXiv:2005.05083*, 2020.

[48] A. U. Haq, A. Zeb, Z. Lei, and D. Zhang, "Forecasting daily stock trend using multi-filter feature selection and deep learning," *Expert Systems with Applications*, vol. 168, p. 114444, 2021.

[49] M. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243–271, 2016.

[50] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowledge-Based Systems*, vol. 71, pp. 322–338, 2014.

[51] M. S. El Sayed, N.-A. Le-Khac, M. A. Azer, and A. D. Jurcut, "A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 4, pp. 1862–1880, 2022.

[52] M. Nakashima, A. Sim, Y. Kim, J. Kim, and J. Kim, "Automated feature selection for anomaly detection in network traffic data," *ACM Trans. on Management Information Systems*, vol. 12, no. 3, pp. 1–28, 2021.

[53] A. B. Rashid, M. Ahmed, L. F. Sikos, and P. Haskell-Dowland, "Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection," *ACM Trans. on Management Information Systems*, vol. 13, no. 3, pp. 1–39, 2022.

[54] L. Wang, Y. Wang, and Q. Chang, "Feature selection methods for big data bioinformatics: a survey from the search perspective," *Methods*, vol. 111, pp. 21–31, 2016.

[55] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[56] J. Li, S. Fong, R. K. Wong, R. Millham, and K. K. Wong, "Elitist binary wolf search algorithm for heuristic feature se-

lection in high-dimensional bioinformatics datasets," *Scientific reports*, vol. 7, no. 1, p. 4354, 2017.
[57] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, "Centralized vs. distributed feature selection methods based on data complexity measures," *Knowledge-Based Systems*, vol. 117, pp. 27–45, 2017.
[58] T. K. Ho, M. Basu, and M. H. C. Law, "Measures of geometrical complexity in classification problems," in *Data complexity in pattern recognition*, pp. 1–23, Springer, 2006.
[59] Y. Gui, "ADAGES: adaptive aggregation with stability for distributed feature selection," in *Proceedings of the ACM-IMS on Foundations of Data Science Conference*, pp. 3–12, 2020.
[60] L. Kong, W. Qu, J. Yu, H. Zuo, G. Chen, F. Xiong, S. Pan, S. Lin, and M. Qiu, "Distributed feature selection for big data using fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 846–857, 2019.
[61] M. Soheili and A. M. Eftekhari-Moghadam, "Dqpfs: Distributed quadratic programming based feature selection for big data," *Journal of Parallel and Distributed Computing*, vol. 138, pp. 1–14, 2020.
[62] S. Zadeh, M. Ghadiri, V. Mirrokni, and M. Zadimoghaddam, "Scalable feature selection via distributed diversity maximization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
[63] P. Cassará, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," *IEEE Trans. on Vehicular Technology*, vol. 71, no. 9, pp. 9937–9950, 2022.
[64] Y. Hu, Y. Zhang, D. Gong, and X. Sun, "Multiparticipant federated feature selection algorithm with particle swarm optimization for imbalanced data under privacy protection," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1002–1016, 2022.
[65] Y. Qin and M. Kondo, "Federated learning-based network intrusion detection with a feature selection approach," in *International Conference on Electrical, Communication, and Computer Engineering*, pp. 1–6, IEEE, 2021.
[66] B. Swingle, "Rényi entropy, mutual information, and fluctuation properties of fermi liquids," *Physical Review B*, vol. 86, no. 4, p. 045109, 2012.
[67] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[68] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.
[69] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, IEEE, 2009.
[70] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.
[71] S. Aeberhard and M. Forina, "Wine." UCI Machine Learning Repository, 1991. acc.: 2024-06-09.
[72] P. Turney, "Deterding Vowel Recognition Data." OpenML, 2014. acc.: 2024-06-09.
[73] Various, "Vehicle Dataset." Kaggle, 2020. acc.: 2024-06-09.
[74] K. Suresh, "Customer Segmentation Classification." Kaggle. acc.: 2024-06-09.
[75] W. W. et al., "Breast Cancer Wisconsin (Diagnostic)." UCI ML Repository, 1995. acc.: 2024-06-09.
[76] S. V. et al, "Ionosphere." UCI ML Repository, 1989. acc.: 2024-06-09.
[77] G. Lee and O. Franz, "Hill-Valley." UCI ML Repository, 2008. acc.: 2024-06-09.
[78] C. Ron and F. Mark, "ISOLET." UCI MLRepository, 1994. acc.: 2024-06-09.
[79] N. I. of Diabetes, Digestive, and K. Diseases, "Diabetes Dataset." Kaggle, 1990. acc.: 2024-06-09.
[80] T. S. Anttal, "Smart Home Dataset with Weather Information." Kaggle, 2019. acc.: 2024-06-09.
[81] G. Koe, "Boston house price data," 2023. acc.: 2024-06-09.
[82] C. Nugent, "California housing prices," 2017. acc.: 2024-06-09.
[83] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

Sourasekhar Banerjee (Student Member, IEEE) received the M.Tech. degree in Computer Science and Engineering from the University of Calcutta, India, in 2018 and currently pursuing PhD in Computer Science in the Department of Computing Science at Umeå University, Sweden. His PhD study is funded by WASP (Wallenberg AI, Autonomous Systems and Software Program), Sweden. His research interests include federated learning and its applications.

Devvjiit Bhuyan received the Bachelor of Technology (B.Tech.) in Electronics and Communication Engineering from Tezpur University, India, in 2023 and is currently involved in multiple research projects with Cyber Analytics and Learning Group at Umeå University. His research interests include federated learning and security in machine learning.

Erik Elmroth (Member, IEEE) is Professor in Computing Science at Umeå University, where he has been department head and deputy head for thirteen years and Deputy Director for a national supercomputer center for another thirteen years. He has established the Umeå University research on distributed systems. Elmroth is a member of the management organization of the 550 M Euro Wallenberg AI, Autonomous Systems and Software Program (WASP) and member of the program management group for the eSSENCE Strategic Research Program. Previously, he was principal investigator for Cloud Control, which was the second largest project ever funded by the Swedish Research Council. Elmroth is member of the Swedish Royal Academy for Engineering Sciences. Previously, he was the Chair of the Board of the Swedish National Infrastructure for Computing; Chair of the Swedish Research Council's (VR's) expert group on e-science infrastructures, and member of VR's Council for Research Infrastructures (RFI). He has been appointed by the Nordic Council of Ministers for developing two international research strategies. He received the Nordea Scientific Award 2011. Pre-historic highlights include being co-winner of the SIAM Linear Algebra Prize 2000, for the most outstanding linear algebra publication world-wide (in any journal) during the preceding 3-year period. International experiences include a year at NERSC, Lawrence Berkeley National Laboratory and one semester at the MIT. He is founder and chair of the Control Workshops series. Elmroth has co-founded Elastisys AB, an expert on security and regulatory compliance in the cloud-native ecosystem.

Monowar Bhuyan (Senior Member, IEEE) earned his Ph.D. degree in computer science and engineering from Tezpur University, Assam, India, in 2014. Currently, Dr. Bhuyan is an Assistant Professor in the Department of Computing Science at Umeå University, Sweden, since January 2020 and leading *Cyber Analytics and Learning Group*, part of the Autonomous Distributed Systems Lab. Prior to this, he worked at various academic institutions, including the Nara Institute of Science and Technology in Japan, Assam Kaziranga University in India, and Umeå University, Sweden, at different levels, from a Junior Scientist to an Associate Professor between January 2009 and December 2019. Dr. Bhuyan has an extensive publication record, with over 100 papers in the leading international journals and conference proceedings and has written an

advanced textbook with Springer. His experience leading/co-leading research projects attracted over 35 MSEK from national, European Commission, and international grant agencies. His research interests span machine learning, anomaly detection, systems and AI security, and distributed systems.

APPENDIX

Here, we discuss the datasets and the learning models in detail. After that, we have one additional result to show the superiority of the performance of Fed-FiS and Fed-MOFS over the state-of-the-art.

A. Dataset

In the empirical studies, we have experimented with Fed-FiS, Fed-MOFS, and state-of-the-art model FSHFL, ANOVA, and RFE on 12 datasets.

1) NSL-KDD99: The NSL-KDD [69] dataset is a refined version of the original KDD Cup 99 dataset for evaluating computer network intrusion detection systems. The KDD Cup 99 dataset was the benchmark for assessing network-based anomaly detection methods. The dataset contains 41 features. These features are derived from network traffic data and include various types of data, such as basic features of network connections, content features, and traffic features based on a two-second temporal window. The dataset has two main classes: normal (benign) and attack. However, the attack class is further divided into four major categories of network attacks, which are Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probing. In this paper, we have only two classes: Benign and attack.

2) Annonymized Credit Card Fraud (ACC): ACC dataset [70] is commonly used in machine learning and data science to detect fraudulent credit card transactions. The dataset is anonymized for privacy reasons. Sensitive personal information is either removed or transformed in a way that individual transactions cannot be traced back to individual cardholders. The dataset typically includes a mix of numerical features transformed using techniques like Principal Component Analysis (PCA). The datasets have 28 feature variables ($V_1, V_2, \ldots, V28$) and 2 classes. Class 1 represents fraudulent transactions, and 0 represents benign transactions. There are 492 frauds out of 284,807 transactions. The dataset is highly unbalanced. The fraudulent transactions account for 0.172% of all transactions. We applied SMOTE [83] to that unbalanced dataset to balance it, and after that, we distributed the data to clients.

3) Wine: Wine [71] is a classic dataset used in multivariate statistics and machine learning for classification tasks. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The dataset characteristic is Tabular. It has 12 features such as Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. The features are real and integer. The dataset has 178 samples.

4) Vowel: The primary goal of the Vowel [72] dataset is to classify different spoken vowels. This dataset has 13 features, where the number of numeric features is 10 and 3 symbolic features. 990 instances and 11 classes.

5) Vehicle: The Vehicle dataset [73], focuses on types of car ownership. It comprises 846 samples and 9 attributes, and the target is categorized into 4 distinct classes. The dataset contained non-numeric data in features, which has been converted into categorical format for analysis. For instance, the ownership category has four types: 'First owner,' 'Second owner,' 'Third owner,' and 'Fourth or Above owner'. These have been encoded as 0, 1, 2, and 3, respectively.

6) Segmentation: This is a customer segmentation classification [74]. It has 10 features such as ID, Gender, Ever_married, Age, Graduated, Profession, Work_Experience, Spending_Score, Family_Size, and Var_1. The target variable is Segmentation. It has four classes. Similar to the Vehicle dataset, it has non-numeric data in some features (For example, Gender, Profession, etc.). Those are converted into categorical.

7) WDBC: This is a Wisconsin Diagnostic Breast Cancer dataset [75]. The dataset characteristics are multivariate. The feature type is real. The dataset has 30 features and 569 samples. The dataset is useful for classification tasks. The number of classes is two (malignant and benign).

8) Ionosphere: This is a classification of radar returns from the ionosphere [76]. The dataset characteristics are multivariate. The dataset has 34 features where the features are integer or real. The entire dataset has 351 samples.

9) Hill_valley: This dataset [77] has 100 features. All have floating-point values. The number of samples is 606. This dataset is used for 2 class classification problems. The class is represented in binary $\{0,1\}$. 0 means valley and 1 means hill.

10) ISOLET: This dataset [78] is useful for classification problems. The goal is to predict which letter name was spoken. Therefore, the target classes are 26. The characteristics of the dataset are multivariate. The number of features is 617, and each feature has real numbers. The total samples are 7797.

11) Diabetes: The objective of this dataset [79] is to predict whether the patient has diabetes or not. This is a binary class classification dataset. The number of features is 8, and the number of samples is 768. All the features have a numeric value. The dataset has the following features: Number of times pregnant, plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in $kg/(heightinm)^2$), Diabetes pedigree function, Age (years).

12) IoT: This dataset contains smart home data [80]. It has 503910 samples. 28 Features and 18 classes. This dataset is useful for classification problems.

13) Synthetic data: We have used a synthetic dataset with 100,000 samples and 200 features specifically

for our experiments with non-iid data. This dataset contains 35 truly informative features, and 65 features that are linearly dependent on the 35. The remaining 100 are purely noise aimed to reduce the learnability of models. A good Feature selection algorithm will be able to filter this and provide high-quality data to a model. All the data points belong to one of 25 classes With a randomly initialized class imbalance adding to the challenge of non-iidness.

For a given iid ratio (γ), a client can have samples comprising of not more than 20% of the total number of class labels if $\gamma = 0.2$. For example, if the total number of classes in the dataset is 25, a single client will have not more than floor(25*0.2) = 5 class labels. The first client may be assigned samples having the first five class labels, the second client will be assigned samples having another set of 5 class labels, and so on. Multiple checks and balances ensure that no sample is repeated in more than one client, all the samples are utilized in the distribution process, no single client has more than (num_classes*γ) number of class labels, number of samples for each client is random.

B. Learning model

1) Neural Network: We created a neural network that has pass through 3 dense layer. The first hidden layer is a dense layer with 128 parameters. The second hidden layer is another dense layer with 64 parameters. The third hidden layer is a dense layer with 32 parameters. In all three hidden layer we have a ReLU activation function. We have an output layer where the number of parameters is equal to number of classes. The activation function of the output layer is either a sigmoid or a softmax, depending on whether the problem is binary classification or multi-class classification.

C. Performance analysis

After conducting training with distributed Random-Forest using both complete feature sets and reduced feature sets produced by feature selection algorithms (RFE, ANOVA, FSHFL, Fed-FiS, and Fed-MOFS), we observed (in Table XIV) for most of the larger datasets, excluding Isolet (i.e. WDBC, HillValley, ACC, IoT, and NSL-KDD99 datasets), Fed-MOFS produced a smaller feature subset than others while maintaining a high test-accuracy. In terms of F1-Score (in Table XV), Fed-MOFS is pretty consistent across most datasets (excluding Vehicle, Segmentation, and Diabetes datasets), providing the highest F1-Score at minimal features.

TABLE XIV: Test Accuracies of the model trained using Random-Forest Algorithm (*mean* $\pm$ *std* / *ratio of feature selected*)

Dataset	All Features	RFE	ANOVA	FSHFL	Fed-FiS	Fed-MOFS
Ionosphere	0.86±0.02/1	0.86±0.05/0.33	0.86±0.03/0.36	0.82±0.01/0.54	**0.89±0.02/0.15**	0.88±0.02/0.21
WBDC	**0.94±0.01/1**	**0.94±0.02/0.25**	**0.94±0.02/0.74**	**0.94±0.01/0.25**	**0.94±0.01/0.22**	**0.94±0.01/0.19**
WINE	0.94±0.02/1	0.94±0.03/0.53	0.94±0.03/0.61	0.91±0.01/0.46	**0.95±0.03/0.53**	**0.95±0.02/0.46**
Hill valley	0.51±0.03/1	0.50±0.02/0.45	0.51±0.03/0.65	0.51±0.00/0.21	0.51±0.01/0.25	**0.55±0.02/0.05**
Vowel	**0.80±0.02/1**	**0.80±0.02/0.83**	**0.80±0.02/0.91**	0.78±0.00/0.58	0.79±0.02/0.91	0.79±0.03/0.66
Vehicle	**0.81±0.01/1**	0.80±0.01/0.75	0.79±0.01/0.87	0.65±0.00/0.62	0.80±0.01/0.87	0.79±0.02/0.87
ACC	**0.99±0.00/1**	**0.99±0.00/0.66**	**0.99±0.00/0.7**	**0.99±0.01/0.66**	**0.99±0.00/0.7**	**0.99±0.00/0.56**
Segmentation	**0.43±0.00/1**	**0.43±0.01/0.88**	**0.43±0.01/0.88**	0.38±0.00/0.66	0.41±0.01/0.77	0.42±0.01/0.88
ISOLET	**0.90±0.00/1**	**0.90±0.00/0.64**	**0.90±0.00/0.77**	0.83±0.00/0.37	**0.90±0.00/0.77**	**0.90±0.00/0.64**
IoT	0.97±0.00/1	**0.98±0.00/0.32**	0.96±0.00/0.67	0.89±0.01/0.64	**0.98±0.00/0.25**	**0.98±0.00/0.25**
Diabetes	0.76±0.01/1	0.76±0.01/0.5	0.75±0.02/0.87	0.69±0.02/0.5	**0.77±0.01/0.75**	0.76±0.01/0.62
NSL KDD99	**0.99±0.00/1**	**0.99±0.00/0.81**	**0.99±0.00/0.86**	0.98±0.01/0.71	**0.99±0.00/0.84**	**0.99±0.00/0.65**

TABLE XV: F1-Scores of the model trained using Random-Forest Algorithm (*mean* $\pm$ *std* / *ratio of feature selected*)

Dataset	All Features	RFE	ANOVA	FSHFL	Fed-FiS	Fed-MOFS
Ionosphere	0.87±0.01/1	0.86±0.05/0.33	0.87±0.02/0.36	0.83±0.01/0.54	0.89±0.02/0.15	**0.94±0.02/0.21**
WBDC	0.94±0.01/1	0.94±0.02/0.25	0.94±0.01/0.74	0.94±0.01/0.25	**0.95±0.01/0.22**	0.94±0.01/0.19
WINE	0.95±0.01/1	0.95±0.03/0.53	0.95±0.03/0.61	0.93±0.01/0.46	**0.96±0.02/0.53**	**0.96±0.02/0.46**
Hill Valley	0.52±0.03/1	0.5±0.02/0.45	0.52±0.03/0.65	0.52±0.00/0.21	0.52±0.01/0.25	**0.56±0.02/0.05**
Vowel	**0.82±0.02/1**	**0.82±0.02/0.83**	**0.82±0.01/0.91**	0.78±0.00/0.58	0.81±0.02/0.91	0.80±0.03/0.66
Vehicle	**0.81±0.01/1**	0.80±0.01/0.75	0.79±0.01/0.87	0.65±0.00/0.62	0.80±0.01/0.87	0.79±0.02/0.87
ACC	**0.99±0.00/1**	**0.99±0.00/0.66**	**0.99±0.00/0.7**	**0.99±0.01/0.66**	**0.99±0.0/0.7**	**0.99±0.0/0.56**
Segmentation	**0.43±0.00/1**	**0.43±0.01/0.88**	**0.43±0.01/0.88**	0.38±0.00/0.66	0.41±0.00/0.77	0.42±0.01/0.88
ISOLET	**0.91±0.00/1**	**0.91±0.00/0.64**	**0.91±0.00/0.77**	0.84±0.00/0.37	**0.91±0.00/0.77**	**0.91±0.00/0.64**
IoT	0.97±0.00/1	**0.98±0.00/0.32**	0.96±0.00/0.67	0.89±0.01/0.64	**0.98±0.00/0.25**	**0.98±0.00/0.25**
Diabetes	0.77±0.01/1	0.76±0.03/0.5	0.76±0.02/0.87	0.69±0.02/0.5	**0.78±0.01/0.75**	0.76±0.02/0.62
NSL KDD99	**0.99±0.00/1**	**0.99±0.00/0.81**	**0.99±0.00/0.86**	**0.98±0.01/0.71**	**0.99±0.00/0.84**	**0.99±0.00/0.65**