Federated Feature Selection for Horizontal Federated Learning in IoT Networks

Xunzheng Zhang, *Student Member, IEEE*, Alex Mavromatis, *Member, IEEE*, Antonis Vafeas, *Member, IEEE*, Reza Nejabati, *Senior Member, IEEE*, and Dimitra Simeonidou, *Fellow, IEEE*

***Abstract*—Under horizontal federated learning (HFL) in the Internet of Things (IoT) scenarios, different user data sets have significant similarities on the feature spaces, the final goal is to build a high-performance global model. However, not all features are great contributors when training the global HFL model, some features even impair the HFL. Besides, the curse of dimension will delay the training time and cause more energy consumption (EC). In this case, it is critical to remove irrelevant features from the local and select the useful overlapping features from a federated global perspective. In addition, the uncertainty of data being labeled and the nonindependent and identically distributed (non-IID) client data should also consider. This article introduces an unsupervised federated feature selection approach (named FSHFL) for HFL in IoT networks. First, a feature relevance outlier detection method is applied to the HFL participants to remove the useless features, which combines with the improved one-class support vector machine. Besides, a feature relevance hierarchical clustering (FRHC) algorithm is proposed for HFL overlapping feature selection. Experiment results on four IoT data sets show that the proposed methods can select better-federated feature sets among HFL participants, thus improving the performance of the HFL system. Specifically, the global model accuracy improves up to 1.68% since fewer irrelevant features. Moreover, FSHFL can lower the average training time as high as 6.9%. Finally, when the global model gets the same test accuracy, FSHFL can decrease the average EC of training the model by approximately 2.85% compared to federated average and roughly 68.39% compared to Fed-SGD.**

***Index Terms*—Data cleaning, feature selection (FS), federated learning (FL), Internet of Things (IoT), unsupervised machine learning (ML).**

I. Introduction

FEATURE selection (FS) is one of the most vital preprocessing steps in data mining and machine learning (ML). Through the process of detecting relevant features and removing irrelevant, redundant, or noisy data, FS improves prediction accuracy, speeds up data mining algorithms, and thus decreases the learning cost of systems [1], [2], [3]. Compared with other dimension-reduction techniques, such as principal component analysis (PCA), FS approaches do not change the original features but merely select a subset of features that are most useful. Without losing the original semantics of the dimensions, FS provides the advantage of interpretability by a domain expert. In recent years, the amount of high-dimensional data that exists in the network has greatly increased, especially in the Internet of Things (IoT) networks. With 75 billion IoT devices expected by 2025 [4], [5], [6], [7], IoT networks integrate a large number of pervasive objects that continuously generate a digital depiction of the physical world.

Data collected by the underlying IoT networks are the foundation of ML [8] methods which often need to learn latent patterns and empirical models from a large amount of data, attempting to give optimal regression or classification results. The increase in granularity and continuous evolution of novel data capture approaches leads to an increase in the input space dimension. In this case, if the ML is adopted in IoT networks without considering FS, the dimension of the input space will be large (i.e., data sets with hundreds or thousands of features) and nonclean (noisy or irrelevant features), also degrading the performance of learning models. Likewise, an efficient and robust FS method that eliminates noisy, irrelevant data, while maintaining the discriminating power of the data, is critical in the IoT system. Therefore, FS becomes very necessary for ML tasks when facing high-dimensional data in the IoT era. Nowadays, FS algorithms have been widely used in different IoT scenarios [9], [10], [11].

With the popularization of data technology and the coverage of IoT networks, data learning paradigms are evolving into distributed data learning. For example, data describing the same learning task in real life may be distributed and stored in different IoT devices (called participants or clients), such as IoT network traffic data and customer data stored by banks to name a few. In this case, traditional FS approaches are facing challenges. If each participant only uses their own data to conduct feature learning, it will inevitably affect the classification accuracy and robustness of the FS results, causing the problem of FS bias [12]. Besides, each participant will have its own data characteristics to some extent, which are nonindependent and identically distributed (non-IID). Therefore, due to privacy restrictions, participants are often unwilling or unable to directly share the data they hold, causing data silos. To solve the data isolation problem, federated learning (FL) [13], [14], [15], [16], [17], [18], [19] becomes a promising research area.

Manuscript received 6 September 2022; revised 4 December 2022; accepted 9 January 2023. Date of publication 16 January 2023; date of current version 23 May 2023. This work was supported in part by the U.K. DCMS funded Project Future RAN: O-RANOS; in part by the U.K. DCMS 2022 UK5G Proteus Research Project; and in part by the China Scholarship Council. *(Corresponding author: Xunzheng Zhang.)*

The authors are with the High Performance Networks Group, Smart Internet Lab, School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, Faculty of Engineering, University of Bristol, BS8 1QU Bristol, U.K. (e-mail: xunzheng.zhang@bristol.ac.uk).

Digital Object Identifier 10.1109/JIOT.2023.3237032

2327-4662 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

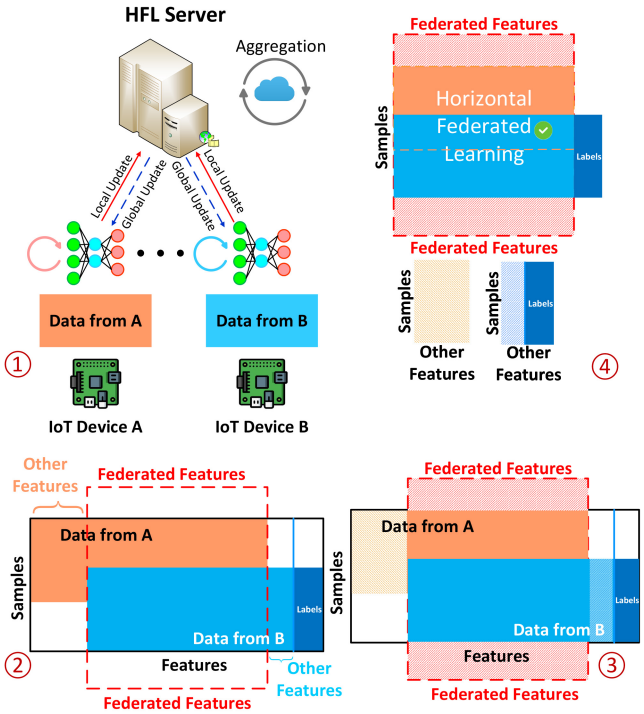

Fig. 1. Four steps of selecting federated features for HFL.

From the clients' data overlapping perspective on sample space or feature space, FL can be categorized as horizontal FL (HFL), vertical FL (VFL), and federated transfer learning (FTL) [17]. Admittedly, FL has some inherent advantages like sharing the jointly trained model for local prediction, moving the model training to the local, faster local response and the FL server only works for models aggregation (e.g., federated average (Fed-Avg) [20], q-FedAvg [21], and FedProx [22]) while keeping the users' data no leakage. However, applying FS methods to FL as federated FS (FFS) faces challenges, such as data cleaning, feature relevance, and unsupervised FS approaches. These issues will be presented and addressed in Sections II–IV below. In particular for HFL, without FS, the high-dimensional significant overlaps on the feature spaces [19] will cause malicious feature interference, bad model accuracy, prolonged training time, and increased energy consumption (EC). So how to find the proper overlapping federated features part (Fig. 1) for the HFL is of great significance.

Further to the challenges in FFS, research is progressing on methods to systematically address those open questions. Hu et al. [23] gives an FFS method based on particle swarm optimization (PSO), Yao et al. [24] showed a feature fusion approach, Cassarà et al. [25] proposed a mutual information (MI)-based FS algorithm. Banerjee et al. [26] displayed an information-theoretic FFS approach, Qin and Kondo [27] used the greedy algorithm to select features, and a secure FFS protocol is proposed by [28]. Meanwhile, we cannot guarantee all the data is labeled. In the unsupervised FS, feature relevance can be evaluated by feature dependence or similarity, like structured optimal graph FS (SOGFS) [29], robust joint graph sparse coding [30], and adaptive neighbors FS [31]. Although these FS algorithms significantly improve the accuracy of obtained feature subset, they do not consider all the problems which should be solved for FFS in HFL: 1) data is not cleaned before the formal FS approach; 2) different features have relevance in the data set under the non-IID condition; and 3) the data of each participant may not always be labeled. There is still no effective way to deal with the FFS problem for IoT networks under the HFL.

To address the above-mentioned challenges, this article combines the idea of outlier detection and unsupervised methods, proposes a feature average relevance one-class support vector machine (FAR-OCSVM) algorithm for outlier feature detection, and a feature relevance hierarchical clustering (FRHC) algorithm in order to gather representative features. After the intersection operation on optimized feature sets from different HFL participants, an FFS method for HFL (we summarily name it FSHFL) is given. Here, are the advantages of the proposed FSHFL.

1) Removing those outlier features irrelevant to overall federated feature expression from local.
2) Clustering local features to highlight relevant representative feature clusters.
3) Collecting and intersecting the optimized feature sets.

FSHFL also gives feedback to local in order to focus the user's attention on the most relevant features that can achieve high-performance HFL.

A. Contribution

The main contributions of this article can be summarized as follows.

1) Modeling: Aiming to handle the performance degradation caused by high-dimensional data, this work first models a feature optimization selection architecture of HFL in IoT networks. The architecture consists of three parts: 1) local unrelated feature cleaning; 2) local correlated feature clustering; and 3) global FFS.

2) Algorithms:

1) *A Feature Detection Algorithm:* An improved one-class support vector machine (OCSVM) algorithm, called FAR-OCSVM is designed for irrelevant feature detection. It can be used to detect features that do not have enough globally representative attributes (e.g., The blue and yellow shadow area in Fig. 1 step 4). The detected features by FAR-OCSVM relatively have less contribution to the final global result.
2) *A Feature Clustering Algorithm:* A novel FRHC algorithm is designed to capture the "essence" of the local data. By collecting local features and clustering them, a family of globally representative feature clusters can be highlighted (by the definition of HFL). Meanwhile, features in each cluster have strong relationships with each other (the main idea of the proposed FRHC algorithm). As a result, more features in one cluster, this feature cluster is more representative of HFL, also has the most prominent contribution to the HFL process.
3) *A Global FFS Method:* After the intersection operation from gathered local optimal HFL feature sets, federated

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

features are identified. These federated features will then be fed back to local users, focusing the user's attention on the most relevant features to the global representation, without harming other local features or data leakage. Due to the uncertainty of labeled data or not, the above-mentioned methods are unsupervised.

3) Simulation Evaluation: The designed algorithms are evaluated by IoT network data sets via simulation. On the one hand, we use four state-of-the-art IoT network data sets: 1) DEFT IoT network data set [32]; 2) Bot-IoT data set [33]; 3) Anonymized credit card transactions (ACCs) [34]; and 4) KDD99 network intrusion detection data set [35]. These data sets are processed as IID and non-IID rather than conservative CIFAR-10 [36], MINIST, FEMINIST, and LEAF [37] data sets. On the other hand, after preprocessing data with the algorithms mentioned above (FSHFL), the performance of Fed-Avg is improved in terms of global accuracy (1.68%), training time (6.9%), and the EC (2.85%) in HFL.

4) Field Deployment: We use the Rook and Ceph block storage technology to meet the horizontal learning requirements. These horizontal learning data storage methods are open-source and scalable. The Rook is also an orchestrator for distributed storage that runs in cloud-native environments for HFL. Exploring the boundaries between real-world use cases enabled by 6G and the fulfillment of FL requirements is achieved through interpreting those requirements into cloud-native technology concepts and using those resources to deliver real-world experimentation.

B. Paper Structure

The remainder of this article is organized as follows. Related work is presented in Section II. The system model is described in Section III. The proposed three steps of the FSHFL are detailed in Section IV. Next, the experimental results analysis and the testbed are separately given in Sections V and VI. Finally, Section VII summarizes the highlights of this work.

II. Related Work

Before we introduce details of the proposed FSHFL, the related work is introduced in two steps. First, outlier detection in data dimension reduction. Second, the FS methods in FL.

A. Outlier Detection in Data Cleaning

There are several studies on outlier detection in data dimensional reduction and cleaning [36], [38], [39], [40], [41]. Wang et al. [36] gave a multidimensional data cleaning method by angle-based multifeature data outlier detection, followed by the support vector machine (SVM) with the kernel function used to fit the cleaning model. Ma et al. [38] proposed a federated data cleaning protocol which is designed to achieve data cleaning without compromising data privacy by efficiently computing the attribute value frequency scores of collected data entries. In [39], the outlier detection (an algorithm base on OCSVM) is performed on the data collected by each IoT node. Combining the autoregressive exogenous model and residual calculation, the impact of noise on the original data is minimized. In the paper [40], Teh et al. gave an anomaly detection framework that uses unsupervised FS on systematically engineered time-series features along with PCA to detect sensor data anomalies. But the PCA algorithm shares the same principle of maximizing variance, it involves feature transformation and obtains a set of transformed features rather than a subset of the original features. Xing et al. [41] adopted weighted features to SVM by using the MI-based approach to assign appropriate weights for each feature, which generally exhibits better generalization performance by experiment results on benchmark data sets.

However, the aforementioned works involve no studies of HFL. For solving the data cleaning problem in HFL, especially aiming at non-IID data sets, every client has its characteristics. Therefore, a threshold-free anomaly detection method should be used to personalize how much data could be marked as outliers at each client, in other words, how much data are not representative enough to show the federated attributes (the size of the yellow grid and blue grid in Fig. 1). Also, the novel unsupervised cleaning approach can be applied to newly deployed HFL clients for which no anomalies have been observed yet. Since there is a lack of studies, in this article, we use an improved OCSVM approach called FAR-OCSVM with adjustable hyperparameters to fill this gap in HFL data cleaning.

B. Federated Feature Selection Methods

In recent years, some scholars focus on the FS for FL [23], [24], [25], [26], [27]. Specifically, Hu et al. [23] proposed a multiparticipant federated evolutionary FS algorithm for imbalanced data under privacy protection. A multilevel joint sample-filling strategy with multiparticipation is proposed to fill imbalanced or empty classes on each participant. Then, a federated evolutionary FS method based on supervised PSO with multiple participants is proposed by periodically sharing the optimal feature subset on participants. Yao et al. [24] proposed a feature fusion method to address heavy communication costs and inevitable performance drop compared with Fed-Avg. But without data cleaning, if the hacker injects some malicious data in feature space and sample space, the fusion results will crack the whole FL system. Cassará et al. [25] showed an MI-based FS algorithm which runs by autonomous vehicles (AVs). With an aggregation function based on the Bayes theorem executed on the edge, all the distributed AV iteratively collaborate to filter out the redundant or irrelevant attributes. Banerjee et al. [26] proposed an information-theoretic FFS approach, named Fed-FiS. Fed-FiS evaluates the feature–feature MI and feature-class MI to get the local feature subset and the global feature set. However, [26] does not clean the data sets before FS, plus, because of the non-IID attributes of different IoT data sets, not all the data are labeled, so the generic FS approach should be unsupervised, [23] did not consider either. A greedy algorithm to select features that achieve better intrusion detection accuracy is given in [27], and multiple global models are generated by the server in FL according to the decided features.

Besides, conventional centralized data processing usually tries to reduce data redundancy and introduce less ambiguity.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

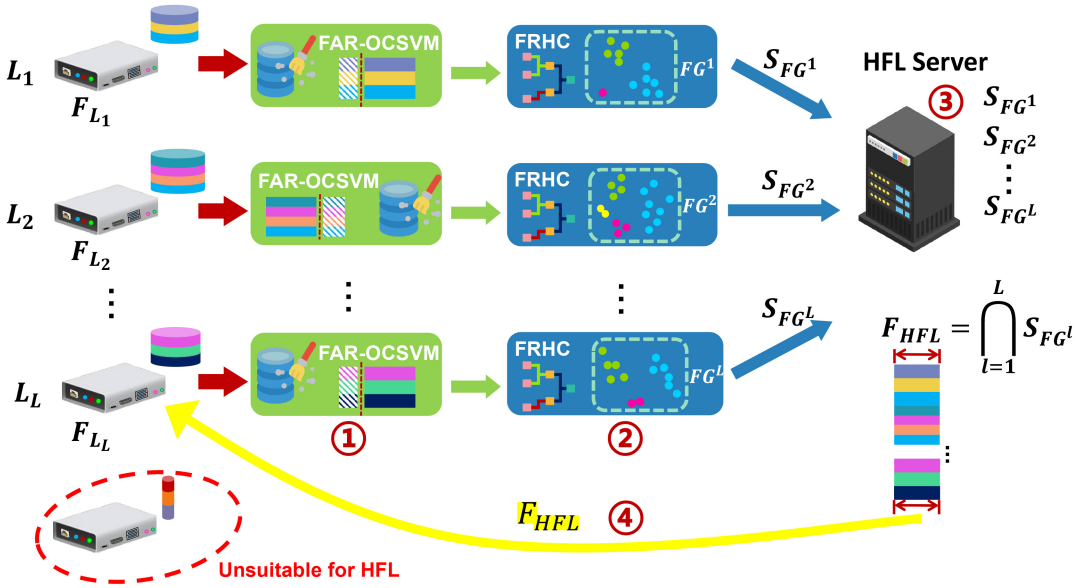

Fig. 2. Architecture of FFS in HFL.

TABLE I
DEGREE OF DIFFICULTY TO HAVE HIGH-PERFORMANCE FL WHICH INFLUENCED BY NON-IID DISTRIBUTION AMONG DATA SETS OF DIFFERENT PARTICIPANTS

Features Distribution / Samples Distribution	IID Features	non-IID Features
IID Samples	EASY	MEDIUM
non-IID Samples	MEDIUM	HARD

TABLE II
DIFFERENT METHODS FOR FS IN FL PROPOSED IN THE LITERATURE. NONE OF THE EXISTING METHODS SATISFIES ALL REQUIREMENTS (*C1*)–(*C3*) OF HFL

Papers	Data Cleaning	Feature Relevance	Unsupervised Method
Hu Y *et al.* [23]	TRUE	TRUE	FALSE
Yao X *et al.* [24]	FALSE	FALSE	TRUE
Cassará P *et al.* [25]	TRUE	TRUE	FALSE
Banerjee S *et al.* [26]	FALSE	TRUE	FALSE
Qin Y *et al.* [27]	FALSE	FALSE	FALSE
FSHFL (Proposed Method)	TRUE	TRUE	TRUE

However, due to the original decentralized nature of distributed IoT and non-IID data, there exist fewer problems of redundancy and excess. On the contrary, due to the sparse data distribution, some algorithms need to increase the redundancy of the data to manifest the main points of data sets from a federated global aspect. The advantages of HFL can provide this redundancy enhancement by distributed training to achieve global optimum. Most features can represent the personality of the data sets, making it easier to find potential patterns from the non-IID data. As a result, the feature clustering algorithm proposed in this article increases redundancy from the perspective of global FL training. FRHC uses the redundancy of features for training to some extent and reflects the advantages of HFL. It can be seen from the experiment part that the FRHC algorithm can effectively improve the expression of features. The reduction of dimensions after the cluster selection also reduces the cost of data training. This work also considers the condition of IID samples with non-IID features and non-IID samples with non-IID features (Table I) in our experiments. Based on the FS problem of the HFL, we conclude that an efficient FS algorithm for HFL needs to fulfill the following conditions. *C1:* It should clean the data before selecting features. *C2:* It should find the relationship between different features. *C3:* It should be unsupervised methods. The pros and cons of the above-mentioned FFS research are summarized in Table II. To the best of our knowledge, this is the first work of the FS method for HFL in IoT networks.

III. FEATURE SELECTION ARCHITECTURE DESIGN

A. Architecture of Federated Feature Selection in HFL

FL is a distributed learning framework that enables multiple local participants to collaboratively train and use a learning model for a given task. The natural distributed attributes establish a hierarchical architecture (Fig. 2), which is designed in this article. In our FL system, we consider L local devices L_l, $l \in [1, L]$ as data owners and an HFL server. We suppose $L \geq 2$, which means more than two participants in the HFL system. If $L = 1$, the system degenerates into a centralized system with all information of one data owner. We use F_{L_l} to indicate the feature set of local L_l. The whole feature

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

set of the IoT HFL system contains n features which can be denoted by $F = \{f_1, f_2, \ldots, f_n\}$ and $F = \sum_{l=1}^{L} F_{L_l}$. The number jth feature in F can be written as f_j, where $f_j \in \mathbb{R}^{1\times n}$. F is distributed across L different local devices and L_l contains d features can be described as $F_{L_l} = \{f_1^{L_l}, f_2^{L_l}, \ldots, f_d^{L_l}\}$ and $d \leq n$. The process of the FSHFL includes four steps from left to right in Fig. 2. First of all, each local device with its own feature set generates a data set without data leakage. Different colors of data in Fig. 2 denote that the data from user devices have fewer overlaps on samples while the flatness (fat or thin) of the data is similar, which means that data sets have significant overlaps on the feature space. If one data set has fewer features, the device which holds this data set may not be suitable to join the HFL, like the device in the red circle in Fig. 2. Since features configured to the device are obtained from a broader set of features, devices with more features are more suitable for HFL, like devices $L_1, L_2, \ldots, L_L$. In step 1, each local device L_l with the feature set F_{L_l} generates a local data set, then the proposed FAR-OCSVM algorithm filters out the low-relevance features and implements data dimensionality reduction. For each data set participating in HFL, most of the features already reflect the characteristics of the entire data set and contribute almost all information gain (IG), which is also the original intention of realizing HFL. We use the feature average relevance (FAR) degree among features as an indicator, features with less FAR mean less relationship with the overall expression. In FAR-OCSVM, features with less FAR are eliminated as anomalies (more detailed experimental verifications are in Section V), so irrelevant and low-representative features (as well as some malicious features) can be picked out and deleted. In addition, due to the non-IID distribution of data sets, the anomaly features may also be different. In this situation, an algorithm with variable thresholds is required. The advantage of our FAR-OCSVM is to personally handle these anomalies at each local. The purpose of FS is to select those features that can express the information of the data set to the fullest extent. Conversely, the unselected features only provide a tiny amount of information when learning from the data set. After cleaning data, we need to find what sets of features can represent this data set. These feature sets should cover most of the information of the data set and be more representative and expressive in HFL. We innovatively use the correlation degree of features as the benchmark to perform hierarchical clustering (HC) with features as clustering objects. Within each feature cluster, features are correlated and dependent on each other, also taking advantage of the properties of non-IID data. The more dependent features in a feature cluster, the more representative this feature cluster is, jointly, these feature clusters are more representative of the data set as a whole. These kinds of feature clusters are also informative about learning from non-IID attributes. By selecting feature subclusters after FRHC, optimized feature sets can be found. Then, in step 2, each local client executes the FRHC algorithm to highlight and explicit representative features. A set of locally ideal feature clusters can be gathered and selected subsets before uploading to the global server. After the FRHC algorithm, features are grouped into different clusters, and the cluster with more features has greater data set representation.

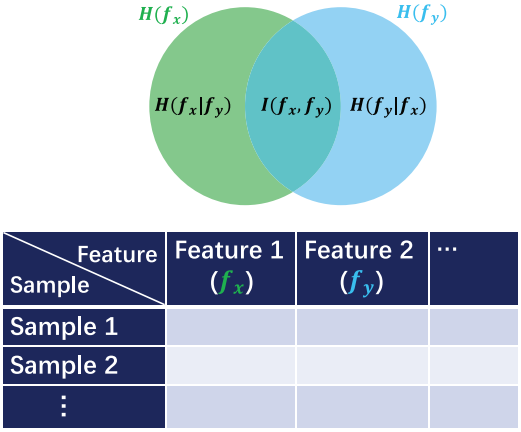

Fig. 3. Example of the dependency between pairwise features in the data set.

After step 1, F_{L_l} has changed to $\hat{F_{L_l}}$ which can be grouped as K feature clusters, $\hat{F_{L_l}} = \{\mathrm{FG}_1{}^l, \mathrm{FG}_2{}^l, \ldots, \mathrm{FG}_K{}^l\}$, $2 \leq K \leq e$, e is the number of features after FAR-OCSVM and $e \leq d$. Then, we can select the subsets of $\hat{F_{L_l}}$ (marked as S_{FG^l}) after finding the best K with silhouette coefficient (SC), in order to have more integral feature expression. In step 3, after gathering the feature sets selected by local users, the global HFL server does the intersection operation. Then, the federated features of the system are obtained. The uploaded features represent the information of data sets to a large extent and can be used as the overlaps part of features in HFL. In step 4, the global server gives feedback to local users about selected federated features to realize local feature enhancement without leaking other clients' data. As a result, the local device will know which features should pay more attention to and try to avoid mistakes, from the aspect of global HFL.

B. Feature–Feature Relevance Measure

To better measure, the relationship between features, symmetric uncertainty (SU) [42] is commonly used as an index. This article currently only considers the relationship between pairwise features, and more complex feature relationships are reserved for future work. SU aims to measure the change of information uncertainty in a system. For one local L_l, the feature set of it can be described as $F_{L_l} = \{f_1^{L_l}, f_2^{L_l}, \ldots, f_d^{L_l}\}$, $d \leq n$, $\mathbf{F_{L_l}} \in \mathbb{R}^d$. F_{L_l} is a discrete set and we choose an arbitrary pair of feature $p(f_x, f_y)$. SU can be used to measure the degree of dependence between f_x, f_y, and can express the information content they have in common (Fig. 3), SU also shows under the condition of one known feature, how much reduction of the uncertainty in another feature. SU measures entropy introduced by information theory and symmetrically treats a pair of variables. It can be computed as follows:

$$SU(f_x, f_y) = 2\left[\frac{MI(f_x, f_y)}{H(f_x) + H(f_y)}\right] \tag{1}$$

where $MI(f_x, f_y)$ is the MI between feature f_x and feature f_y, MI quantifies the "amount of information" obtained about one random variable f_y by observing the other random variable f_x. $H(f_x)$ and $H(f_y)$ are the entropy of feature f_x and f_y (the

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

feature is marked as a discrete random variable). The methods to calculate each part in (1) are as follows:

$$MI(f_x, f_y) = H(f_x) - H(f_x|f_y) \tag{2}$$

$$H(f_x) = -\sum_{f_x} P(f_x)\log P(f_x) \tag{3}$$

$$H(f_x|f_y) = -\sum_{f_y} P(f_y) \sum_{f_x} P(f_x|f_y)\log P(f_x|f_y). \tag{4}$$

In (2)–(4), $P(f_x)$ is the probabilities that the feature f_x with different values, same as $P(f_y)$ to f_y. $P(f_x|f_y)$ is the posterior probability which means when the feature f_y takes a certain possible value, the probability of the feature f_x takes the corresponding possible value. $H(f_x|f_y)$ is the conditional entropy. Analyzing the above (1)–(4) show that the value range of $SU(f_x, f_y)$ is in [0, 1]. $SU(f_x, f_y) = 0$ means that the two features are not related; $SU(f_x, f_y) = 1$ indicates that two features f_x and f_y are completely correlated, also expresses that the two features have the same amount of information in the data set. Therefore, the larger value of the $SU(f_x, f_y)$ is, the more dependency exists between features f_x and f_y.

Most of the feature overlaps between the data sets from HFL participants, therefore, the data containing these overlapping features will eventually contribute the most amount of information in the learning process of HFL. These data also reflect the homogeneity of the data sets concerning each other, and this homogeneity highlights the underlying pattern and provides it to HFL.

From the perspective of unsupervised learning, we innovatively liked the feature set of each data set to a whole cluster, each element in the cluster is a feature. The more a feature relates to other features, the more related features can be aggregated together. These gathered features can reflect the information of the overall feature cluster. On the contrary, those features with the lowest correlation compared with other features are also less correlated with the performance of the overall feature cluster, the "cohesion ability" of features is lower. These features can be regarded as insignificant attributes and eliminated. Features with low cohesiveness relatively have less IG, not the other way around. In HFL, it can be considered that there is a large overlap between the feature cluster from A and the feature cluster from B (Fig. 1), the overlapping part to a large extent represents how many features can be aggregated together at each local. Consequently, those features with the minimum average correlation can be considered exceptions and eliminated, like the FAR detection in Fig. 4.

In this work, we use the FAR to measure the amount of information contained in the features. The more features with higher relevance, the easier to gather them. So this overall feature cluster can better reflect the information contained in the data set. Conversely, features with minimum average relevance are not easily associated with other features, nor easy to participate in the whole feature aggregation. The information contained in such features is much lower, and we mark these features as anomalous features. For one feature $f_i^{L_l}$ in feature set F_{L_l} of device L_l, the FAR of $f_i^{L_l}$ is the average value of the SU with the feature $f_i^{L_l}$ and each feature in the feature set F_{L_l}. The definition of FAR of the feature $f_i^{L_l}$ in feature set F_{L_l}

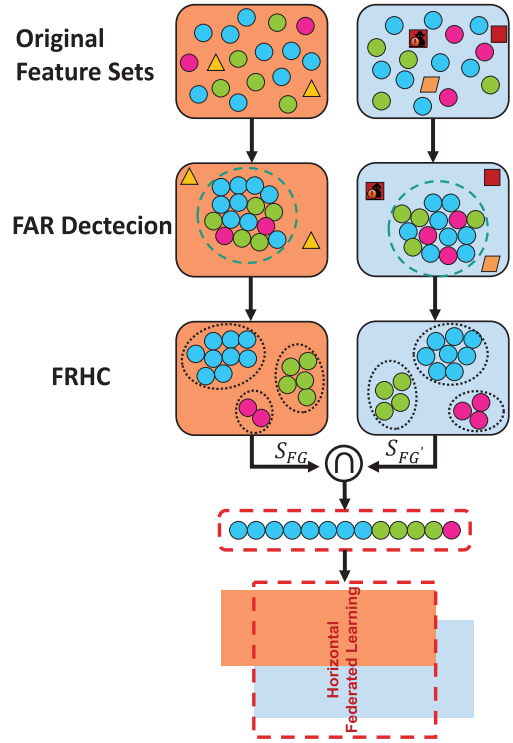

Fig. 4. Main idea of FS in this article.

is defined as follows:

$$r_{L_l}\left(f_i^{L_l}\right) = \frac{1}{d}\sum_{i=1}^{d} SU\left(f_i^{L_l}, f_j^{L_l}\right). \tag{5}$$

In our FSHFL system (Fig. 2), the FAR-OCSVM algorithm is used on each device to handle the data cleaning and dimension problem, so for one local L, (5) can be simplified as

$$r_L\left(f_i^L\right) = \frac{1}{d}\sum_{i=1}^{d} SU\left(f_i^L, f_j^L\right) \tag{6}$$

d is the number of features in local L, $r_L(f_i^L)$ is the FAR of f_i^L. To this point, we give a FAR weight to the data of each feature (from f_1^L to f_d^L) in the data set of L, the FAR weight vector can be described as follows:

$$\mathbf{r_L} = \left[r_L\left(f_1^L\right), r_L\left(f_2^L\right), \ldots, r_L\left(f_d^L\right)\right]^T. \tag{7}$$

To facilitate the derivation of the following FAR-OCSVM, we simplify it as follows:

$$\mathbf{r_L} = [r_1, r_2, \ldots, r_d]^T \tag{8}$$

which means the FAR is weighted on features in $\{f_1^L, f_2^L, \ldots, f_d^L\}$ of data at local L. With this FAR weight vector, we can add it to the OCSVM. There are two obvious advantages of our FAR-OCSVM data cleaning (dimension reduction) algorithm.

1) Selecting the feature data with the greater relationship as the training set, and the feature data with less relationship as anomalies to maximize the feature difference.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

Thus, the classifier can be enabled to have a better discrimination performance.

2) Meanwhile, each classifier of HFL participants can customize the threshold of dimension reduction, according to the characteristics of the local data.

In this case, local features that are not related to the global features can be highlighted and deleted. Plus, the advantage of the FRHC algorithm is gathering relational features into different clusters by obtaining the optimal number of clusters without human intervention. Clusters which have more informative features are clear in visualization. The proposed two algorithms will be introduced in detail in the next section.

IV. Algorithms and Solutions

In this section, the core algorithms mentioned in this article are introduced. We will first explain how to detect the anomaly data and then find the classification function to build the high-dimensional data cleaning model. After gathering the cleaned data, algorithms for selecting the local features and global federated features are given for HFL.

A. FAR-OCSVM Algorithm

1) One-Class SVM Outlier Detection: Unlike the traditional SVMs, one-class SVM is proposed by Schölkopf et al. [43] for novelty detection. One-class classification allows classifying just one-class objects and distinguishing them from all other possible objects. Objects can be classified well by the classifier, but the others will be classified as outliers. We considered a data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^d$, which is the given *target class*. The idea behind the OCSVM is to describe the target class by a function that maps most of it to a region where the function is nonzero. For this reason, OCSVM creates a closed inlier boundary (as to a hyperplane) and finds the maximal gap hyperplane that separates the target class from the origin (i.e., the only member of the nontarget class is the origin). To deal with nonlinear problems, the hyperplane is defined in a high-dimensional Hilbert space $\mathbf{H}$ where the samples are mapped through a nonlinear transformation Φ. The closed boundary is found by the following formulation:

$$\min_{\boldsymbol{\omega},\boldsymbol{\xi}\in\mathbb{R}^N,\rho\in\mathbb{R}} \left\{ \frac{1}{2}||\boldsymbol{\omega}||^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^N \xi_i \right\} \tag{9}$$

constrained to

$$\begin{aligned} &\boldsymbol{\omega}\cdot\Phi(\mathbf{x}_i) \geq \rho - \xi_i \\ &\xi_i \geq 0, i = 1, 2, \ldots, N. \end{aligned} \tag{10}$$

Here, $\boldsymbol{\omega}$ is a weight vector perpendicular to the hyperplane, ρ is the offset which means the distance to the origin. Since the soft decision, a set of slack variables $\xi_i \geq 0$ is introduced. N is the number of observations. The parameter $\nu \in (0, 1]$ is the upper bound on the fraction of the outliers and the lower bound on the fraction of the support vectors. According to the Lagrange multipliers α_i, the Karush–Kuhn–Tucker (KKT) conditions and the strong duality theorem, $\boldsymbol{\omega} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i)$. Introducing a kernel function whose elements are defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$, and $\Phi(\mathbf{x})$ is a mapping function from input space $\mathbf{X}$ to the Hilbert space $\mathbf{H}$. $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ is the inner product of $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$. In this research, we choose the widely used radial basis function (RBF) kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-[||\mathbf{x}_i - \mathbf{x}_j||^2]/[2\sigma^2])$. The dual problem becomes the following formula with constraints:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\left\{ \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N} \\ & \sum_{i=1}^N \alpha_i = 1. \end{aligned} \tag{11}$$

After solving the dual problem, a set of model weights ($\boldsymbol{\omega}$ is a linear combination of α_i) is obtained, and the decision function for any test $\mathbf{x}_i$ is given by

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho \right). \tag{12}$$

Apart from finding a hyperplane which separates the samples of the target class from the origin, there also exists a hyper ball with minimum volume surrounding only the target class samples. This is an equivalent formulation of the OCSVM called support vector domain description (SVDD) which was introduced in [43]. All samples lying inside the hyper ball belong to the target class, whereas the samples lying outside are outliers (Fig. 5). By giving the center b and radius R of the hyper ball, the problem is defined as follows:

$$\begin{aligned} \min_{R,\boldsymbol{\xi}\in\mathbb{R}^N, b\in\mathbb{R}} \quad & \left\{ R^2 + \frac{1}{\nu N}\sum_{i=1}^N \xi_i \right\} \\ \text{s.t.} \quad & ||\Phi(\mathbf{x}_i) - b||^2 \leq R^2 + \xi_i \\ & \xi_i \geq 0, i = 1, 2, \ldots, N. \end{aligned} \tag{13}$$

The proposed FAR-OCSVM uses two above-mentioned methods (OCSVM and SVDD) to separate the data from high-relevance features and low-relevance features, the algorithm will be introduced in the following part.

2) FAR-OCSVM: Since there is no fixed standard to distinguish irrelevant features from relevant features under non-IID conditions, the main idea of this technical part is to use adaptable hyperparameters in one-class classification problems. OCSVM can exactly meet the requirement. Anomaly detection aims to quickly and effectively detect abnormal points from data sets. In many cases, the IoT network data exist in a high-dimensional form. Especially in HFL, high-dimensional data will inevitably introduce noise, interference, and even some malicious data. In HFL, features have more overlaps, so the overlapping part can contribute the most information to global federated training, and most features are useful. In order to pick out that useless part of features, inspired by unsupervised learning, we creatively gather the features from the data set of one HFL participant into a whole cluster. Each element in the cluster is a feature. Considering data relevance [44], [45], the more a feature relates to other

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

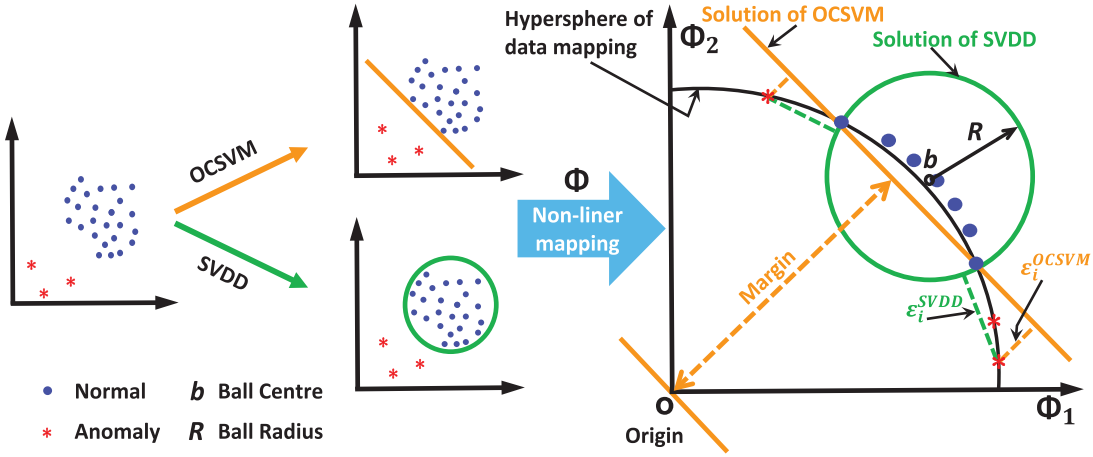

Fig. 5. Sample classification based on OCSVM and SVDD.

Algorithm 1: FAR-Based Outlier Detection

Input: $D_l \in D = \{D_1, D_2, ..., D_L\}$ $D_l = \{f_1^{L_l}, f_2^{L_l}, ..., f_{d_l}^{L_l}\}$ $\mathbf{r_{L_l}} = [r_{L_l}(f_1^{L_l}), r_{L_l}(f_2^{L_l}), ..., r_{L_l}(f_{d_l}^{L_l})]^T$ Relevance threshold $\boldsymbol{T_l}(\boldsymbol{T_l} \in [0, 1])$

Output: Dataset $D_c^l (l = 1, 2, ..., L)$ without minimum FAR features with one label y

```
function Data dimensional reduction
    for l = 1, 2, ..., L do
        D_l ← {}, o_l = 0
        for j = 1, 2, ..., d_l do
            if (r_{L_l}(f_j^{L_l}) ≤ T_l or r_{L_l}(f_j^{L_l}) = 1) then
                Mark data of feature f_j^{L_l} as outlier data
                o_l = o_l + 1
            else
                y_j = +1
                Mark data of feature f_j^{L_l} as training data
                D_l ← D_l + (f_j^{L_l}, y_j)
        D_c^l = D_l
        push(D_c^l, o_l)
    return D_c^l, o_l
```

features, the more related features can be aggregated together. Then, the gathered feature group can reflect the information of the overall feature cluster. On the contrary, features with the lowest relationship compared with other features are also less related to the performance of the overall feature cluster. The cohesion ability of these features is lower, also less correlation with the expression of the overall features. These features can be regarded as insignificant attributes and eliminated. Features with low cohesiveness relatively have less IG, but not vice versa.

To this end, we first design the FAR-based outlier detection algorithm (Algorithm 1). D_l indicates the data set of participant $L_l, (l = 1, 2, \ldots, L)$ in HFL, $D_l = \{f_1^{L_l}, f_2^{L_l}, \ldots, f_{d_l}^{L_l}\}$ means data with d_l-dimensional features, d_l is not constant and $d_l \leq n$. $\mathbf{r_{L_l}}$ is the values of FAR for each feature in data set D_l. $\boldsymbol{T_l}$ are relevance thresholds for each HFL clients L_l and $\boldsymbol{T_l} \in [0, 1]$. In practical experiments, all data sets cannot be shared to the algorithm, so each device L separately executes the innermost loop of Algorithm 1 to pick out the features that have relatively low $r_L(f_i^L)$. Plus, if the FAR of a feature is 1, which means in one D_L, this feature is relevant to every other feature, consequently, the data of this feature is 0 (which will also be proved in our experiments) and this feature contributes no information, which can be seen as an outlier and deleted this column. $\boldsymbol{T_l}$ is the threshold on each device to decide the degree of irrelevance, because of the non-IID data set, different devices have different $\boldsymbol{T_l}$ which can be decided by L_l. After getting the training data, for a device L, the original data set $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^d$, has changed to $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^e$, $e_l = d_l - o_l$, considering one device, e_l is simplified as $e, e \leq d$.

The traditional and improved SVM assume that all the features of samples in a given training set have equal contributions to get the optimal hyperplane. However, for a certain real-world data set, some features may possess more relevance to final results, while the other features may have less relevance. Thus, the features with more relevance are more important to train an optimal SVM. Therefore, it is feasible to construct a kind of SVM with different features possessing different weights [41]. To make more obvious and amplify the influence of data from relevant features, FAR weights are added to the training data of OCSVM, we call it FAR-OCSVM (Algorithm 2). The parameter ν can also change how much data can be seen as outliers. The FAR-weight can make the training data more distinguishable from the outliers which are the data of less related features from a federated information expression perspective. In this way, after training, the classifier can filter out the data of anomaly features, so as to realize data cleaning. To simplify analysis, we consider one local L, it has a training data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ after Algorithm 1, $\mathbf{x}_i \in \mathbb{R}^e$, $y_i = \{+1\}$, the FAR weight vector $\mathbf{r_L} \in \mathbb{R}^e$ for the features of training samples in L is predefined as

$$\mathbf{r_L} = [r_1, r_2, \ldots, r_e]^T \tag{14}$$

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: FAR-OCSVM Learning Algorithm

Input: The training data set $D_c^l = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$, among $\mathbf{x}_i \in \mathbf{X}$, $y_i = 1$, $i = 1, 2, ..., N$

Output: Decision function $f(\mathbf{x})$ for classfication

1: **function** Data outliers detection

2: Choose RBF kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ and proper hyperparameter ν to solve the problem of optimization in (22) or (26).

3: Find the optimal solution $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_N^*)^T$.

4: Select a component of $\alpha*$ which satisfies $0 < \alpha_i < \frac{1}{\nu N}$ and calculate (23) or (28)

5: Construct decision function

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\mathbf{B}\mathbf{x}_i, \mathbf{B}\mathbf{x}) - \rho\right)$$

OR

$$f(\mathbf{x}) = \operatorname{sgn}\left(R^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K\big(\mathbf{B}\mathbf{x}_i, \mathbf{B}\mathbf{x}_j\big) + 2\sum_{i=1}^{N} \alpha_i K(\mathbf{B}\mathbf{x}_i, \mathbf{B}\mathbf{x}) - K(\mathbf{B}\mathbf{x}, \mathbf{B}\mathbf{x})\right)$$

where $\operatorname{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$

6: **return** $f(\boldsymbol{x})$

$\{r_1, r_2, \ldots, r_e\}$ means the FAR from (6) of the selected training features from Algorithm 1 step 10, also satisfying

$$\frac{1}{d} \leq r_k < 1, (k = 1, 2, \ldots, e). \tag{15}$$

Therefore, toward the OCSVM with weighted features FAR, the optimization problem can be designed as follows:

$$\min_{\boldsymbol{\omega}, \boldsymbol{\xi} \in \mathbb{R}^N, \rho \in \mathbb{R}} \left\{\frac{1}{2}||\boldsymbol{\omega}||^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i\right\} \tag{16}$$

constrained to

$$\begin{aligned} &\boldsymbol{\omega} \cdot \operatorname{diag}(\mathbf{r_L}) \cdot \Phi(\mathbf{x}_i) \geq \rho - \xi_i, && i = 1, 2, \ldots, N \\ &\xi_i \geq 0, && i = 1, 2, \ldots, N \\ &r_k \geq \frac{1}{d}, && k = 1, 2, \ldots, e \\ &r_k < 1, && k = 1, 2, \ldots, e. \end{aligned} \tag{17}$$

To solve this problem, the Lagrange function can be used as follows:

$$\begin{aligned} &L(\boldsymbol{\omega}, \boldsymbol{\xi}, \rho, \mathbf{r_L}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}.\boldsymbol{\delta})_{\alpha_i, \beta_i \geq 0, \lambda_k, \delta_k \geq 0} \\ &= \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} - \rho + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i \\ &\quad - \sum_{i=1}^{N} \alpha_i\big[\boldsymbol{\omega} \cdot \operatorname{diag}(\mathbf{r_L}) \cdot \Phi(\mathbf{x}_i) - \rho + \xi_i\big] \\ &\quad - \sum_{i=1}^{N} \beta_i \xi_i - \sum_{k=1}^{e} \lambda_k\left[r_k - \frac{1}{d}\right] \\ &\quad - \sum_{k=1}^{e} \delta_k[1 - r_k]. \end{aligned} \tag{18}$$

From the strong duality and KKT conditions of feasible constraints, no feasible descent, complementary slackness, and the positive Lagrange multipliers

$$\begin{aligned} &\frac{\partial L}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} - \sum_{i=1}^{N} \alpha_i \cdot \operatorname{diag}(\mathbf{r_L}) \cdot \Phi(\mathbf{x}_i) = 0 \\ &\frac{\partial L}{\partial \boldsymbol{\xi}} = \frac{1}{\nu N} - \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \beta_i = 0 \\ &\frac{\partial L}{\partial \rho} = 0, \frac{\partial L}{\partial r_k} = 0, r_k \neq 0 \end{aligned} \tag{19}$$

we can have the following results:

$$\begin{aligned} &\boldsymbol{\omega} = \sum_{i=1}^{N} \alpha_i \cdot \operatorname{diag}(\mathbf{r_L}) \cdot \Phi(\mathbf{x}_i) \\ &0 \leq \alpha_i \leq \frac{1}{\nu N} \\ &\sum_{i=1}^{N} \alpha_i = 1 \\ &\lambda_k = 0, \delta_k = 0 \end{aligned} \tag{20}$$

$$\begin{aligned} &L(\boldsymbol{\omega}, \boldsymbol{\xi}, \rho, \mathbf{r_L}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}.\boldsymbol{\delta})_{\alpha_i, \beta_i \geq 0, \lambda_k, \delta_k \geq 0} \\ &= \min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \big[\operatorname{diag}(\mathbf{r_L})\Phi(\mathbf{x}_i)\big]^T \cdot \big[\operatorname{diag}(\mathbf{r_L})\Phi\big(\mathbf{x}_j\big)\big]. \end{aligned} \tag{21}$$

We set $\operatorname{diag}(\mathbf{r_L}) = \mathbf{B}$, so the strong duality problem becomes the following formula:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\left\{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K\big(\mathbf{B}\mathbf{x}_i, \mathbf{B}\mathbf{x}_j\big)\right\} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N} \\ & \sum_{i=1}^{N} \alpha_i = 1. \end{aligned} \tag{22}$$

When the support vector on the margin

$$\underset{0<\alpha_i<\frac{1}{\nu N}}{\rho} = \boldsymbol{\omega} \cdot \mathbf{B}\Phi(\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_j K\big(\mathbf{B}\mathbf{x}_j, \mathbf{B}\mathbf{x}_i\big). \tag{23}$$

The decision function is as follows:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\mathbf{B}\mathbf{x}_i, \mathbf{B}\mathbf{x}) - \rho\right). \tag{24}$$

Instead of the hyperplane, we also give the hyper ball (SVDD) derivation

$$\min_{R, \boldsymbol{\xi} \in \mathbb{R}^N, b \in \mathbb{R}} \quad \left\{R^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i\right\}$$

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

$$\begin{aligned} \text{s.t.} \quad & ||\text{diag}(\mathbf{r_L})\Phi(\mathbf{x}_i) - b||^2 \le R^2 + \xi_i \quad i = 1, 2, \ldots, N \\ & \xi_i \ge 0, \quad i = 1, 2, \ldots, N \\ & r_k \ge \frac{1}{d}, \quad k = 1, 2, \ldots, e \\ & r_k < 1, \quad k = 1, 2, \ldots, e \end{aligned} \tag{25}$$

The dual problem and solutions are in (26) and (27)

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K\big(\mathbf{Bx}_i, \mathbf{Bx}_j\big) - \sum_{i=1}^{N} \alpha_i K(\mathbf{Bx}_i, \mathbf{Bx}_i) \right\} \\ \text{s.t.} \quad & 0 \le \alpha_i \le \frac{1}{\nu N} \\ & \sum_{i=1}^{N} \alpha_i = 1 \end{aligned} \tag{26}$$

$$b = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{Bx}_i). \tag{27}$$

When the support vector on the hyper-ball

$$\begin{aligned} \underset{0<\alpha_i<\frac{1}{\nu N}}{R^2} &= ||\Phi(\mathbf{Bx}_i) - b||^2 \\ &= K(\mathbf{Bx}_i, \mathbf{Bx}_i) - 2\sum_{j=1}^{N} \alpha_j K\big(\mathbf{Bx}_i, \mathbf{Bx}_j\big) \\ &\quad + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K\big(\mathbf{Bx}_i, \mathbf{Bx}_j\big). \end{aligned} \tag{28}$$

The decision function is as follows:

$$\begin{aligned} f(\mathbf{x}) = \text{sgn}\Bigg(& R^2 - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K\big(\mathbf{Bx}_i, \mathbf{Bx}_j\big) \\ & + 2\sum_{i=1}^{N} \alpha_i K(\mathbf{Bx}_i, \mathbf{Bx}) - K(\mathbf{Bx}, \mathbf{Bx}) \Bigg). \end{aligned} \tag{29}$$

Combining Algorithms 1 and 2, we named it the FAR-OCSVM data cleaning method. There are two obvious advantages of the FAR-OCSVM.

1) Selecting the data with the greater relationship as the training set, and the data with less correlation as anomalies. By adding FAR weight, features with more relevance can be highlighted, these features are also more important for training.
2) Meanwhile, each classifier of HFL participants can customize the threshold and parameter of dimension reduction according to the characteristics of the local data.

B. Local Representative Feature Selection

After FAR-OCSVM, we get the cleaned and dimension-reduced data for HFL. Intuitively, we are inspired by gene expression in biology, the highly expressed genes can decide more on a specific disease or protein [46], [47]. The same idea is used in the FRHC for the HFL FS. In each data set, we can gather related features together to generate different feature clusters, the more features in one cluster, this cluster of features has the higher expressions for the results. In each local L, we manifest its representative feature clusters by Algorithm 3, which shows the pseudocode of the agglomerative hierarchical procedure. Two results will be the output, $\mathbf{K}$ and D_L. $\mathbf{K}$ is the optimal number of clusters calculated by SC. D_L gives visualized clustering results.

Algorithm 3: FRHC Algorithm

Input: $D_L = D_c^L = \{f_1^L, f_2^L, ..., f_e^L\}$
Output: K scores, clustering results D_L

```
function Features agglomerative hierarchical clustering
    for i, j = 1, 2, ..., e, i < j do
        compute δ = cor(i, j)
        store [δ, i, j] in Q
    for t = 1, 2, ..., e − 1 do
        [δ̄, ī, j̄] = Gavg_{q=[δ,i,j]∈Q} q[1]
        remove f_ī^L and f_j̄^L from D_L
        remove entries [∗, ī, ∗], [∗, ∗, ī], [∗, j̄, ∗], [∗, ∗, j̄]
          from Q
        insert f_ī = f_ī ∘ f_j̄ in to D_L
        foreach h:[∗, h, ∗], [∗, ∗, h] ∈ Q do
            δ = cor(h, ī)
            if h < ī then
                store [δ, h, ī] in Q
            else
                store [δ, ī, h] in Q
    clustering results = D_L
    for K = 2, 3, ..., e do
        SC_K(D_L)
    find the descending order{SC_K(D_L)} with
      corresponding {K}
    return D_L, K
```

In Algorithm 3, there are three functions: 1) cor(); 2) $Gavg()$; and 3) $SC_{\mathbf{K}}()$. cor() computes the correlation between two features. In $Gavg()$, we choose the group average as the prioritized linkage measure (among single, complete, average, and ward methods) based on [48]. $SC_{\mathbf{K}}()$ function calculates the optimal number ($\mathbf{K}$) of clusters and also shows what kind of features can be grouped together to reveal some characteristics of the data. Q [ ] is a data structure for storing triples $q = (\delta, i, j)$ where δ is the correlation distance between the *ith* and the *jth* clusters of features. Q [ ] is an upper or lower triangular matrix. The algorithm takes input as the set of e features with data to cluster. Each of them is considered as a single cluster, hence the size of the initial clustering is e. Before starting to iterate, the correlation distances between any pair of features are computed and stored in the entries above the main diagonal of the matrix Q [ ].

At each iteration, the two closest clusters are identified ($Gavg()$ operation performed on $q[1]$, the first component of triples q); their rows and columns in Q [ ] are removed; then, they are merged ($\circ$) and the smallest of their two indexes is

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

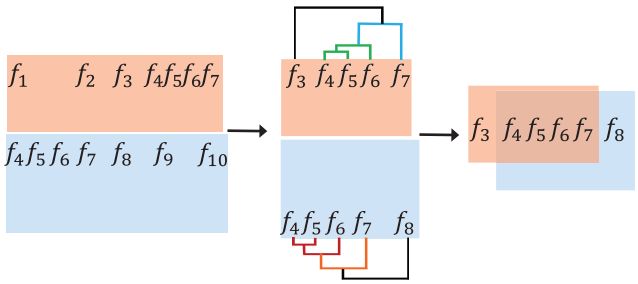

Fig. 6. Idea of proposed FFS.

assigned to the new cluster; finally, the distance of the new cluster from any other cluster is computed and stored in the upper part of Q [ ]. After ending the iteration, D_L verifies with $SC_{\mathbf{K}}()$ to ensure obtaining quality grouping and the optimal number of clusters.

Then, the feature set of data in local L_l can be seen as $\hat{F_{L_l}} = \{\mathrm{FG}_1{}^l, \mathrm{FG}_2{}^l, \ldots, \mathrm{FG}_K{}^l\}, (2 \leq K \leq e_l)$. Then, we can select the representative sub-feature groups from $\hat{F_{L_l}}$ (marked as S_{FG^l}), to have more integral feature expression. The selected S_{FG^l} reserves all or most information of D_l. Ultimately, the feature part from local users for the global overlap is selected. S_{FG^l} can be written as

$$S_{\mathrm{FG}^l} = \mathrm{argmax}\left(|\mathrm{FG}_i^l|\right) + \mathrm{argsubmax}\left(|\mathrm{FG}_j^l|\right) + g \tag{30}$$

where $|\mathrm{FG}_i^l|(i \in [1, \mathbf{K}])$ means how many features in the cluster i, $\mathrm{argmax}(|\mathrm{FG}_i^l|)$ means the cluster which contains most features, $\mathrm{argsubmax}(|\mathrm{FG}_j^l|, i \neq j)$ means the cluster which contains the second most features. g is the compensation feature. Then, L_l uploads S_{FG^l} to the HFL server.

C. Global Federated Feature Selection

Each HFL local does functions of FAR-OCSVM data cleaning, FRHC features agglomerative HC, and subfeature group selection. After the selected $\{S_{\mathrm{FG}^1}, S_{\mathrm{FG}^2}, \ldots, S_{\mathrm{FG}^L}\}$ is gathered by the HFL server, the interaction operation is implemented as follows:

$$F_{\mathrm{HFL}} = S_{\mathrm{FG}^1} \cap S_{\mathrm{FG}^2} \cap \cdots \cap S_{\mathrm{FG}^L} \tag{31}$$

to find the global overlapping federated features (F_{HFL}) for HFL (Fig. 6). Then, the HFL server gives feedback to HFL participants about the selected federated features. F_{HFL} will be distributed to every local client thus the users can understand what kind of features can play main roles in HFL and should pay more attention, without leaking data from any clients.

D. Complexity Analysis

Since the attributes of HFL, each local client $l, l \in [1, L]$ is only in charge of data cleaning and local feature clustering, then, it sends the selected S_{FG^l} to the server. In each device, the time complexity of the FAR-based outlier detection is $\mathcal{O}(n)$. For the distance matrix in Algorithm 3, the time complexity is derived as: 1) distances matrix construction: $\mathcal{O}(n^2)$; 2) sorting of the distances (from the closest to the farthest): $\mathcal{O}(n^2 \log n)$; 3) the grouping of the items is done by iterating over the sorted list of distances: $\mathcal{O}(n^2)$; and 4) finally, optimization of the number of clusters: $\mathcal{O}(n^2)$. The overall time complexity is $\mathcal{O}(n^2) + \mathcal{O}(n^2 \log n) \ + \mathcal{O}(n^2) + \mathcal{O}(n^2) = \mathcal{O}(n^2 \log n)$.

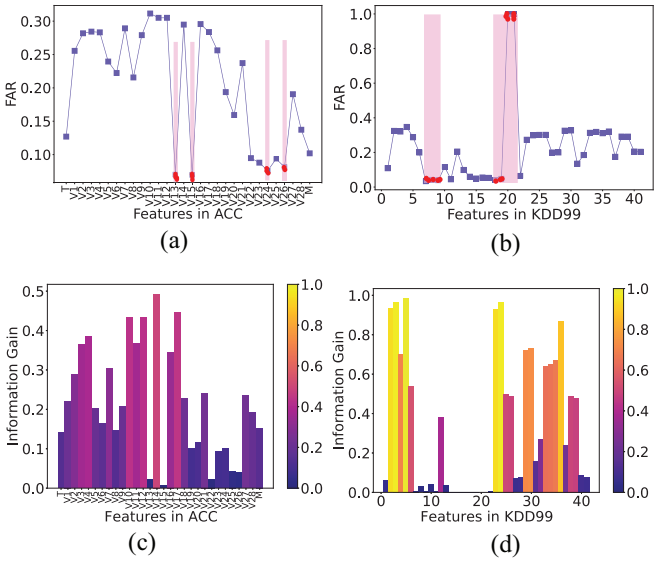

Fig. 7. Anomaly features detection results in (a) ACC and (b) KDD99. (c) IG of each feature in ACC and (d) KDD99.

V. Experimental Results

The experiments are described in detail in this section. Including data cleaning, features agglomerative HC, features selection, and HFL results on four advanced IoT data sets. The analytical discussions on corresponding evaluation results are followed.

A. Experiment Settings and Data Sets Description

We consider an HFL system with ten users and one HFL server in the following experiments. At the very beginning, all users are given an original model for local training. FL is mostly suited for parameterized learning, especially for many types of neural networks. In our tests, each client has a multilayer perceptron (MLP) model (6-layer MLP). Rather than decay the learning rate from the aspect of the number of local epochs, our experiments concern the decay with the number of global federated aggregations to verify how FS can benefit the HFL system. The final metrics are accuracy, convergence time, and EC of the HFL system. Stochastic gradient descent (SGD) is the default optimizer with cross-entropy loss functions. The data in experiments is horizontally partitioned. Parameter averaging will be weighed based on the proportion of data points contributed by each participant as the Fed-Avg, one of the pioneering works on FL [20]. Each client indicates the number of data points they trained with while updating the server with new parameters after each local training step.

Each client sets initialized weights to the current parameters. The local clients then train for one epoch. After training, the new weights are sent to the server which summarizes all the scaled local trained weights and updates the global model for the next aggregation. E is the number of local epochs on each

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

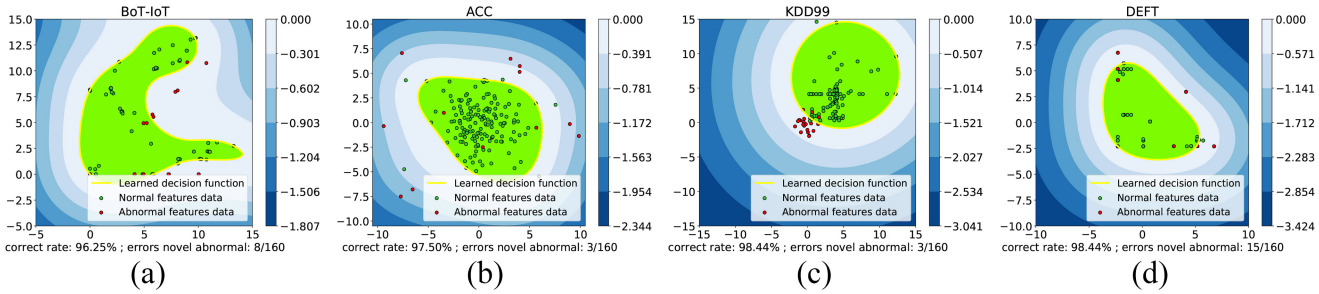

Fig. 8. FAR-OCSVM detection results on four data sets. (a) Bot-IoT. (b) ACC. (c) KDD99. (d) DEFT.

TABLE III
SUMMARY OF EXPERIMENTAL DATA SETS PARAMETERS

Dataset	Features	Sizes	Classes
Bot-IoT	43	668522	2
ACC	30	284807	2
KDD99	41	494021	5
DEFT	111	7289	16

global round, and B is the local minibatch size. B is set as 10, the learning rate is set as 0.01. We set the total global iterations to be 100 and test the trained global model after each communication round on the test data, referring to some state-of-the-art research [20], [49], [50].

Considering non-IID data in HFL, we employ two configurations of data distributions: 1) IID and 2) non-IID. For the IID case, we assume that the training data set of each client is an IID sample of the original overall training data set. As for the non-IID case, we refer to the Appendix of [51], split and set every client to hold a random proportion of the entire training data set. This approach makes every client has its data personality. The non-IID distribution will also be shown in the next part from the aspect of different samples and different SC in different clients.

Four state-of-the-art IoT network data sets are tested in our HFL experiments: 1) IoT network traffic data set (DEFT [32], [52]) for device classification; 2) Bot-IoT data set [33] for IoT botnet attack detection; 3) ACC data set [34] to predict whether this deal is fraud or not; and 4) KDD99 network intrusion detection data set [35] which can be used to get a predictive model and distinguish between intrusions and normal network connections. The above four data sets are processed as IID and non-IID distribution rather than ossified CIFAR-10, MINIST, and FEMINIST data sets. Table III shows the summary of the experiment data sets parameters. *Data set* denotes the abbreviated names of data sets, *Features* represent the number of features in the data sets, *Sizes* are the sample sizes of the data sets, and *Classes* represent the number of categories contained in the data sets. It can be seen from Table III that the feature dimension of the selected data sets ranges from 30 to 111. The number of samples ranges from 7289 to 668522. For BoT-IoT, we choose all features IoT_Botnet_Full5pc_4 data set in [33] and 16 IoT devices data is used in DEFT. Then, we use the proposed FSHFL to test with these data sets, aiming to realize the optimized HFL.

B. Results and Analysis

1) Local Anomaly Features Selection: In Fig. 7, we show that how the FAR-based outlier detection algorithm works on ACC and KDD99 data sets. Since many network traffic features need to be removed in the BoT-IoT and DEFT data sets, to simplify the display of results, here we only show the detection results of Algorithm 1 on the ACC and KDD99 data sets. Fig. 7(a) and (b) shows that after Algorithm 1, features have less FAR value are picked out from each data set. In ACC, we set $\boldsymbol{T}$ as 0.088, $\boldsymbol{T}$ as 0.048 in KDD99. From Fig. 7(a), features V13, V15, V24, V26 are selected. From Fig. 7(b), numbers 7, 8, 9, 18, 19, 20, and 21 features are marked as irrelevant features. The FAR of number 20 and 21 features are 1, which means that these two features are completely relevant with every other feature in the KDD99, unsurprisingly, the value of these two features are zero and contribute no information. Correspondingly, Fig. 7(c) and (d) separately gives the IG of each feature in ACC and KDD99. Comparing Fig. 7(a) with (c) and Fig. 7(b) with (d), we can find that Algorithm 1 effectively detects the features which have less contribution to the final results. From results in [33], we also find that Algorithm 1 can effectively detect features in Bot-IoT with less IG.

Fig. 8 displays that the FAR-OCSVM with RBF kernel is used to train the results which are generated by Algorithm 1. To make the classification results more visible, the data are properly normalized and scaled. The yellow curve represents the training result model $f(\mathbf{x})$. Since only one class (green dots), the data from anomaly features marked by Algorithm 1 is used as validation (red dots) to test the learning curve boundary. It can be seen that the correct rate of $f(\mathbf{x})$ in four data sets cannot 100% fit the normal data training set, this is also in line with the nature of ML by avoiding overfitting. Obviously, there is a certain error (2%–12%) in judging the newly collected data using the FAR-OCSVM, and we can continuously improve the accuracy of the model by changing the hyperparameters. Data dots in Fig. 8 also show some attributes of the IoT data, like stable, predictable, structured, and standard formats. Not only dimensional reduction but we can also utilize these characteristics of data to set proper parameters. In this case, the learned boundary can filter noise and clean the malicious data. After data dimension reduction (removing low-relationship features), we select a representative subset from the remaining features through the FRHC algorithm before the final intersection operation.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

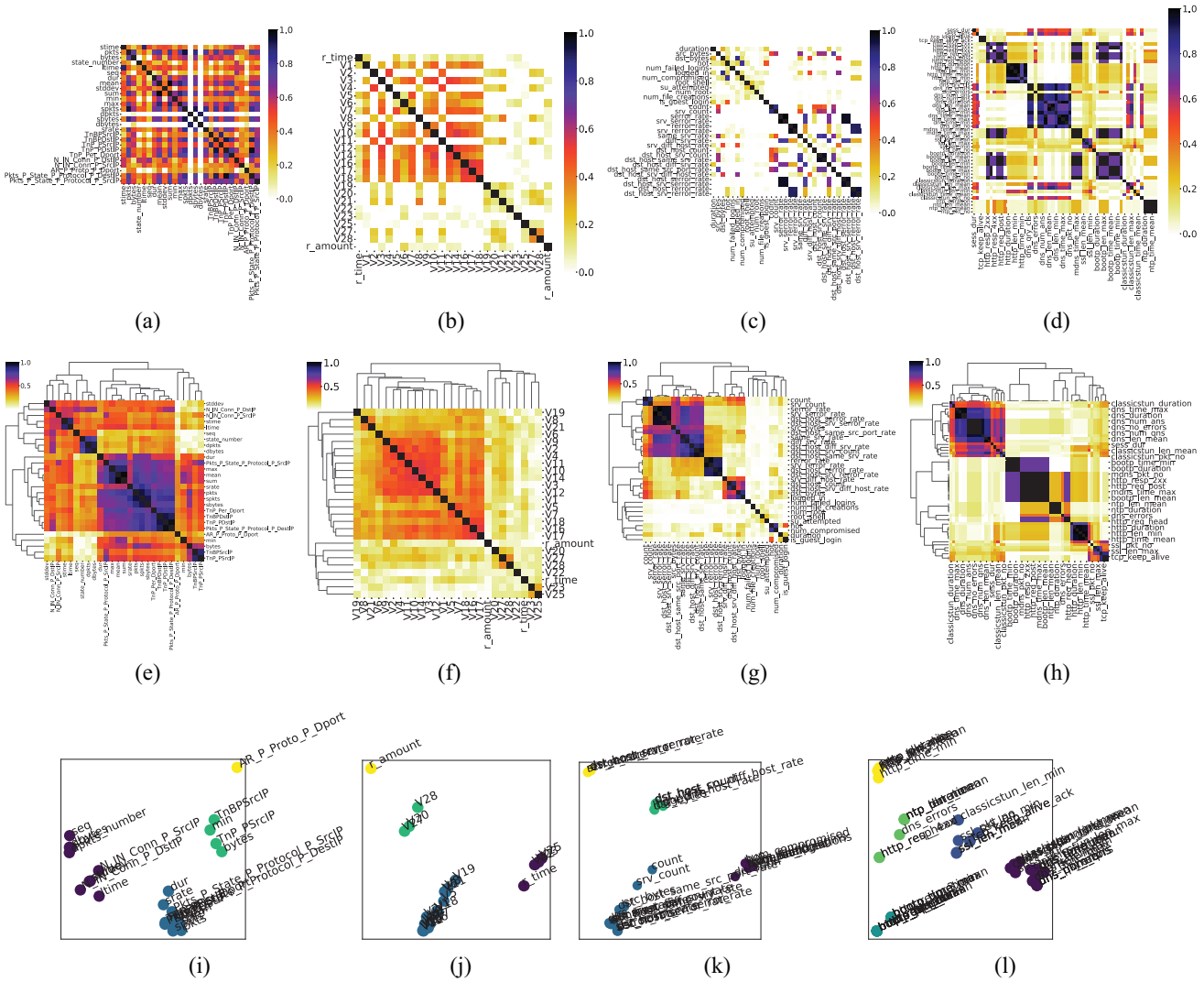

Fig. 9. Feature correlation heatmap (a)–(d). FRHC results (e)–(h) and features cluster results (i)–(l). (a) Bot-IoT. (b) ACC. (c) KDD99. (d) DEFT. (e) BoT-IoT. (f) ACC. (g) KDD99. (h) DEFT. (i) BoT-IoT. (j) ACC. (k) KDD99. (l) DEFT.

2) Local Feature Clustering: Taking the SU as indicators, we give the feature correlation heatmap of four experiment data sets after FAR-OCSVM, as shown in Fig. 9(a)–(d). Colors visualize correlation measures between different features. The more correlated the two features, the darker the color in the heatmap and vice versa. Fig. 9(e)–(h) gives the heatmap results after the FRHC method on four data sets, we can see from the heatmap that features with strong relationships are clustered together. In HFL, the features of each participant are clustered, so the relationship between different features is manifested, observing Fig. 9(a) and (e), (b) and (f), (c) and (g), and (d) and (h). Compared with the original heat map, it can be seen that the dependency between features is more explicit through FRHC. Features that can reflect similar patterns in a data set are clustered together, playing a key role in the HFL.

3) Global Federated Feature Selection: Fig. 9(i)–(l) shows the features cluster results. Features are presented as dots, feature dots with the same color mean that these features are clustered in the same group. The FRHC algorithm can get great results. For example in Fig. 9(i), the experiment shows that the algorithm selects most S_{FG^L} features from the purple and blue-grey clusters, which contain 9 of the 10 best features summarized in [33]. Essential features are correctly divided into priority selection clusters. We can also compare Figs. 9(j) and 7 (c) and Figs. 9(k) and 7(d), features which contain more IG are also grouped together. Users cut the hierarchical tree at the desired level to get cluster assignments, much of it depends on what a user knows as a "natural" grouping to be. There are many popular techniques for selecting the optimal number of clusters like gap statistic, elbow method, SC, and Calinski–Harabasz index. Here, we use the SC method. After FRHC, each HFL participant has cleaned and selected feature sets. These feature sets will be uploaded to the HFL server and do the intersection option as the final federated features part for HFL. The HFL server then gives feedback to each client, so the local devices can know which features should pay more attention when they take part in the HFL.

In the end, FS reduces the interference of HFL and also improves the robustness of HFL. The data dimension reduction

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
PERFORMANCE OF HFL WITH AND WITHOUT FSHFL

Datasets	FSHFL			Without FSHFL		
	Federated Feature set	Accuracy	Training Time (s)	Feature set	Accuracy	Training Time (s)
BoT-IoT IID	23	0.99782	702.791	43	0.99782	749.132
BoT-IoT non-IID	22	0.99745	652.365	43	0.99782	700.762
ACC IID	26	0.96415	509.559	30	0.96415	514.421
ACC non-IID	26	0.95608	494.454	30	0.94595	502.765
KDD99 IID	20	0.99821	732.311	41	0.99740	739.104
KDD99 non-IID	20	0.99643	697.215	41	0.99500	700.161
DEFT IID	87	0.95389	615.890	111	0.95257	619.581
DEFT non-IID	84	0.94466	581.274	111	0.94730	597.365

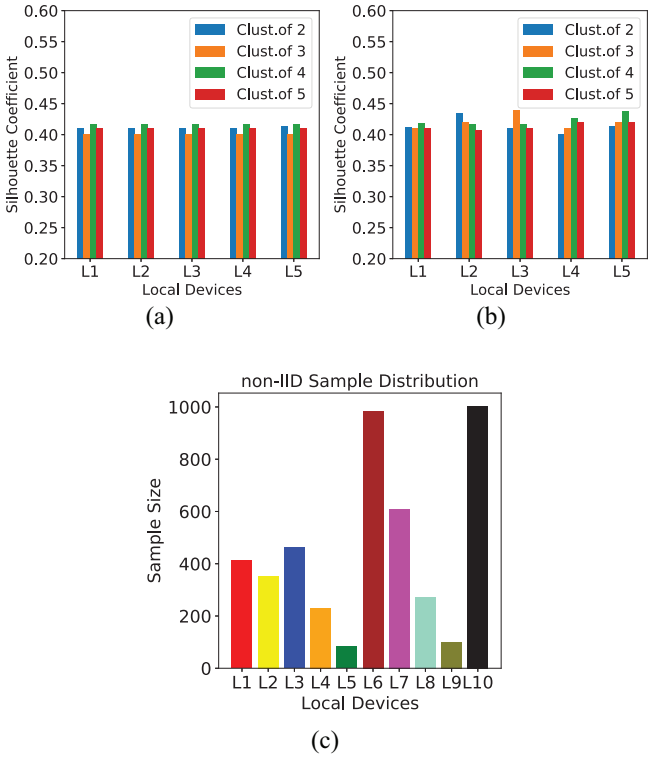

Fig. 10. IID and non-IID in Bot-IoT. (a) IID BoT-IoT SC. (b) non-IID BoT-IoT SC. (c) non-IID sample distribution Bot-IoT.

means less training time, which declines the energy consumed by HFL model training.

4) IID and non-IID Data Test in the HFL Scenario: Fig. 10(a) and (b) illustrates the SC of the same data set (e.g., BoT-IoT) when it is IID and non-IID. Here, in Fig. 10(b), we exemplify the different SC of the five HFL users to show the non-IID characteristics of the data. It can be seen from Fig. 10(a) that the data with the same distribution have the same optimal SC after HC with group number selection. However, in Fig. 10(b), the SC under the non-IID condition has different $\mathbf{K}$ values due to the data distribution. Fig. 10(c) illustrates the non-IID distribution of samples in ten devices. Between each HFL device, the number of features and classes is also different, which means each local data has its characteristics and the distribution of HFL data is non-IID.

Fig. 11 demonstrates the impact of FFS in HFL. Fed-FSHFL is the proposed FFS method which is used in the Fed-Avg. As a comparison, Fed-Avg uses the original data without cleaning and FFS. Plus, Fed-SGD is also considered in the tests without FSHFL, to highlight the EC decrease after applying the FSHFL. In our experiments, the local batch size sets as 10, and each local runs one epoch before one global aggregation. In Fig. 11, from BoT-IoT [Fig. 11(a) and (e)] and DEFT [Fig. 11(d) and (h)] data sets, we can see the proposed FSHFL method does not cause too much loss of accuracy. In ACC [Fig. 11(b) and (f)] and KDD99 [Fig. 11(c) and (g)] data sets, the accuracy of the global model can be improved by up to 1.68%, compared with Fed-Avg without FSHFL.

From the aspect of data distribution (IID or non-IID), we can see that the non-IID data attributes indeed lead to the drop in the accuracy of global models by comparing the experiment results of IID and non-IID in four data sets [Fig. 11(a) and (e), (b) and (f), (c) and (g), (d) and (h)]. We also see a high fluctuation of accuracy in global average operation. This is because the value of the divergences of local and global loss functions, the variance of SGD of IID data sets case is smaller than those of non-IID data sets case. Considering the training time consumption, the proposed FSHFL method reduces the HFL training time in IID and non-IID. We also find in non-IID cases, the optimization of training parameters is harder than in IID. Excessive training after reaching the accuracy requirements will cause more training time and computational cost. Distributed parameter optimization of HFL training is also an important research direction to explore. The number of federated features, global accuracy, and training time in IID and non-IID condition of four data sets are summarized in Table IV. From the table, we can see that the proposed FSHFL can reduce the training time of HFL as high as 6.9%. Little loss of accuracy is also inevitable in data sets. The degree of non-IID configuration in the experiment is not large, so the accuracy rate does not decrease much compared to the IID. In the experiment, the local and global communication are set to be optimal, so the convergence time for non-IID to reach its accuracy is less than that for IID. If the non-IID training is overfitting, the time cost of HFL training will be very high, so it is vital to choose appropriate training parameters and accuracy requirements.

From the perspective of EC, Fig. 12 shows the average EC of the HFL system for 2500 training sessions. Here, we only consider the EC of ten HFL participants training the model and the EC of HFL server aggregation with $B = 10$ (Fed-FSHFL, Fed-Avg), $B = \infty$ (Fed-SGD) and $E = 1$. We set the

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

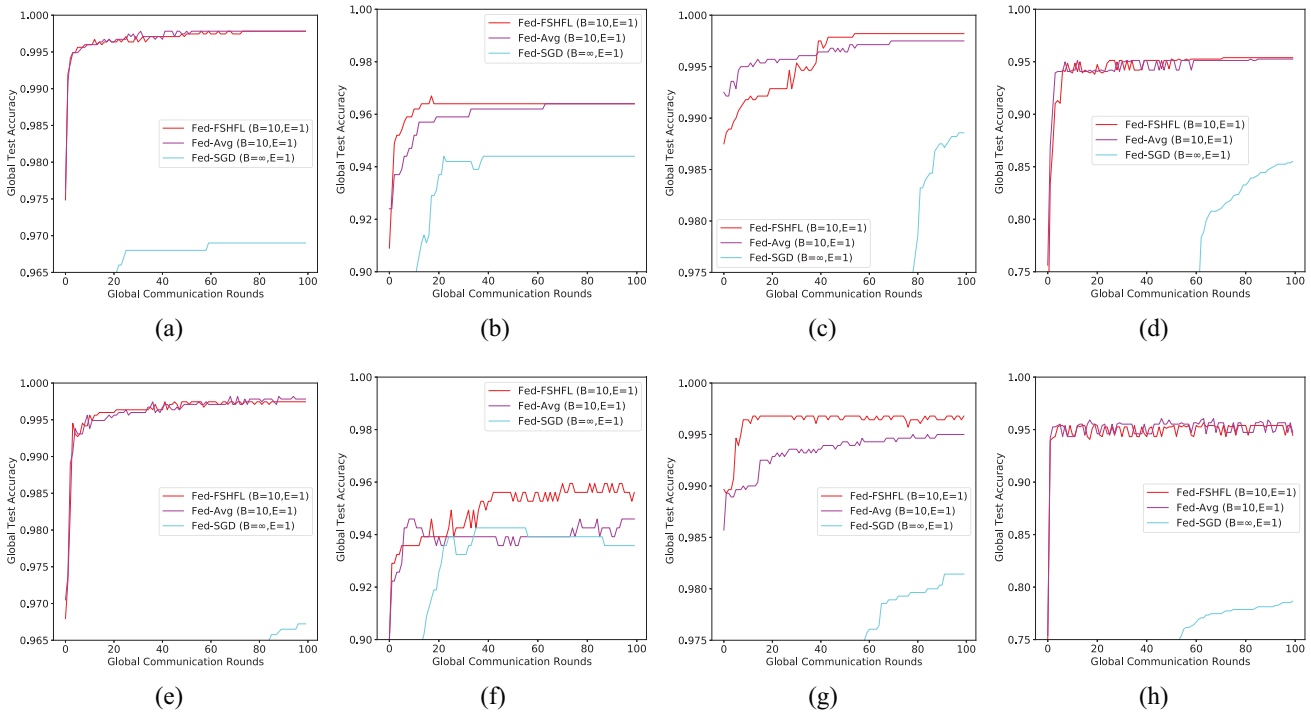

Fig. 11. Evaluation of global test accuracy on four data sets with IID [(a)–(d)] and non-IID [(e)–(h)]. (a) Bot-IoT-IID. (b) ACC-IID. (c) KDD99-IID. (d) DEFT-IID. (e) BoT-IoT-non-IID. (f) ACC-non-IID. (g) KDD99-non-IID. (h) DEFT-non-IID.

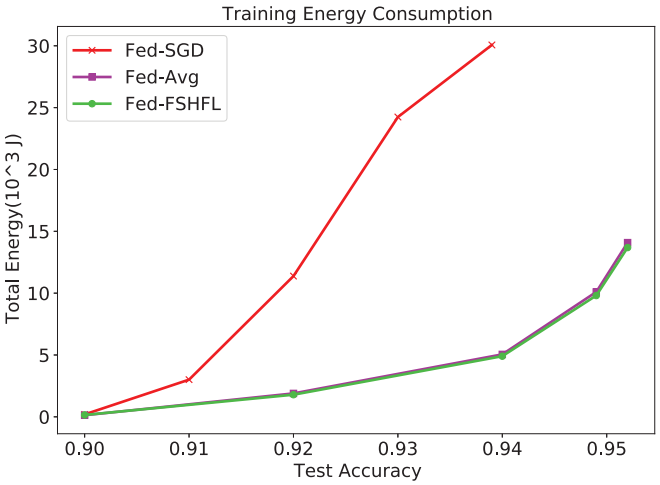

Fig. 12. EC of HFL training.

power of each device as 2.0 W referring to [53]. After each global iteration, the program outputs the global test accuracy, and the training time required to achieve this accuracy also calculates EC according to $W = Pt$. In Fig. 12, the x-axis represents the accuracy of the global model of the system, and the y-axis gives the energy consumed by the system after meeting the training accuracy requirements. The green dotted line represents the proposed FSHFL in this article. It can be seen from the comparison in the figure that our proposed algorithm achieves less EC by processing fewer dimensions when the accuracy requirements are met. Compared with Fed-Avg and Fed-SGD, the proposed FSHFL reduces the average training EC by 2.85% and 68.39%. As the number of iterations increases, fewer dimensions bring faster processing speed, so the EC of the system is significantly reduced in later stages. Fed-SGD converges more slowly with the same learning rate, takes longer to meet the training accuracy requirement, and consumes more energy. It is important to note that the selection of training parameters should also be considered when the data is non-IID. Overfitting problems will bring enormous training time costs to HFL, thereby increasing the EC. The robustness of the HFL system and the global model generalizability will also be influenced by overfitting.

VI. Testbed Deployment

This section will describe how the proposed FSHFL works on the IoT test and the architecture, we already had the platform. Fig. 13 presents the storage of local and global HFL features. Selected features are encrypted and stored in the persistent volume. Due to restrictions in the access mode, the persistent volume is not accessible from claimants that do not hold the permission to access those features. Access mode is governed by (31). Feedback of the HFL features is communicated to the clients to let them know which features to focus more for global training. The persistent volume request is sent to the physical network attached storage which is realized by the Rook Ceph described in Section I-A4).

With the advance of micro-services and role-based access control (RBAC), concepts, such as distributed data lakes, have become increasingly important in computing cloud and edge systems. Fig. 13 presents an example of role-based access between different stakeholders of a network-based storage object. Each stakeholder has predefined read and writes access

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

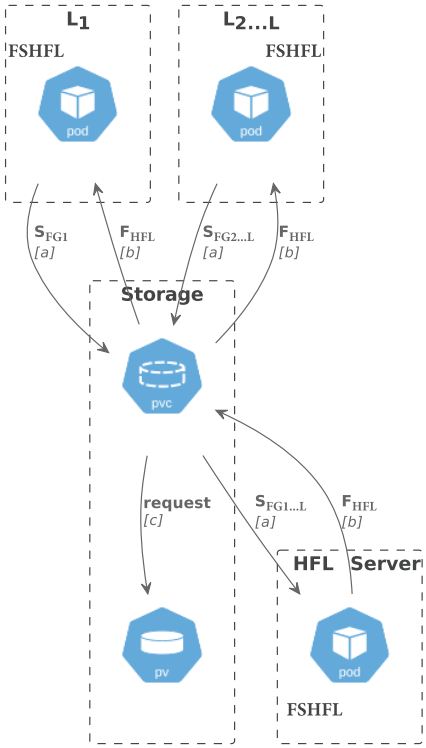

Fig. 13. Testbed architecture: [a] $S_{\mathrm{FG1}\cdots L}$ gathered features from each client. [b] F_{HFL} feedback of the HFL features and [c] is the persistent volume request to the physical storage infrastructure.

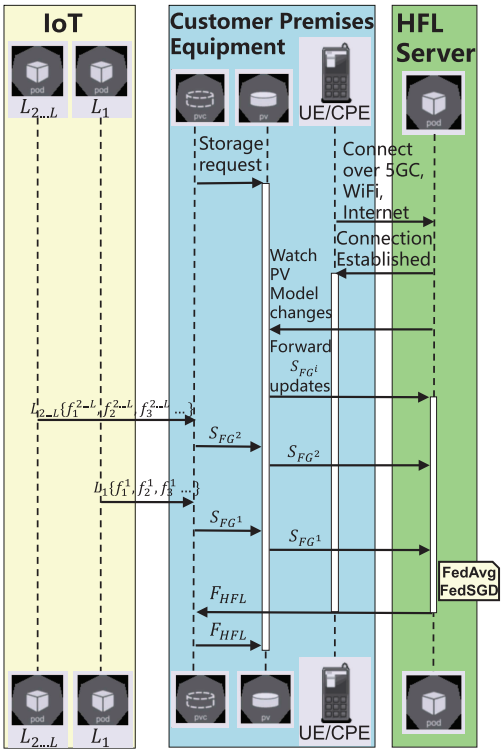

Fig. 14. Testbed connectivity flow.

to storage claims that are being fulfilled by authorized network requests.

The exchange of data can happen across secure data communication links and in real-time. Fig. 14 presents a sequence of events that allows user equipment (UE) to establish connectivity with a cloud HFL server. While the connection is maintained, data from IoT sensors are received on the customer premises equipment (CPE) and communicated with the remote HFL server. The exchange of data is the sequential form of the architecture of FFS in HFL presented in Fig. 2.

VII. Conclusion

FL will play an important role in the next generation of networks and IoT scenarios by collecting and training data without information leakage. Thus, new data cleaning and dimensional reduction techniques become vital to eliminate irrelevant data and find the optimal dimension in order to realize robust FL, especially HFL. In this article, a new data cleaning method and an FFS approach were presented in face of IoT data and the need for finding optimal features in HFL. First, FAR was applied to clean the data for eliminating features that are weakly related to global performance. The relevance weights were used in our proposed FAR-OCSVM to highlight the learning boundary. Then, the FRHC algorithm was designed for explicit feature relationships. From relevant feature groups, we can find the proper features which can be used as the overlapping part of HFL among different participants. In the end, four IoT data sets were tested to find the influence of data dimension reduction on the HFL process. The experimental results showed that our method was able to efficiently clean the data and find good federated features in HFL, which has obvious advantages in terms of accuracy, training time, and system EC. This work also has reference value for ubiquitous learning techniques about data selection and feature extraction in next-generation (6G) networks [54], [55]. However, some limitations of this work in HFL still need to be improved.

1) In the IoT domain, different application scenarios might have different representative features or attributes, a domain-free or generic FS approach may cause backfire. It is important to find a specific IoT application and consider how much overlap can be used as HFL, a third party may be needed.
2) Fairness and equality are not considered in this work. During FS (or not selection) in HFL, a tradeoff between fairness and equality among HFL participants could also be a solution for high-performance HFL.
3) Classical feature wrapper methods like forward selection, backward elimination [e.g., recursive feature elimination (RFE)], and step-wise selection should also be improved by adapting to the HFL. More complicated work is ongoing.

References

[1] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: A review," *Complex Intell. Syst.*, vol. 8, pp. 2663–2693, Jan. 2022.
[2] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[3] X. Chen, G. Yuan, W. Wang, F. Nie, X. Chang, and J. Z. Huang, "Local adaptive projection framework for feature selection of labeled and unlabeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6362–6373, Dec. 2018.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

[4] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[5] M. Ishtiaq, N. Saeed, and M. A. Khan, "Edge computing in IoT: A 6G perspective," 2021, *arXiv:2111.08943*.

[6] N. H. Mahmood, H. Alves, O. A. López, M. Shehab, D. P. M. Osorio, and M. Latva-Aho, "Six key features of machine type communication in 6G," in *Proc. 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[7] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11891–11915, Aug. 2021.

[8] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[9] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101863.

[10] S. Egea, A. Rego Mañez, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Intelligent IoT traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1616–1624, Jun. 2018.

[11] S. Hou, Y. Chu, and J. Fei, "Intelligent global sliding mode control using recurrent feature selection neural network for active power filter," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7320–7329, Aug. 2021.

[12] J. Krawczuk and T. Łukaszuk, "The feature selection bias problem in relation to high-dimensional gene data," *Artif. Intell. Med.*, vol. 66, pp. 63–71, Jan. 2016.

[13] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[14] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[15] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1–24, Jan. 2022.

[16] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, Sep. 2020.

[17] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[18] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[19] *IEEE Guide for Architectural Framework and Application of Federated Machine Learning*, IEEE Standard 3652.1–2020, pp. 1–69, 2021.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[21] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*.

[22] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[23] Y. Hu, Y. Zhang, D. Gong, and X. Sun, "Multi-participant federated feature selection algorithm with particle swarm optimization for imbalanced data under privacy protection," *IEEE Trans. Artif. Intell.*, early access, Jan. 25, 2022, doi: 10.1109/TAI.2022.3145333.

[24] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: A feature fusion approach," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 175–179.

[25] P. Cassará, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9937–9950, Sep. 2022.

[26] S. Banerjee, E. Elmroth, and M. Bhuyan, "Fed-FiS: A novel information-theoretic federated feature selection for learning stability," in *Proc. Int. Conf. Neural Inf. Process.*, 2021, pp. 480–487.

[27] Y. Qin and M. Kondo, "Federated learning-based network intrusion detection with a feature selection approach," in *Proc. Int. Conf. Elect. Commun. Comput. Eng. (ICECCE)*, 2021, pp. 1–6.

[28] L. Wang, Q. Pang, S. Wang, and D. Song, "FED-χ^2: Privacy preserving federated correlation test," 2021, *arXiv:2105.14618*.

[29] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 1302–1308.

[30] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017.

[31] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 944–956, Apr. 2018.

[32] B. Chakraborty, D. M. Divakaran, I. Nevat, G. W. Peters, and M. Gurusamy, "Cost-aware feature selection for IoT device classification," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11052–11064, Jul. 2021.

[33] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-iot dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[34] 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 Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[35] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "Analysis of KDD'99 intrusion detection dataset for selection of relevance features," in *Proc. World Congr. Eng. Comput. Sci. (WCECS)*, vol. 1, 2010, pp. 20–22.

[36] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, "Big data cleaning based on mobile edge computing in industrial sensor-cloud," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1321–1329, Feb. 2020.

[37] S. Caldas et al., "Leaf: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

[38] L. Ma, Q. Pei, L. Zhou, H. Zhu, L. Wang, and Y. Ji, "Federated data cleaning: Collaborative and privacy-preserving data cleaning for edge intelligence," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6757–6770, Apr. 2021.

[39] G. Han, J. Tu, L. Liu, M. Martínez-García, and Y. Peng, "Anomaly detection based on multidimensional data processing for protecting vital devices in 6G-enabled massive IIoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5219–5229, Apr. 2021.

[40] H. Y. Teh, I. Kevin, K. Wang, and A. W. Kempa-Liehr, "Expect the unexpected: Unsupervised feature selection for automated sensor anomaly detection," *IEEE Sensors J.*, vol. 21, no. 16, pp. 18033–18046, Aug. 2021.

[41] H.-J. Xing, M.-H. Ha, D.-Z. Tian, and B.-G. Hu, "A novel support vector machine with its features weighted by mutual information," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, 2008, pp. 315–320.

[42] S. Liu, X. Chen, W. Liu, J. Chen, Q. Gu, and D. Chen, "FECAR: A feature selection framework for software defect prediction," in *Proc. IEEE 38th Annu. Comput. Softw. Appl. Conf.*, 2014, pp. 426–435.

[43] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[44] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, "Overcoming noisy and irrelevant data in federated learning," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 5020–5027.

[45] L. Nagalapatti, R. S. Mittal, and R. Narayanam, "Is your data relevant?: Dynamic selection of relevant data for federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7859–7867.

[46] F. Li et al., "An improved clear cell renal cell carcinoma stage prediction model based on gene sets," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–15, 2020.

[47] E. P. Ahi et al., "Comparative transcriptomics reveals candidate carotenoid color genes in an east african cichlid fish," *BMC Genomics*, vol. 21, no. 1, pp. 1–15, 2020.

[48] A. M. Jarman, "Hierarchical cluster analysis: Comparison of single linkage, complete linkage, average linkage and centroid linkage method," Georgia Southern Univ., Statesboro, GA, USA, 2020.

[49] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," 2021, *arXiv:2102.02079*.

[50] C. Xu, S. Liu, Z. Yang, Y. Huang, and K.-K. Wong, "Learning rate optimization for federated learning exploiting over-the-air computation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3742–3756, Dec. 2021.

[51] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.

[52] V. Thangavelu, D. M. Divakaran, R. Sairam, S. S. Bhunia, and M. Gurusamy, "DEFT: A distributed IoT fingerprinting technique," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 940–952, Feb. 2019.
[53] Y. Gao et al., "End-to-end evaluation of federated learning and split learning for Internet of Things," 2020, *arXiv:2003.13376*.
[54] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Mar. 2020.
[55] Y. Xiao, G. Shi, and M. Krunz, "Towards ubiquitous AI in 6G with federated learning," 2020, *arXiv:2004.13563*.

Xunzheng Zhang (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in communication engineering from Shandong University, Jinan, China. He is currently pursuing the Ph.D. degree with the University of Bristol, Bristol, U.K.

He has experience with Internet of Things and edge–cloud collaboration testbed deployment under Industry 4.0. His current research interests include 6G massive Internet of Things, federated learning, edge computing, and machine learning.

He has the qualification of electronic technology application engineer, owns patents and won the Best Paper Award of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2020, London, U.K.

Alex Mavromatis (Member, IEEE) received the B.S. degree in computer science, ATEI, Thessaloniki, Greece, in 2016, and the Ph.D. degree in 5G Internet of Things (IoT) with the University of Bristol, Bristol, U.K., in 2020.

He is a Senior Research Associate with the University of Bristol. He has experience with IoT and 5G projects, such as smart cities and 5G testbed deployment and experimentation, funded by the EU and U.K. Government. His current research interests include industrial IoT deployments, customized Industry 4.0 architectures, smart home architectures, software-defined networking/NFV, cloud/edge networking platforms, data management, and production quality assurance of machine learning models.

Antonis Vafeas (Member, IEEE) received the Ph.D. degree with the University of Bristol, Bristol, U.K., in 2021.

He is a Senior Research Associate with 5G Network Systems Engineering and Administration, University of Bristol. He specializes in multimodal sensing technology and embedded hardware.

Reza Nejabati (Senior Member, IEEE) is the Chair Professor of Intelligent Networks and the Head of the High-Performance Network Group with the Department of Electrical and Electronic Engineering, University of Bristol, Bristol, U.K. He is also a Visiting Professor and a Cisco Chair with the Cisco Centre for Intent-Based Networking, Curtin University, Perth, Australia. He has established successful and internationally recognised research activities in "Autonomous and Intent Based Networks" as well as "Quantum Networks." Building on his research, he co-founded a successful start-up company (Zeetta Networks Ltd.). It has currently 25 employees and 6m VC and external funding.

Mr. Nejabati's research received the prestigious IEEE Charles Kao Award in 2016 and has done important contributions in 5G, smart city, quantum communication, and future Internet experimentation.

Dimitra Simeonidou (Fellow, IEEE) is a Full Professor with the University of Bristol, Bristol, U.K., the Co-Director of the Bristol Digital Futures Institute, Bristol, and the Director of Smart Internet Lab, Bristol. She is increasingly working with social sciences on topics of digital transformation for society and businesses. She has been the Technical Architect and the CTO of the Smart City Project Bristol Is Open. She is currently leading the Bristol City/Region 5G urban pilots. She has authored and coauthored over 600 publications, numerous patents, and several major contributions to standards. She has been the co-founder of two spin-out companies, the latest being the University of Bristol VC funded spin-out Zeetta Networks, delivering SDN solutions for enterprise and emergency networks. Her research is focusing in the fields of high-performance networks, programmable networks, wireless-optical convergence, 5G/6G, and smart city infrastructures.

Prof. Simeonidou is a Fellow of Royal Academy of Engineering and a Royal Society Wolfson Scholar.

Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on October 22,2024 at 12:50:05 UTC from IEEE Xplore. Restrictions apply.