

Attack-aware Federated Learning via Prototype-guided Cross-cluster Knowledge Transfer for Label-space Heterogeneous Intrusion Detection

Marwa Badrouni

`badrouni.i.marwa@gmail.com`

Institut Supérieur de Gestion de Tunis

Chaker Katar

Institut Supérieur de Gestion de Tunis

Wisse Inoubli

Université d'Artois

Zahra Kodia

Institut Supérieur de Gestion de Tunis

Research Article

Keywords: Federated Learning, Label-Space Heterogeneity, Intrusion Detection Systems, Attack-Aware Client Clustering, Knowledge Transfer

Posted Date: October 19th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11173311/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Attack-aware Federated Learning via Prototype-guided Cross-cluster Knowledge Transfer for Label-space Heterogeneous Intrusion Detection

Marwa Badrouni¹, Chaker Katar¹, Wissem Inoubli², Zahra Kodia¹

¹Department of Computer Science SMART LAB, ISG, University of Tunis, Tunis, Tunisia .

²CNRS, UMR 8188, Computer Science Research Institute of Lens (CRIL), Université d'Artois , 62300 Lens, France .

Contributing authors: badrouni.marwa@gmail.com;
chaker.katar@gmail.com; wissem.inoubli@univ-artois.fr;
zahra.kodia@isg.rnu.tn;

Abstract

Federated learning (FL) enables collaborative intrusion detection without requiring the direct sharing of distributed network data. However, label-space heterogeneity, where clients observe different attack families, challenges conventional aggregation and existing clustered collaboration strategies, which may either mix incompatible knowledge or isolate complementary attack expertise. To address this challenge, we propose Attack-Fed, a two-level attack-aware federated learning method that jointly considers attack-space compatibility and complementary knowledge across clients. Unlike collaboration strategies restricted to independent client clusters, Attack-Fed combines attack-aware clustering with selective cross-cluster knowledge transfer. At the first level, an Attack Category Profile (ACP) characterizes clients' attack expertise and guides Affinity Propagation to identify compatible attack spaces. At the second level, prototype-guided knowledge transfer exploits complementary attack relationships between clusters to reduce knowledge isolation while limiting the exchange of potentially irrelevant knowledge. This two-level design enables specialized intra-cluster learning and targeted inter-cluster knowledge sharing under heterogeneous label spaces. Extensive experiments on intrusion detection datasets evaluate Attack-Fed against conventional and clustered federated learning baselines, demonstrating its performance under varying degrees of label-space heterogeneity.

Keywords: Federated Learning, Label-Space Heterogeneity, Intrusion Detection Systems, Attack-Aware Client Clustering, Knowledge Transfer.

1 Introduction

Cybersecurity has become one of the most critical challenges of the digital era as modern societies increasingly rely on interconnected computer networks, cloud infrastructures, Internet of Things (IoT) devices, and intelligent cyber-physical systems [1]. The rapid expansion of these interconnected environments has considerably increased the attack surface available to malicious actors, while cyber threats have become increasingly diverse, sophisticated, and difficult to detect. Attacks such as distributed denial-of-service (DDoS), denial-of-service (DoS), botnets, brute-force attacks, infiltration, spoofing, and web-based attacks can severely disrupt network services, compromise sensitive information, and threaten the reliability of critical infrastructures. Consequently, developing effective and adaptive mechanisms for detecting malicious network activities has become an important research priority [2]. Network Intrusion Detection Systems (NIDSs) constitute a fundamental line of defense by continuously monitoring network traffic and identifying potentially malicious activities. Traditional signature-based IDSs are effective for previously known attack patterns but are inherently limited [3] when facing novel or evolving threats. This limitation has motivated the extensive adoption of machine learning (ML) and deep learning (DL) techniques for intrusion detection [4], owing to their ability to learn complex patterns from network traffic and automatically distinguish benign from malicious behaviors. Although data-driven IDSs have achieved promising detection performance, their effectiveness strongly depends on access to sufficiently diverse and representative training data [5].

In practical cybersecurity environments, however, network traffic is naturally distributed across organizations, infrastructures, and IoT deployments [6]. Centralizing such data for training a common intrusion detection model is often impractical because network traffic may contain sensitive operational information and may be subject to privacy requirements, confidentiality constraints, and organizational policies. FL has therefore emerged as a promising paradigm for enabling collaborative model training [7] without requiring participants to directly share their raw data. In a typical FL setting [8], clients locally train their models on private network traffic and exchange model information with a coordinating server, allowing knowledge to be aggregated across distributed participants while keeping the underlying data local. Despite these advantages, the effectiveness of FL-based intrusion detection is strongly affected by the statistical heterogeneity of participating clients [9]. In realistic cybersecurity environments, clients do not necessarily observe similar traffic patterns or attack behaviors. More importantly, different network environments may be exposed to substantially different attack landscapes. A client deployed in one environment may frequently observe DDoS and DoS attacks, whereas another may primarily encounter reconnaissance,

botnet, spoofing, or web attacks. Consequently, clients may not only exhibit different label distributions but may also possess fundamentally different local label spaces [10].

Statistical heterogeneity in FL [8] can arise from several sources, including quantity heterogeneity [11], feature distribution heterogeneity [12], label distribution heterogeneity [13], and label-space heterogeneity [14]. Quantity heterogeneity refers to differences in the amount of training data available across clients. Feature distribution heterogeneity occurs when clients observe different data distributions over a shared feature space. Label distribution heterogeneity describes variations in class proportions among clients that share the same label space. In contrast, label-space heterogeneity occurs when clients observe different subsets of the global label space. Although these forms of heterogeneity may coexist in practical IoT environments, this work specifically focuses on label-space heterogeneity in federated intrusion detection. In the considered setting, clients share a common classification task and feature representation, while their local datasets may contain different subsets of attack families. This distinction is important because our framework targets differences in the attack categories available to each client and the resulting incompatibility of their local learning objectives, rather than feature distribution shifts or differences in local dataset sizes.

Formally, if $\mathcal{Y}$ denotes the global set of attack categories and $\mathcal{Y}_i$ denotes the attack categories observed by client i , then label-space heterogeneity occurs when

$$\mathcal{Y}_i \neq \mathcal{Y}_j, \quad i \neq j. \quad (1)$$

This setting differs from conventional label distribution skew, where clients share the same label space but exhibit different class proportions. Under label-space heterogeneity, some attack categories may be completely absent from one client’s local training data while being present in another client’s data. Consequently, participating clients may effectively optimize different local classification objectives despite sharing the same overall intrusion detection task.

This characteristic creates a critical limitation for conventional federated aggregation mechanisms, as discussed in subsequent sections. These methods implicitly assume that local models contribute knowledge toward a sufficiently compatible global objective [15]. When clients have substantially different attack spaces, directly averaging their model updates may introduce conflicting optimization signals. Knowledge specialized for attacks observed by one client can interfere with the learning process of another client that has never observed those categories [5]. Such indiscriminate aggregation can therefore weaken locally relevant decision boundaries, particularly for rare or completely unseen attack families, and may ultimately degrade the global intrusion detection capability.

Existing approaches to non-IID federated learning have mainly addressed heterogeneity through techniques such as proximal regularization, personalized models, representation alignment, or client clustering based on model updates and learned representations [1]. Although these strategies can alleviate certain forms of statistical heterogeneity, they do not necessarily characterize the compatibility of clients’ local attack spaces explicitly. In particular, grouping clients solely according to model similarity does not necessarily reveal whether they possess compatible attack expertise [16]. Moreover, simply isolating heterogeneous clients into independent clusters

introduces another limitation. Once clients are separated, complementary knowledge about different attack families may remain confined within individual clusters. Thus, an effective collaboration strategy should address two complementary requirements: preserving collaboration among clients with compatible attack spaces while enabling selective knowledge sharing between clients or clusters possessing complementary attack knowledge.

To address these challenges, we propose Attack-Fed, a two-level attack-aware federated learning method for heterogeneous intrusion detection. The first level explicitly models the attack expertise of each client through an ACP, which summarizes the presence, frequency, and diversity of locally observed attack families. These profiles provide an attack-aware representation of client expertise and are used to quantify compatibility between clients. Based on the resulting similarity matrix, an attack-aware client clustering mechanism using Affinity Propagation identifies groups of clients with compatible local attack spaces, enabling more coherent intra-cluster model aggregation.

The second level addresses the knowledge isolation that may arise from cluster-wise collaboration. Although attack-aware clustering reduces potentially harmful interactions between incompatible clients, independently trained clusters may still possess complementary knowledge about attack families that are absent from other clusters. We therefore construct an Attack Relationship Graph to model relationships between cluster-level attack knowledge. A prototype-guided knowledge propagation mechanism exploits these relationships to selectively transfer complementary attack knowledge across clusters rather than indiscriminately aggregating their complete models. A subsequent global knowledge completion mechanism integrates the transferred knowledge to improve the overall detection capability while maintaining the specialization achieved through attack-aware clustering.

The main contributions of this work are summarized as follows:

- We formulate and explicitly address label-space heterogeneity in federated intrusion detection, where clients may observe different subsets of attack families and consequently optimize different local classification objectives.
- We introduce an ACP that captures the local attack expertise of each client through the presence, frequency, and diversity of observed attack families. The ACP is used to construct an attack-aware client similarity matrix for identifying compatible collaboration groups.
- We develop an attack-aware client clustering mechanism based on Affinity Propagation to automatically organize clients according to the compatibility of their local attack spaces, enabling more coherent intra-cluster knowledge aggregation.
- We propose an Attack Relationship Graph and a prototype-guided cross-cluster knowledge propagation mechanism to address knowledge isolation caused by clustering and selectively exchange complementary attack knowledge across heterogeneous clusters.
- We conduct extensive experiments under heterogeneous label-space settings, including comparative, ablation, and robustness analyses, to evaluate the effectiveness of the proposed framework and the contribution of its individual components.

The remainder of this paper is organized as follows. Section 2 reviews related work on federated learning for intrusion detection, clustered federated learning for non-IID data, and label-space heterogeneity and knowledge transfer. Section 3 presents the proposed Attack-Fed method, including the attack-aware clustering and knowledge-guided cross-cluster transfer mechanisms. Section 4 describes the experimental setup and evaluation methodology. Section 5 presents and discusses the main experimental results. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2 Related Work

The related work is organized into three subsections. The first reviews federated learning approaches for intrusion detection. The second examines clustered federated learning methods developed to address non-IID data. The third focuses specifically on label-space heterogeneity and knowledge transfer, which are directly related to the challenges addressed by our proposed Attack-Fed method.

2.1 Federated Learning for Intrusion Detection

Federated learning has increasingly been investigated as a privacy-preserving paradigm for distributed intrusion detection, mainly because network traffic is naturally generated and stored across different organizations, edge devices, and IoT environments. Instead of requiring the centralized collection of potentially sensitive network traffic, FL enables participating clients to train local intrusion detection models and exchange model information with a coordinating server. Early studies primarily focused on demonstrating the feasibility of combining conventional machine learning or deep learning architectures with the federated learning paradigm.

Sarhan et al. [17] proposed a federated learning-based cyber threat intelligence sharing scheme for network intrusion detection. Their approach enables multiple organizations to collaboratively improve intrusion detection while keeping raw network traffic locally stored. Zhao et al. [18] proposed an FL-assisted LSTM framework for intelligent intrusion detection. Their approach trains LSTM models locally at distributed servers and periodically aggregates the resulting model parameters at a central server. The study demonstrated that federated training can achieve competitive intrusion detection performance while avoiding direct transmission of local training data. Similarly, Chen et al. [19] proposed FedAGRU, an attention-based gated recurrent unit architecture for intrusion detection in wireless edge networks. In addition to federated model training, their method incorporates an attention mechanism to prioritize informative client updates and reduce communication overhead. These early approaches established the feasibility of federated intrusion detection while highlighting the importance of communication efficiency and distributed model optimization. Man et al. [20] further investigated federated intrusion detection in edge-assisted IoT environments and proposed FedACNN, which integrates a convolutional neural network with federated learning and an attention mechanism. Their results showed that federated training can provide effective intrusion detection while reducing the number of communication rounds compared with conventional federated optimization. In

a broader study, Campos et al. [21] evaluated federated intrusion detection in IoT environments under different data distribution settings. Their analysis demonstrated that the effectiveness of federated IDS is strongly influenced by the distribution of local data and emphasized the challenges associated with non-identically distributed client datasets. These findings are particularly relevant because the statistical diversity of network environments is an inherent characteristic of practical federated intrusion detection.

Several subsequent studies have focused more explicitly on improving federated optimization and anomaly detection. Agrawal et al. [22] provided a comprehensive analysis of federated learning for intrusion detection systems, discussing its potential advantages as well as challenges related to communication cost, non-IID data, security, and model aggregation. Janati Idrissi et al. [23] proposed Fed-ANIDS, a federated anomaly-based network intrusion detection framework that combines federated learning with autoencoder-based detection. Their evaluation also showed the importance of the aggregation strategy under distributed and heterogeneous conditions, with FedProx providing improvements over FedAvg in their considered settings. More recently, Sun et al. [24] proposed FedMADE, a robust federated learning framework for IoT intrusion detection that dynamically aggregates local models according to device traffic patterns. In particular, FedMADE introduces a clustering mechanism to group devices according to their traffic characteristics and uses their contribution to guide model aggregation, aiming to alleviate the adverse effects of heterogeneous attack distributions. Recent work has also investigated federated intrusion detection in resource-constrained edge environments. For example, the data-efficient federated learning framework proposed for edge network intrusion detection in [25] considers heterogeneous network distributions while keeping raw training traffic local. Such studies reinforce the practical relevance of federated intrusion detection and demonstrate that privacy preservation alone is not sufficient: the heterogeneity of distributed network environments must also be explicitly considered during collaborative optimization.

Overall, existing FL-based intrusion detection approaches have successfully demonstrated the feasibility of collaborative detection without centralized access to raw network traffic. However, most of these studies focus primarily on model architecture, aggregation optimization, communication efficiency, or general statistical heterogeneity. Even when client clustering or adaptive aggregation is introduced, client relationships are generally inferred from model updates, traffic characteristics, or prediction behavior rather than from the actual attack categories observed by each client [26]. Consequently, the semantic compatibility of clients local attack expertise remains largely unexplored. This limitation motivates the need for an attack-aware collaboration strategy that explicitly considers the structure of clients local attack spaces before determining how knowledge should be aggregated and shared.

2.2 Clustered Federated Learning for Non-IID Data

Clustered Federated Learning (CFL) has emerged as an effective strategy for mitigating the adverse effects of statistical heterogeneity in federated learning. Unlike conventional federated learning, which aggregates the contributions of all participating clients into a single global model, CFL first identifies groups of clients with similar

learning characteristics and subsequently learns specialized models for the resulting groups. The underlying intuition is that clients with more similar local data distributions or learning objectives are more likely to benefit from collaborative optimization. Consequently, clustering can reduce the interference caused by heterogeneous local updates and provide more personalized federated models. One of the foundational CFL approaches was introduced by Sattler et al. [27] through clustered federated learning, which exploits the geometric properties of local model updates to recursively partition clients into groups with jointly trainable distributions. The proposed approach performs clustering based on the similarity between client updates and allows each resulting group to learn a specialized model. This work demonstrated that separating clients with substantially different optimization behavior can improve federated learning under heterogeneous data distributions. Ghosh et al. [28] proposed the Iterative Federated Clustering Algorithm (IFCA), which formulates CFL as a joint estimation problem involving both client cluster assignments and cluster-specific model parameters. In each iteration, clients identify the model corresponding to their local objective, while the server updates the models associated with the different clusters. Unlike post-training clustering approaches, IFCA integrates the clustering process directly into federated optimization and provides theoretical convergence guarantees under suitable conditions. This formulation highlights the importance of learning multiple specialized models when participating clients do not share an identical statistical objective. Briggs et al. [29] proposed FL+HC, which applies hierarchical clustering to local model updates. Clients are initially trained through conventional federated optimization, after which the similarity between local updates and the global model is used to construct a hierarchical clustering structure. The resulting client groups are then trained independently using specialized models. Their experiments demonstrated that hierarchical client organization can improve convergence and accuracy in several non-IID settings compared with learning a single global model. Duan et al. [30] proposed FedGroup, an efficient clustered federated learning framework that groups clients according to the similarity of their optimization directions. To overcome the difficulty of clustering high-dimensional model updates, the authors introduced a decomposed data-driven similarity measure and designed a mechanism for handling newly joining clients. FedGroup therefore illustrates that the choice of the client similarity representation and distance measure can substantially influence the efficiency and quality of federated clustering. More recently, Islam et al. [31] introduced FedClust, a weight-driven client clustering framework designed to reduce the communication and computational costs associated with repeated cluster formation. Rather than requiring a large number of communication rounds to stabilize the clustering structure, FedClust uses selected local model weights to estimate similarities between clients and performs clustering in a one-shot manner. The method also addresses the limitation of approaches that require the number of clusters to be specified in advance. These characteristics demonstrate the continued interest in developing more efficient and adaptive client-clustering mechanisms for heterogeneous federated learning. CFL has also received increasing attention in federated intrusion detection. FedMSP-SPEC [1] addresses non-IID intrusion detection by combining a multi-scale parallel convolutional network with adaptive client clustering. In this framework, each client

is represented using an average soft-prediction vector obtained from its local data. Pairwise Euclidean distances between these representations are then used to construct client clusters, while the silhouette coefficient is employed to determine the number of clusters. The resulting groups are used for hierarchical aggregation, first within clusters and subsequently across clusters. Although this approach demonstrates the effectiveness of clustering for reducing aggregation bias in federated intrusion detection, its notion of client similarity is derived from model predictions rather than from the attack categories observed by each client.

Similarly, ClusterFed [32] investigates clustering-based federated intrusion detection in decentralized and heterogeneous network environments. The method combines soft clustering with a relational learning objective and incorporates knowledge distillation at both local and global levels to maintain consistent cluster representations. This approach demonstrates that combining clustering and knowledge distillation can improve the robustness of federated intrusion detection under heterogeneous data distributions. Nevertheless, the clustering mechanism remains primarily representation-driven and does not explicitly characterize the semantic compatibility of the attack categories observed across clients. Overall, existing clustered federated learning methods demonstrate that separating statistically incompatible clients can alleviate the negative effects of non-IID data and improve model specialization. However, most existing approaches determine client similarity from local model parameters, gradients, optimization directions, soft predictions, or learned representations. Such information reflects the behavior of the learned models but does not explicitly describe the attack expertise available at each client. This distinction is particularly important in intrusion detection, where two clients may exhibit similar model representations while observing substantially different attack categories, or may possess complementary expertise that could be beneficial to one another. Therefore, an important research gap remains in developing a client-clustering strategy that explicitly considers the compatibility of locally observed attack categories rather than relying solely on model- or representation-level similarity. Addressing this gap requires a label-aware characterization of client attack expertise, which motivates the attack-aware client organization introduced in our proposed framework.

2.3 Label-Space Heterogeneity and Knowledge Transfer

Federated learning has traditionally been developed under the assumption that participating clients share a common prediction space. However, in practical distributed environments, clients may observe different subsets of categories, making direct aggregation of their learned knowledge more challenging. Recent studies have therefore investigated alternative strategies for improving collaboration under heterogeneous label spaces, including prototype-based learning, personalized representations, classifier collaboration, and knowledge distillation. FedProto [33] introduced a prototype-based federated learning paradigm in which clients communicate class-level prototypes instead of directly exchanging model gradients. Local prototypes are aggregated at the server to construct global prototypes, which are then returned to clients to regularize local training. This design reduces the dependence on model parameters

and provides a compact representation of class-related knowledge. However, the prototype aggregation mechanism primarily assumes that corresponding classes can be identified across clients. Consequently, it does not explicitly characterize the compatibility between clients with different local label spaces or determine which clients should collaborate more strongly according to their observed classes. Personalized federated learning has also been explored as a way to mitigate the negative effects of label heterogeneity. FedPAC [34] proposes a framework based on feature alignment and classifier collaboration, where the representation learning component is shared while classifier aggregation is personalized according to the characteristics of each client. This separation allows clients to preserve part of their local specificity while benefiting from shared representations [35]. Nevertheless, the approach mainly addresses heterogeneity through representation alignment and personalized classifier collaboration rather than explicitly identifying groups of clients according to the semantic compatibility of their local label spaces.

More recently, FedCoSR [36] investigates personalized federated learning under label heterogeneity by introducing contrastive shareable representations. The method considers shallow model parameters and local representations as shareable information and employs contrastive learning between local and global representations to enrich local knowledge. An adaptive local aggregation mechanism is further introduced to account for differences in local data availability. While this approach improves collaboration under heterogeneous label distributions, its main objective is to learn more transferable representations rather than explicitly organizing clients according to the attack categories available in their local datasets. Sub-field of reserach focuses on knowledge distillation to transfer information without directly aggregating complete local models. FedGPD [37] introduces global prototype distillation, where aggregated global class prototypes are used as distilled knowledge to guide local training. Instead of relying exclusively on the global model’s logits, the method exploits prototypes as compact representations of class-level information and incorporates them into the local learning objective. This strategy demonstrates that prototypes can serve as an effective medium for knowledge transfer under heterogeneous data distributions. However, the knowledge is propagated through global prototypes and is not conditioned on the relationships between groups of clients possessing different types of knowledge. FedTKD [38] further investigates knowledge distillation for heterogeneous federated learning by introducing an adaptive knowledge fusion mechanism. The method aims to identify unreliable or potentially malicious client information before incorporating transferred knowledge into the global model. Such approaches demonstrate the potential of knowledge distillation for federated environments where direct model aggregation is undesirable. Nevertheless, the transferred knowledge is primarily determined by client-level information and trustworthiness, rather than by explicit relationships between heterogeneous groups of clients or by the complementary nature of their learned class knowledge. Other studies have investigated the label-space problem more directly. Breitholtz et al. [39] studied federated learning with heterogeneous and private label sets, showing that reducing the number of labels available to individual clients can substantially degrade federated model performance. Their study further distinguishes between settings where the label sets are publicly known and those

Table 1 Comparison of representative federated learning approaches with respect to data heterogeneity, client organization, knowledge sharing, and aggregation mechanisms.

Method	Non-IID	Label Heterogeneity	Client Clustering	Knowledge Sharing Unit	Aggregation Mechanism
FedAvg [40]	✓	–	–	Model parameters	Weighted parameter averaging
FedProx [41]	✓	–	–	Model parameters	Proximal regularized aggregation
CFL [27]	✓	–	✓	Model parameters	Cluster-wise model aggregation
IFCA [28]	✓	–	✓	Model parameters	Cluster-specific aggregation
FedGroup [30]	✓	–	✓	Model parameters	Group-wise aggregation
FedClust [31]	✓	–	✓	Model parameters	Cluster-wise aggregation
FedMSP-SPEC [42]	✓	–	✓	Soft prediction vectors	Hierarchical aggregation
ClusterFed [32]	✓	–	✓	Learned representations	Knowledge distillation
FedProto [33]	✓	✓	–	Class prototypes	Prototype aggregation
FedPAC [43]	✓	◦	–	Feature representations	Classifier collaboration
FedCoSR [36]	✓	✓	–	Shareable representations	Contrastive alignment
FedGPD [37]	✓	◦	–	Global prototypes	Prototype distillation
FedTKD [38]	✓	–	–	Distilled knowledge	Adaptive knowledge distillation
Attack-Fed	✓	✓	✓	Attack profiles + prototypes	Attack-aware aggregation + cross-cluster transfer

Note: ✓ denotes explicit consideration of the corresponding property, ◦ denotes indirect or partial consideration, and “–” indicates that the property is not explicitly addressed.

where clients only reveal information to the central server. These findings highlight the importance of explicitly considering the structure of local label spaces when designing federated learning algorithms. However, the study focuses primarily on evaluating and adapting existing federated learning and classifier-combination strategies rather than exploiting relationships between clients label spaces to structure collaboration and transfer knowledge.

Overall, existing approaches demonstrate that label heterogeneity can be mitigated through personalized representations, classifier adaptation, prototype sharing, or knowledge distillation. However, two important limitations remain. First, existing methods generally do not explicitly construct a semantic representation of each client’s local attack expertise and use it as the basis for identifying compatible collaboration groups. Second, once heterogeneous clients are separated or personalized, complementary knowledge learned by different groups may remain isolated, while existing knowledge-transfer mechanisms typically propagate information globally or at the individual-client level without explicitly modeling relationships between heterogeneous groups. These observations motivate our two-level collaboration strategy. In contrast to existing approaches, our framework first characterizes the attack expertise of each client and uses this information to identify clients with compatible attack spaces. It subsequently models the relationships between the resulting groups to enable selective transfer of complementary attack knowledge. Therefore, the proposed approach combines attack-aware client organization with relationship-guided cross-cluster knowledge transfer, aiming to preserve local specialization while recovering complementary knowledge that would otherwise remain isolated.

Table 1 summarizes representative federated learning approaches according to their ability to handle Non-IID data, label-space heterogeneity, client clustering, knowledge sharing, and aggregation mechanisms. Existing methods mainly address statistical heterogeneity through global optimization or client clustering, while more recent approaches explore label-aware personalization and knowledge sharing using prototypes, representations, or distillation. However, these mechanisms generally consider client heterogeneity and knowledge transfer as separate problems. In contrast, our framework jointly considers label-space heterogeneity, attack-aware client clustering, and knowledge sharing to support collaboration among clients with complementary attack knowledge.

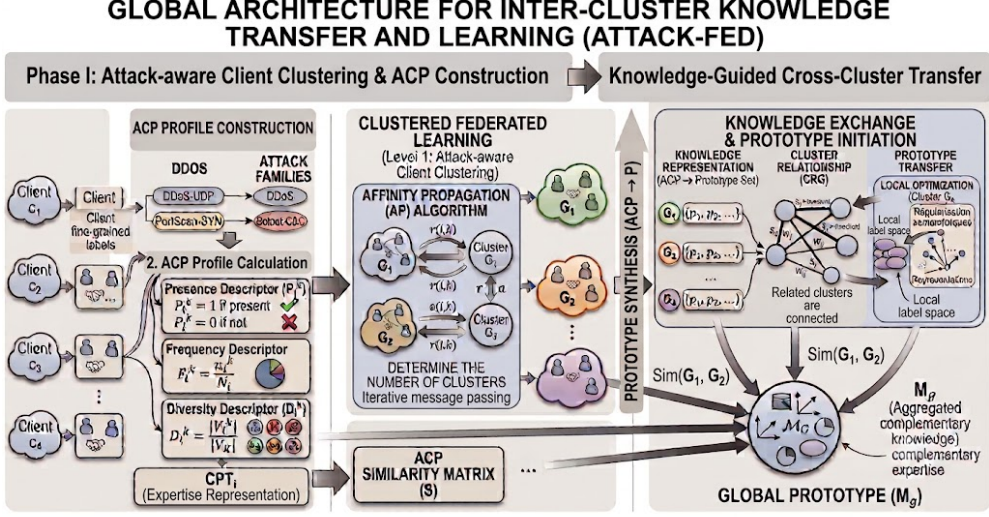


Fig. 1 Attack-Fed architecture.

3 Proposed Method

3.1 Overview of the Proposed Method

In heterogeneous federated learning environments for cybersecurity applications, the clients often observe different subsets of attack categories due to variations in network architectures, deployment scenarios, nodes behaviors, stored data and processes. Although conventional federated learning approaches enable collaborative model training without sharing raw data, they implicitly assume that all clients optimize a common learning objective with a consistent label space. This assumption becomes unrealistic in cybersecurity scenarios, where different IoT nodes may be targeted by attack types. As a result, directly aggregating local models from clients with incompatible label spaces can introduce conflicting updates, degrade knowledge aggregation, and negatively affect the global model performance.

To address this challenge, we propose a two-level method, as illustrated in Fig. 1, that adapts federated learning according to the compatibility and relationships among client tasks. Unlike existing approaches that mainly focus on statistical distribution alignment or feature-level heterogeneity, our method considers the knowledge characteristics of local label spaces and the relationships among attack categories.

The proposed method consists of two complementary levels. In the first level, a label-space-aware client clustering mechanism is introduced to identify groups of clients with compatible attack distributions. By analyzing the local label composition of each participant, clients are organized into groups where knowledge aggregation is performed among semantically related tasks. This step prevents the negative impact caused by forcing heterogeneous label spaces into a single global aggregation process. However, restricting communication only within homogeneous clusters may limit the

exploitation of valuable knowledge shared across different but related attack categories. Therefore, the second level introduces a knowledge-guided cross-cluster transfer mechanism. An attack knowledge structure is constructed to capture relationships between cybersecurity categories, enabling controlled knowledge exchange among different client clusters. Instead of blindly sharing model updates, knowledge transfer is regulated according to the semantic proximity between attack categories, allowing clients to benefit from related experiences while avoiding harmful information propagation. Overall, the proposed approach establishes a hierarchical collaboration paradigm in federated cybersecurity learning, where the first level determines which clients should collaborate, while the second level determines how knowledge can be safely transferred across heterogeneous client groups. This two-level collaboration method aims to improve robustness under heterogeneous label-space conditions while preserving the decentralized nature of federated learning.

The following section presents the first level of the proposed method, which focuses on attack-aware client clustering based on the compatibility of local attack label spaces.

3.2 Level 1: Attack-aware Client Clustering

The first level of the proposed method Fig. 2 aims to regroup clients into groups according to the compatibility of their local label spaces. Unlike conventional clustered federated learning approaches, which generally rely on model updates or feature representations to measure client similarity, our method exploits the characteristics of local attack distributions. The underlying intuition is that clients observing similar attack categories are more likely to learn compatible decision boundaries and can therefore benefit from stronger collaboration during federated optimization. To characterize the attack knowledge available at each client and quantify the compatibility between their local attack spaces, we first construct an Attack Category Profile for each client.

3.2.1 Attack Category Profile (ACP)

In heterogeneous federated intrusion detection systems, each client is exposed to a different subset of attack categories due to the diversity of network environments, deployment scenarios, and local traffic characteristics. Consequently, the local label space of a client only reflects its partial security knowledge and cannot accurately characterize its overall intrusion detection capability. Directly comparing clients based solely on their raw label sets is therefore insufficient, since two clients may exhibit similar expertise despite observing different attack variants. To enable a meaningful comparison between clients while preserving data privacy, we propose an ACP, a compact statistical representation that summarizes the local attack knowledge of each client (similar to Label signature). The proposed ACP is directly derived from the semantic organization of attack labels and does not require any additional representation learning process. For example:

$$\mathcal{C} = \{c_1, c_2, \dots, c_N\} \quad (2)$$

where $\mathcal{C}$ denotes the set of clients, and N represents the total number of participating clients. We further define the set of attack families as follows:

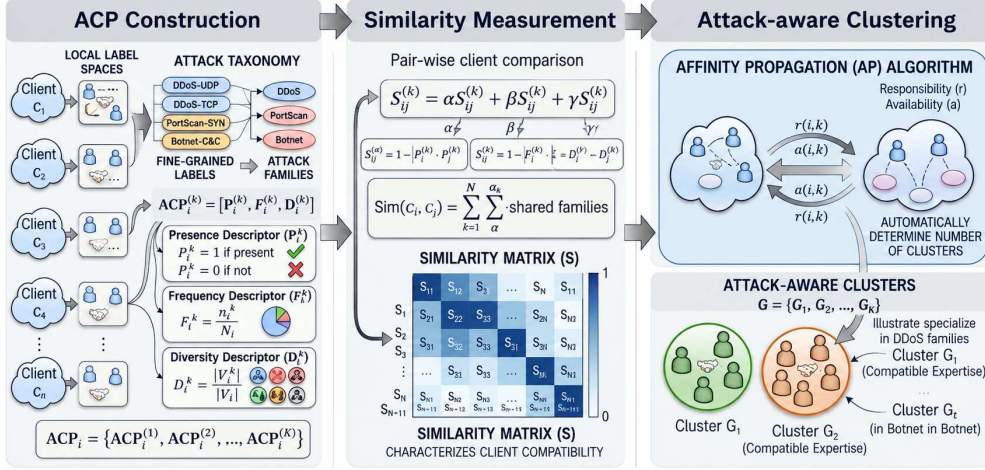


Fig. 2 Level 1: Attack-aware Client Clustering

$$\mathcal{F} = \{F_1, F_2, \dots, F_K\}, \quad (3)$$

where $\mathcal{F}$ denotes the predefined set of attack families, and K represents the total number of attack families. Each attack family contains multiple fine-grained attack labels that share similar malicious behaviors. For example, the DDoS family can be defined as:

$$F_{DDoS} = \{DDoS-UDP, DDoS-TCP, DDoS-SYN, DDoS-ICMP\}. \quad (4)$$

For each client c_i , the original attack labels observed in its local dataset are first mapped into their corresponding attack families according to the predefined attack taxonomy. This transformation allows clients observing different attack variants belonging to the same family to be compared within a unified semantic space. After the label transformation, each client constructs an ACP by computing a complementary statistical descriptors for each attack family. The first descriptor evaluates whether a client observes a specific attack family. For client c_i and attack family F_k , the presence indicator is defined as:

$$P_i^{(k)} = \begin{cases} 1, & \text{if } F_k \in Y_i, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where Y_i denotes the local label space of client c_i . This binary descriptor indicates the attack categories available at each client.

The presence information alone cannot reflect the importance of an attack family in the local dataset. Therefore, we introduce a frequency descriptor that measures the proportion of samples belonging to each attack family. Let $n_i^{(k)}$ denote the number of

samples associated with attack family F_k at client c_i , and let N_i represent the total number of samples available at this client. The frequency descriptor is defined as:

$$F_i^{(k)} = \frac{n_i^{(k)}}{N_i}. \quad (6)$$

This descriptor captures the local distribution of attack categories and reflects the relative expertise of each client.

Clients may observe the same attack family but with different levels of knowledge about its variants. To capture this aspect, we introduce a diversity descriptor. Let V_k denote the complete set of attack variants belonging to family F_k , while $V_i^{(k)}$ represents the subset of variants observed by client c_i . The diversity descriptor is calculated as:

$$D_i^{(k)} = \frac{|V_i^{(k)}|}{|V_k|}. \quad (7)$$

A higher value indicates that the client has observed a larger variety of attack behaviors within the corresponding family.

Finally, the ACP of client c_i regarding attack family F_k is formulated as:

$$ACP_i^{(k)} = \left[P_i^{(k)}, F_i^{(k)}, D_i^{(k)} \right]. \quad (8)$$

The complete attack expertise representation of client c_i is obtained by aggregating the profiles of all attack families:

$$ACP_i = \{ACP_i^{(1)}, ACP_i^{(2)}, \dots, ACP_i^{(K)}\}. \quad (9)$$

The proposed ACP provides a privacy-preserving and a meaningful representation of client expertise. It avoids sharing raw traffic samples and enables the identification of compatible clients according to their observed attack knowledge. This representation will be exploited in the following section to perform attack-aware client clustering.

3.2.2 ACP-based Client Similarity Measurement

To quantify the similarity between clients, each client is represented by an ACP, which summarizes its local attack expertise without revealing private training data. For client c_i and attack family k , the ACP is defined as:

$$ACP_i^{(k)} = \left[P_i^{(k)}, F_i^{(k)}, D_i^{(k)} \right], \quad (10)$$

where $P_i^{(k)}$ indicates the presence of attack family k at client c_i , $F_i^{(k)}$ represents its relative frequency, and $D_i^{(k)}$ denotes the diversity of attacks within this family.

Since the frequency and diversity values are initially standardized using Z-score normalization, an additional Min-Max transformation is applied to map the standardized values to the interval $[0, 1]$. Accordingly, the normalized frequency and diversity values are defined as:

$$\hat{F}_i^{(k)} = \frac{Z_{F_i^{(k)}} - \min(Z_F^{(k)})}{\max(Z_F^{(k)}) - \min(Z_F^{(k)})}, \quad (11)$$

$$\hat{D}_i^{(k)} = \frac{Z_{D_i}^{(k)} - \min(Z_D^{(k)})}{\max(Z_D^{(k)}) - \min(Z_D^{(k)})}. \quad (12)$$

Here, $Z_{F_i}^{(k)}$ and $Z_{D_i}^{(k)}$ denote the Z-score-normalized frequency and diversity values, respectively. The Min-Max transformation ensures that the resulting values lie within $[0, 1]$.

For two clients c_i and c_j , the similarity for attack family k is computed using three complementary components: presence, frequency, and diversity. The overall family-level similarity is defined as:

$$S_{ij}^{(k)} = \alpha S_{P,ij}^{(k)} + \beta S_{F,ij}^{(k)} + \gamma S_{D,ij}^{(k)}, \quad (13)$$

where α , β , and γ are weighting coefficients satisfying:

$$\alpha + \beta + \gamma = 1, \quad \alpha, \beta, \gamma \geq 0. \quad (14)$$

The presence similarity is calculated as:

$$S_{P,ij}^{(k)} = 1 - |P_i^{(k)} - P_j^{(k)}|. \quad (15)$$

Similarly, the frequency and diversity similarities are defined as:

$$S_{F,ij}^{(k)} = 1 - |\hat{F}_i^{(k)} - \hat{F}_j^{(k)}|, \quad (16)$$

$$S_{D,ij}^{(k)} = 1 - |\hat{D}_i^{(k)} - \hat{D}_j^{(k)}|. \quad (17)$$

The absolute differences represent dissimilarity measures, which are converted into similarity scores through the transformation $1 - |\cdot|$. Because the normalized values belong to $[0, 1]$, all three similarity components are bounded within $[0, 1]$.

To focus on attack families shared by both clients, the set of common attack families is defined as:

$$\mathcal{K}_{ij} = \left\{ k \mid P_i^{(k)} = 1 \wedge P_j^{(k)} = 1 \right\}. \quad (18)$$

The overall similarity between clients c_i and c_j is then calculated as the average family-level similarity over their common attack families:

$$\text{Sim}(c_i, c_j) = \begin{cases} \frac{1}{|\mathcal{K}_{ij}|} \sum_{k \in \mathcal{K}_{ij}} S_{ij}^{(k)}, & |\mathcal{K}_{ij}| > 0, \\ 0, & |\mathcal{K}_{ij}| = 0. \end{cases} \quad (19)$$

Finally, the pairwise client similarities are organized into a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$, where N denotes the number of clients:

$$\mathbf{S}_{ij} = \text{Sim}(c_i, c_j), \quad \mathbf{S}_{ii} = 1, \quad 0 \leq \mathbf{S}_{ij} \leq 1. \quad (20)$$

A similarity value of zero is assigned when two clients do not share any attack family. The resulting matrix provides a quantitative representation of the compatibility between clients and serves as the input to the subsequent clustering process. Based on this similarity structure, clients with comparable attack expertise can be grouped together, while clients with substantially different local label spaces remain separated.

3.2.3 Attack-aware Client Clustering

The objective of this step is to discover groups of clients sharing compatible attack expertise according to their local label spaces. Given the ACP-based similarity matrix, where each element S_{ij} represents the similarity between clients c_i and c_j , the clustering problem is formulated as a similarity-based partitioning problem. A high value of S_{ij} indicates that both clients observe comparable attack categories and possess related attack expertise, whereas a low value reflects heterogeneous local label spaces.

However, directly applying conventional clustering algorithms such as K-means [44] is not suitable in this context. Indeed, K-means requires explicit vector representations in an euclidean space and assumes that the number of clusters is predefined [45]. In the considered federated cybersecurity scenario, clients may exhibit highly diverse and partially overlapping attack categories, making the number and structure of client groups unknown in advance. To address these limitations, we adopt the affinity propagation (AP) [46] clustering algorithm to identify attack-aware client communities directly from the similarity matrix. AP is a message-passing-based clustering method that operates on pairwise similarities instead of explicit label vectors. Therefore, it naturally fits the proposed ACP representation, where the relationship between clients is more meaningful than their individual positions in a label space. Another advantage of affinity propagation is that it automatically determines the number of clusters by identifying representative clients, called exemplars, during the clustering process. This property is particularly beneficial in heterogeneous cybersecurity environments, where the number of attack expertise groups cannot be assumed beforehand.

The Affinity Propagation algorithm takes the similarity matrix $\mathbf{S}$ as input and iteratively exchanges two types of messages between clients, responsibility and availability. The responsibility value $r(i, k)$ evaluates the suitability of client k to represent client i :

$$r(i, k) = S(i, k) - \max_{k' \neq k} \{a(i, k') + S(i, k')\}, \quad (21)$$

where $a(i, k)$ denotes the availability message reflecting the evidence that client k can serve as an exemplar for client i .

The availability is updated as:

$$a(i, k) = \min \left(0, r(k, k) + \sum_{i' \notin \{i, k\}} \max(0, r(i', k)) \right). \quad (22)$$

After convergence, each client is assigned to the exemplar maximizing the combined responsibility and availability values:

$$e_i = \arg \max_k (a(i, k) + r(i, k)). \quad (23)$$

Clients associated with the same exemplar are grouped into the same attack-aware cluster:

$$\mathcal{G} = \{G_1, G_2, \dots, G_K\}. \quad (24)$$

Each cluster G_k therefore represents a community of clients with compatible attack knowledge. Instead of forcing all clients to participate in a single global aggregation process, federated optimization can subsequently be performed within each discovered cluster, reducing the negative impact of label-space heterogeneity.

The overall attack-aware client clustering procedure is summarized in algorithm 1. Starting from the ACP-based similarity matrix, Affinity Propagation iteratively updates the responsibility and availability messages to identify representative clients (exemplars). Each client is then assigned to the exemplar with the highest combined responsibility and availability score. Clients sharing the same exemplar are finally grouped into attack-aware clusters, with the number of clusters determined automatically by the clustering process.

Algorithm 1 Attack-aware Client Clustering using Affinity Propagation

```

1: Input: Similarity matrix  $\mathbf{S} \in \mathbb{R}^{N \times N}$ , damping factor  $\lambda$ , maximum number of
   iterations  $T$ 
2: Output: Attack-aware client clusters  $\mathcal{G} = \{G_1, G_2, \dots, G_K\}$ 
3: Initialize responsibility matrix  $\mathbf{R} \leftarrow \mathbf{0}$ 
4: Initialize availability matrix  $\mathbf{A} \leftarrow \mathbf{0}$ 
5: for  $t = 1$  to  $T$  do
6:   for  $i = 1$  to  $N$  do
7:     for  $k = 1$  to  $N$  do
8:        $r(i, k) \leftarrow S(i, k) - \max_{k' \neq k} \{a(i, k') + S(i, k')\}$ 
9:     end for
10:   end for
11:   for  $i = 1$  to  $N$  do
12:     for  $k = 1$  to  $N$  do
13:        $a(i, k) \leftarrow \min \left( 0, r(k, k) + \sum_{i' \notin \{i, k\}} \max(0, r(i', k)) \right)$ 
14:     end for
15:   end for
16:   Apply damping factor  $\lambda$  to  $\mathbf{R}$  and  $\mathbf{A}$ 
17:   if convergence criterion is satisfied then
18:     break
19:   end if
20: end for
21: for  $i = 1$  to  $N$  do
22:    $e_i \leftarrow \arg \max_k \{a(i, k) + r(i, k)\}$ 
23: end for
24: Group clients having the same exemplar:
25:  $\mathcal{G} \leftarrow \{G_k \mid G_k = \{c_i : e_i = k\}\}$ 
26: Remove empty groups from  $\mathcal{G}$ 
27: Return:  $\mathcal{G}$ 

```

Although attack-aware clustering improves collaboration by restricting aggregation to clients with compatible label spaces, it may isolate complementary knowledge distributed across different clusters. Therefore, a controlled knowledge transfer mechanism is required to exploit meaningful relationships between attack communities while preserving the benefits of cluster.

3.3 Level 2: Knowledge-Guided Cross-Cluster Transfer

The first learning level addresses label-space heterogeneity by grouping clients with compatible attack distributions into attack-aware clusters. Within each cluster, clients can therefore collaborate without directly aggregating models trained on incompatible label spaces. However, this specialization creates a second challenge, different clusters may acquire complementary knowledge about different attack categories, while their learned models remain isolated from one another. The objective of the second learning level is therefore to enable selective knowledge exchange between the clusters identified in the first level. Importantly, the goal is not to merge heterogeneous cluster models or to force all clusters to learn the same label space. Instead, Attack-Fed represents the knowledge learned by each cluster through attack prototypes, identifies meaningful relationships between clusters, and transfers only relevant knowledge along these relationships.

The proposed mechanism consists of three successive steps. First, each attack-aware cluster summarizes its learned attack knowledge using a set of prototypes. Second, these prototype representations are used to construct a Cluster Relationship Graph (CRG), which identifies clusters with related learned representations. Third, prototypes are selectively propagated between connected clusters and incorporated into local optimization through a prototype-based regularization term. This design establishes a clear separation between the two learning levels, the first level determines which clients should collaborate, whereas the second level determines which knowledge can be transferred across the resulting clusters. In this way, Attack-Fed preserves cluster-specific specialization while exploiting complementary attack knowledge distributed across the federated environment.

3.3.1 Cluster Knowledge Representation

Following the attack-aware client clustering performed in the first learning level, each cluster G_k conducts an independent federated optimization process using the clients assigned to it. As a result, each cluster gradually learns discriminative representations of the attack categories available in its local label space. To enable knowledge exchange across clusters without sharing model parameters or raw network traffic, Attack-Fed summarizes the knowledge learned by each cluster using attack prototypes. A prototype represents the characteristic latent representation learned for a particular attack category. Thus, instead of using the complete cluster model as the unit of knowledge exchange, Attack-Fed uses compact representations of the learned attack knowledge.

More formally, let $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ denote the latent representation of sample $\mathbf{x}_i$. For an attack category a observed within cluster G_k , its prototype is computed as the centroid of the latent representations associated with that category:

$$\mathbf{p}_a^{(k)} = \frac{1}{N_a^{(k)}} \sum_{i: y_i=a} \mathbf{z}_i. \quad (25)$$

where $N_a^{(k)}$ denotes the number of samples belonging to attack category a within cluster G_k .

The locally learned prototype set of cluster G_k can consequently be represented as:

$$\mathcal{P}_k^{\text{local}} = \left\{ \mathbf{p}_a^{(k)} \mid a \in \mathcal{Y}_k \right\}. \quad (26)$$

where $\mathcal{Y}_k$ denotes the set of attack categories locally represented within cluster G_k . This set is maintained locally and is not required to be disclosed as a whole to other clusters. In particular, Attack-Fed does not transmit the size of $\mathcal{Y}_k$ or the complete local prototype set as an explicit description of the cluster's label space.

Instead, the subsequent cross-cluster knowledge-transfer stage evaluates the relevance of individual prototypes with respect to the target cluster. Only prototypes selected as relevant for transfer are exposed to the corresponding target cluster. We denote the resulting transferable prototype set by:

$$\mathcal{P} * k^{\text{transfer}} \subseteq \mathcal{P} * k^{\text{local}}. \quad (27)$$

This distinction allows the prototype representation to serve as a compact knowledge interface between clusters while limiting the disclosure of local label-space information. The transmitted knowledge consists of selected latent prototypes rather than raw network traffic or complete model parameters. Moreover, the communication cost is determined by the number of prototypes selected for transfer rather than by the size. Finally, the resulting prototype representations provide a common latent space for evaluating the relevance of cross-cluster knowledge before it is transferred, as described in the following section.

3.3.2 Cluster Relationship Graph

Once each cluster has been represented by its prototype set, the next step is to determine which clusters can exchange knowledge. Direct communication between every pair of clusters is undesirable because unrelated knowledge may lead to negative transfer. Attack-Fed therefore constructs a cluster relationship graph to restrict knowledge exchange to meaningful relationships. Unlike the first learning level, where client relationships are determined from their local attack-category distributions, the CRG is constructed from the learned prototype representations. It therefore reflects relationships between the attack knowledge learned by clusters rather than simply their label distributions. The Cluster Relationship Graph is defined as:

$$G_C = (V_C, E_C, W), \quad (28)$$

where $V_C = \{G_1, G_2, \dots, G_K\}$ represents the set of attack-aware clusters, E_C denotes the set of potential knowledge-exchange links, and W is the weighted adjacency matrix describing the strength of the relationships between clusters.

Each cluster G_k is represented by its prototype set:

$$\mathcal{P}_k = \{\mathbf{p}_1^{(k)}, \mathbf{p}_2^{(k)}, \dots, \mathbf{p}_{m_k}^{(k)}\}. \quad (29)$$

The relationship between two clusters is estimated from the similarity of their learned prototype representations. For two clusters G_i and G_j , the pairwise relationship score is computed as the average cosine similarity between their prototypes:

$$S_{ij} = \frac{1}{|\mathcal{P}_i||\mathcal{P}_j|} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{\mathbf{q} \in \mathcal{P}_j} \frac{\mathbf{p}^T \mathbf{q}}{\|\mathbf{p}\| \|\mathbf{q}\|}, \quad (30)$$

where $\mathbf{p}$ and $\mathbf{q}$ denote prototypes belonging to clusters G_i and G_j , respectively. Cosine similarity measures the alignment between the learned latent representations while being independent of their magnitudes. The resulting similarity values form the cluster relationship matrix:

$$\mathbf{S} = [S_{ij}]_{K \times K}, \quad (31)$$

where S_{ij} indicates the degree of representational compatibility between clusters G_i and G_j .

To avoid unnecessary knowledge exchange, only sufficiently important relationships are retained. Specifically, an edge is established between two clusters when their relationship score exceeds a predefined threshold δ :

$$E_{ij} = \begin{cases} 1, & S_{ij} \geq \delta, \\ 0, & S_{ij} < \delta. \end{cases} \quad (32)$$

Here, E_{ij} acts as a binary transfer mask that determines whether knowledge exchange between clusters G_i and G_j is permitted. When $E_{ij} = 1$, the relationship is considered sufficiently relevant and the corresponding prototypes can be considered for cross-cluster knowledge transfer. Conversely, when $E_{ij} = 0$, the relationship is discarded and no prototype transfer is performed between the two clusters. Therefore, E_{ij} restricts knowledge transfer to sufficiently relevant cross-cluster relationships and prevents unnecessary or potentially irrelevant knowledge exchange.

3.3.3 Prototype-Guided Cross-Cluster Transfer

The CRG identifies the cluster pairs that are allowed to participate in knowledge exchange. The next step is to use these relationships to transfer useful knowledge without modifying the cluster-specific label spaces. For a given cluster G_k , its neighboring clusters are defined as:

$$\mathcal{N}_k = \{G_j \mid W_{kj} > 0\}. \quad (33)$$

The external knowledge available to cluster G_k is then obtained from the prototype sets of its neighboring clusters:

$$\mathcal{P}_{ext}^{(k)} = \bigcup_{G_j \in \mathcal{N}_k} \mathcal{P}_j. \quad (34)$$

Importantly, these external prototypes are not introduced as new local classes. The local classification task remains unchanged, and cluster G_k continues to optimize its

classifier using only its own label space. The local classification objective is therefore defined as:

$$\mathcal{L}_{cls} = - \sum_{i=1}^{N_k} \sum_{c \in L_k} y_{i,c} \log(\hat{y}_{i,c}), \quad (35)$$

where L_k denotes the local label space of cluster G_k , $y_{i,c}$ is the ground-truth label, and $\hat{y}_{i,c}$ is the predicted probability for class c . The transferred prototypes instead provide additional information at the representation level. Given a local sample embedding $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$, the semantic discrepancy between this representation and an external prototype $\mathbf{p}$ is measured using cosine distance:

$$d(\mathbf{z}_i, \mathbf{p}) = 1 - \frac{\mathbf{z}_i^T \mathbf{p}}{\|\mathbf{z}_i\| \|\mathbf{p}\|}. \quad (36)$$

For each local representation, the most relevant prototype is identified according to the minimum cosine distance. The resulting prototype-guided regularization term is formulated as:

$$\mathcal{L}_{proto} = \frac{1}{N_k} \sum_{i=1}^{N_k} \min_{\mathbf{p} \in \mathcal{P}_{ext}^{(k)}} d(\mathbf{z}_i, \mathbf{p}), \quad (37)$$

where the minimum operator selects the external prototype that is most closely aligned with the current local representation. The final optimization objective for cluster G_k is therefore:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{proto}, \quad (38)$$

where λ controls the contribution of the transferred knowledge.

This formulation preserves the original classification objective while introducing cross-cluster information at the representation level. Therefore, the transferred prototypes do not require the receiving cluster to introduce new attack labels or modify its classifier. Instead, they provide references that can guide the learned representation toward attack knowledge already captured by related clusters. Through successive communication rounds, each cluster can consequently benefit from knowledge distributed across its neighboring clusters while maintaining its local specialization. The resulting mechanism provides a controlled alternative to direct model aggregation across heterogeneous label spaces. Finally, the accumulation of knowledge across the different clusters provides a progressively broader representation of the attack space at the federated level. Let $\mathcal{P}_k$ denote the prototype set produced by cluster G_k . The global prototype repository can be represented as:

$$\mathcal{M}_G = \bigcup_{k=1}^K \mathcal{P}_k, \quad (39)$$

where K is the number of attack-aware clusters.

The repository provides a compact representation of the attack knowledge distributed across the federated environment. Unlike a conventional global model

obtained by directly averaging heterogeneous local models, $\mathcal{M}_G$ preserves the complementary expertise learned by the different clusters. Consequently, global knowledge completion does not mean forcing every cluster to learn all attack categories. Rather, it denotes the progressive integration of complementary attack representations while each cluster continues to optimize according to its own local label space. Through this process, Attack-Fed combines local specialization with cross-cluster knowledge sharing: attack-aware clustering limits incompatible collaboration, whereas prototype-guided transfer allows relevant knowledge from different clusters to contribute to a broader global understanding of the attack space.

4 EXPERIMENTS

This section presents experiments conducted on different datasets to detect intrusions in a federated setting. The effectiveness of the proposed strategy Attack-Fed is evaluated by comparing it with various baseline methods.

4.1 Datasets

We conduct our experiments on two widely used and publicly available benchmark datasets for network intrusion detection, CIC-IoT-2023 [47] and UNSW-NB15 datasets [48].

CIC-IoT-2023 is a large-scale benchmark for IoT intrusion detection. This dataset contains diverse network traffic collected from realistic IoT environments and includes both benign and malicious activities [49]. Each sample is described by 47 network flow-based features, capturing statistical and protocol-level characteristics such as flow duration, packet statistics, and TCP/IP. For the purpose of this study, we select a subset of representative label that are commonly observed across IoT devices, ensuring a consistent label space across all clients.

The UNSW-NB15 dataset consists of raw network packets that were generated by a tool called IXIA perfectStorm in the cyber range Lab of the Australian centre for cyber Security (ACCS) [50]. It contains a hybrid of real modern normal activities and synthetic contemporary attack behaviours. The dataset has nine types of attacks, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The Argus and Bro-IDS tools were used, and twelve algorithms were developed to generate 49 features along with the class label.

4.2 Partition data

We simulate label-space heterogeneity using a partially class-disjoint data (PCDD) strategy, where each client observes only a subset of the global label space.

4.2.1 PCDD definition

To make this clear, we first define the notations for the partially class-disjoint data distribution in federated learning. Let $\mathcal{C}$ denote the set of all classes in the global label space, and let $\mathcal{P}$ denote the set of all local clients. Considering real-world constraints such as privacy and environmental limitations, each local client only owns samples

from a subset of the global classes. Formally, for the k -th client $\mathcal{P}_k$, its corresponding local dataset $\mathcal{D}_k$ can be expressed as [51]:

$$\mathcal{D}_k = \{(x_{k,i}, y_{k,i}) \mid y_{k,i} = c, c \in \mathcal{C}_k\}, \quad \mathcal{C}_k \subsetneq \mathcal{C}. \quad (40)$$

The number of samples belonging to class c , where $c \in \mathcal{C}_k$, at client $\mathcal{P}_k$ is denoted by N_k^c . In typical label distribution skew scenarios, the number of samples per class, N_k^c , varies across clients, while the number of classes $|\mathcal{C}_k|$ generally remains the same or similar across clients. In contrast, under the partially class-disjoint distribution (PCDD) scenario, the local label spaces $\mathcal{C}_k$ can differ substantially across clients.

4.3 Experimental Settings

4.3.1 Evaluation metrics

To evaluate the performance of the proposed method under label heterogeneity, we adopt several standard metrics commonly used in intrusion detection systems. These metrics provide complementary insights into the effectiveness and robustness of the models accuracy [52], recall [53], F1-score [54] and precision [47].

4.3.2 Baselines

To evaluate the effectiveness of our approach under label heterogeneity, we compare it against several widely used and readily available federated learning baselines that represent the main categories of methods in the literature, FedAvg [55], FedProx [56], FedProto [33], and IFCA [28].

4.3.3 Experimental settings

We implement Attack-Fed and all baseline methods using Python 3.9.10. The experiments are conducted on a machine equipped with an Intel[®] Xeon[®] Silver 4210 CPU running at 2.20 GHz, with 10 physical cores and 20 logical processors, and an NVIDIA GPU. The federated learning process is simulated using 10 and 20 clients for all datasets, with all clients participating in every communication round. All methods are trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. The federated training process is performed for 100 communication rounds. For the representation learning component of Attack-Fed, the latent representation dimension is set to 32. The prototype-guided knowledge transfer is controlled by the regularization coefficient λ , while the connectivity of the Cross-cluster Relationship Graph is determined by the similarity threshold δ . The same training configuration is applied to all methods whenever applicable to ensure a fair comparison.

5 Main Results and Discussion

This section provides a comprehensive evaluation of Attack-Fed ¹ from several complementary perspectives. We first examine whether the proposed attack-aware client

¹Attack-Fed

Table 2 Performance comparison of different federated learning methods on CIC-IoT-2023.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
FedAvg	84.21	82.76	79.84	80.93
FedProx	85.07	83.54	81.12	82.18
CFL	86.34	84.91	82.76	83.68
IFCA	87.02	85.47	83.91	84.68
FedProto	87.61	86.24	84.72	85.41
Attack-Fed	90.18	89.37	87.94	88.56

clustering can effectively organize clients according to their local attack expertise and improve federated learning under label-space heterogeneity. We then investigate the contribution of the selective cross-cluster knowledge-transfer mechanism and assess whether transferring relevant attack knowledge can further improve the detection performance. The robustness of Attack-Fed is subsequently evaluated under different levels of label-space heterogeneity and different numbers of participating clients. To provide a more detailed understanding of the learned knowledge, we further analyze the performance across individual attack categories. Finally, we study the convergence behavior and execution time of the considered methods to assess the training stability and practical efficiency of the proposed method.

5.0.1 Effectiveness of Attack-aware Client Clustering

The objective of this experiment is to evaluate whether explicitly considering the attack expertise of participating clients can provide a more effective collaboration structure under label-space heterogeneity. In contrast to conventional federated learning, where all clients contribute to a common global model, the proposed first-level mechanism identifies groups of clients with compatible attack spaces before performing model aggregation. We assess the proposed label-aware client clustering strategy by comparing it with conventional federated aggregation and representative clustered federated learning approaches. This comparison investigates whether incorporating clients attack-category profiles into the collaboration structure provides an advantage over treating all clients equally or relying solely on generic similarity-based clustering.

The comparison includes FedAvg and FedProx as global aggregation baselines, as well as representative clustered federated learning approaches, namely CFL and IFCA. We additionally consider FedProto as a class-aware federated baseline, since it enables knowledge sharing at the class level rather than relying exclusively on direct model-parameter aggregation. Table 2 presents the results obtained on the CIC-IoT-2023 dataset with 20 clients.

The results indicate that explicitly exploiting attack-category information when constructing client groups provides a clear advantage over conventional federated aggregation. FedAvg obtains a F1 score of 80.93%, while FedProx slightly improves this result to 82.18%. This limited improvement suggests that regularizing the local optimization process alone is insufficient to address the incompatibility between clients with different local classification spaces. Clustered federated learning methods achieve better performance than global aggregation. CFL and IFCA reach F1 scores of 83.68%

Table 3 Performance comparison under different numbers of clients on UNSW-NB15.

Method	10 Clients		20 Clients	
	Recall (%)	F1-score (%)	Recall (%)	F1-score (%)
FedAvg	84.76	85.61	79.84	80.93
FedProx	85.63	86.42	81.12	82.18
CFL	87.14	87.96	82.76	83.68
IFCA	87.82	88.53	83.91	84.68
FedProto	88.46	89.21	84.72	85.41
Attack-Fed	91.37	92.14	87.94	88.56

and 84.68%, respectively. This improvement confirms the benefit of recognizing that the federation may contain groups of clients with different learning characteristics. However, these methods do not explicitly use the set of attack categories observed by each client as the basis for defining collaboration groups. FedProto further improves the performance to 85.41% F1, demonstrating the potential benefit of class-level knowledge sharing. Nevertheless, the proposed ACP-based clustering achieves the highest performance, reaching 88.56% F1, 87.94% recall, and 90.18% accuracy. Compared with the strongest baseline, FedProto, the proposed clustering strategy provides an absolute improvement of 3.15% points in F1 and 3.22% points in recall. The improvement in recall is particularly relevant for the considered cybersecurity application. Unlike accuracy, recall gives equal importance to individual attack categories and therefore provides a more informative measure when the distribution of attack categories differs across clients. The higher recall obtained by the proposed approach indicates that label-aware client grouping helps preserve knowledge associated with attack categories that are not uniformly observed across the federation. The results also reveal an important distinction between generic client clustering and label-aware clustering. CFL and IFCA demonstrate that clustering clients can improve federated learning under heterogeneous conditions. However, their performance remains below that of the proposed approach. This difference suggests that identifying similar clients solely from model behavior or learned representations does not necessarily capture the most relevant source of heterogeneity in the considered setting. In contrast, the proposed strategy directly exploits the attack-category composition of each client when establishing collaboration groups.

To further investigate this effect, we analyze the performance across different numbers of clients. Table 3 reports the results for 10 and 20 clients.

The performance gap becomes more pronounced when the number of clients increases from 10 to 20. With more participating clients, the diversity of local attack-category profiles also increases, making indiscriminate global aggregation less suitable. FedAvg experiences a F1 reduction from 85.61% to 80.93%, whereas the proposed method decreases from 92.14% to 88.56%. The smaller degradation observed with ACP-based clustering indicates that the proposed collaboration structure is more robust to an increasing number of heterogeneous participants. Overall, these results demonstrate that the improvement does not simply originate from using fewer clients in each aggregation round. Instead, the criterion used to establish collaboration groups

Table 4 Effect of knowledge-guided cross-cluster transfer on CIC-IoT-2023.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
FedAvg	84.21	82.76	79.84	80.93
FedProx	85.07	83.54	81.12	82.18
CFL	86.34	84.91	82.76	83.68
IFCA	87.02	85.47	83.91	84.68
FedProto	87.61	86.24	84.72	85.41
ACP Clustering	90.18	89.37	87.94	88.56
Attack-Fed	92.47	91.83	90.76	91.28

plays a central role. By exploiting the attack-category composition of each client, ACP-based clustering creates more semantically compatible groups and reduces the interference caused by aggregating knowledge originating from substantially different classification spaces. However, the results also reveal a limitation of clustering-based collaboration. Although clients within the same cluster can exchange highly compatible knowledge, potentially useful information may remain isolated between different clusters. For example, two clusters may observe different but semantically related attack categories. Completely independent cluster-wise collaboration would prevent the federation from exploiting these complementary relationships. This limitation motivates the second component of the proposed framework, namely knowledge-guided cross-cluster transfer, whose effectiveness is evaluated in the following subsection.

5.0.2 Effectiveness of Knowledge-guided Cross-cluster Transfer

Following the evaluation of the ACP-based clustering strategy, we next investigate whether knowledge-guided transfer can further improve collaboration across heterogeneous client clusters. The motivation behind this experiment is that label-aware clustering improves the compatibility of clients within the same cluster, but may also restrict the direct exchange of knowledge between different clusters. Consequently, potentially useful information associated with complementary attack categories may remain isolated within individual clusters. To evaluate the contribution of cross-cluster knowledge transfer, we compare the proposed complete method with the main federated learning baselines, the ACP-based clustering strategy without cross-cluster transfer, and the complete ACP-based approach equipped with knowledge-guided transfer. All methods are evaluated under the same experimental configuration, including the same dataset partitioning, optimization parameters, number of clients, and communication rounds. Table 4 reports the obtained results on CIC-IoT-2023 with 20 clients.

The results show a consistent improvement when knowledge-guided cross-cluster transfer is incorporated into the ACP-based clustering framework. The ACP-based clustering strategy achieves a F1 score of 88.56%, whereas the complete approach reaches 91.28%. This corresponds to an absolute improvement of 2.72% points and demonstrates that the benefit of the proposed framework does not originate solely from label-aware client clustering. The same tendency can be observed for recall, which increases from 87.94% with ACP-based clustering to 90.76% with knowledge-guided transfer. This improvement is particularly important in the considered cybersecurity

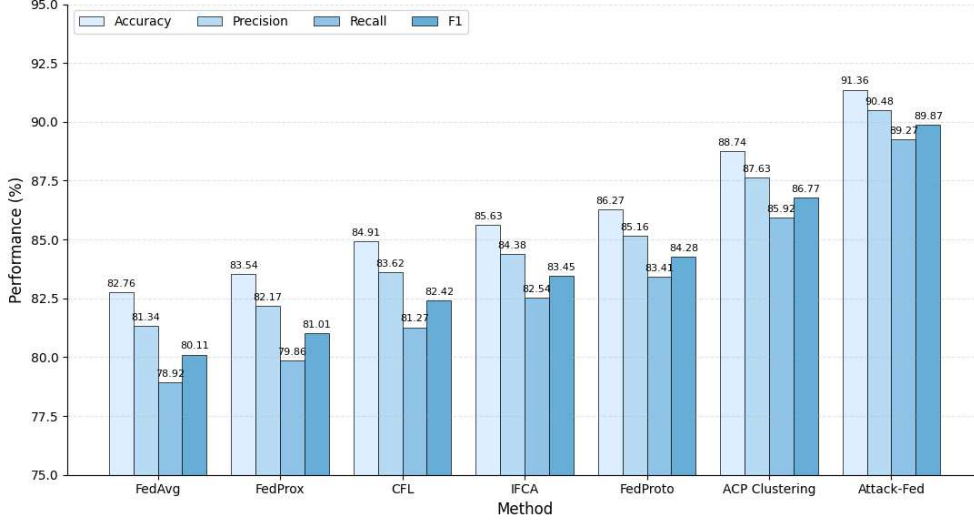


Fig. 3 Effect of knowledge-guided cross-cluster transfer on UNSW-NB15 with 20 clients.

scenario because recall gives equal importance to the different attack categories. Therefore, the additional gain suggests that cross-cluster knowledge transfer helps recover information associated with attack categories that are not uniformly available across the federation. Compared with FedAvg, the complete approach improves F1 from 80.93% to 91.28%, corresponding to an absolute gain of 10.35% points. More importantly, the comparison between ACP-based clustering and the full approach isolates the contribution of the second level of the proposed method. The additional 2.72% point improvement confirms that knowledge-guided transfer provides complementary benefits beyond the clustering mechanism.

Fig 3 provides the results on the UNSW-NB15 dataset with 20 clients, it shows a consistent performance improvement across the considered approaches. FedAvg achieves the lowest performance, while FedProx, CFL, IFCA, and FedProto progressively improve the results. The ACP Clustering strategy further enhances the performance, confirming the benefit of grouping clients with similar attack characteristics. Finally, Attack-Fed achieves the best results across all metrics, reaching 91.36% accuracy, 90.48% precision, 89.27% recall, and 89.87% F1-score. These results indicate that the proposed attack-aware knowledge transfer mechanism can better exploit the heterogeneity among clients and improve both detection accuracy and performance.

5.0.3 Comparison of Cross-Cluster Transfer Strategies

To determine whether the improvement is specifically attributable to the knowledge-guided mechanism, rather than simply to the introduction of additional cross-cluster communication, we compare three transfer strategies: (i) no cross-cluster transfer, (ii) random cross-cluster transfer, and (iii) knowledge-guided cross-cluster transfer.

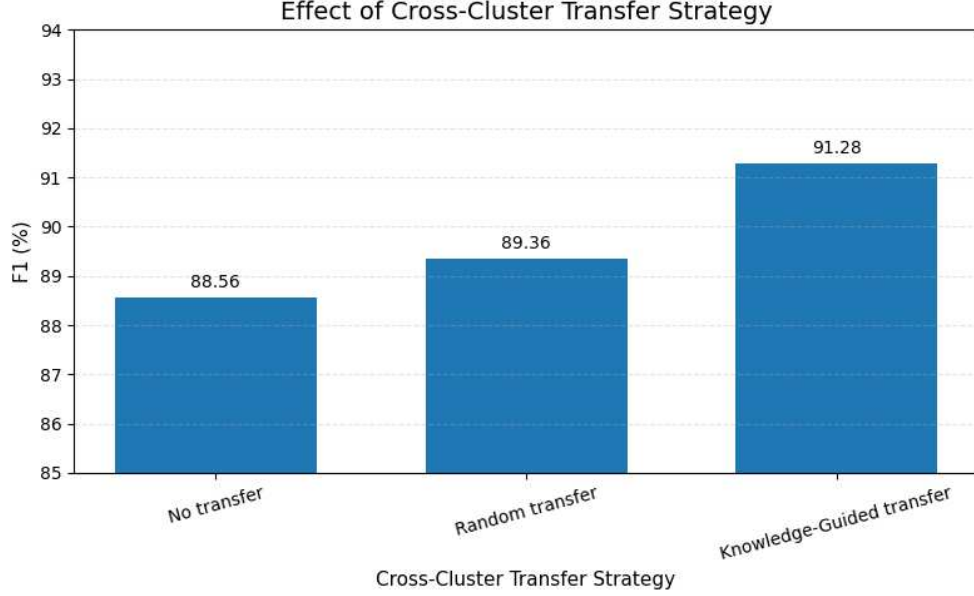


Fig. 4 F1-score comparison of different cross-cluster knowledge transfer strategies on the CIC-IoT-2023 dataset.

The comparison in Fig. 4 provides an important additional observation. Random cross-cluster transfer improves the F1 from 88.56% to 89.36%, indicating that additional information exchange can be beneficial to some extent. However, this improvement remains substantially smaller than that obtained with knowledge-guided transfer. The proposed knowledge-guided mechanism achieves the highest F1 of 91.28%. This result supports the central hypothesis that cross-cluster knowledge should be transferred selectively according to the semantic relationships between attack categories rather than indiscriminately across all clusters.

As illustrated in Fig 5, the results show a clear improvement as the cross-cluster transfer strategy becomes more informed. The F1 increases from 87.16% with no transfer to 88.22% with random transfer, while knowledge-guided transfer achieves the highest performance of 89.43%. This confirms that the improvement is not solely due to additional performance cross-cluster communication, but mainly results from selecting and transferring relevant knowledge between compatible clusters.

These results demonstrate the overall effectiveness of the proposed cross-cluster knowledge transfer mechanism. However, aggregate performance metrics may mask differences in the ability of the model to recognize individual attack categories. Therefore, we further examine the per-attack-category performance to assess whether the transferred knowledge benefits different types of attacks consistently.

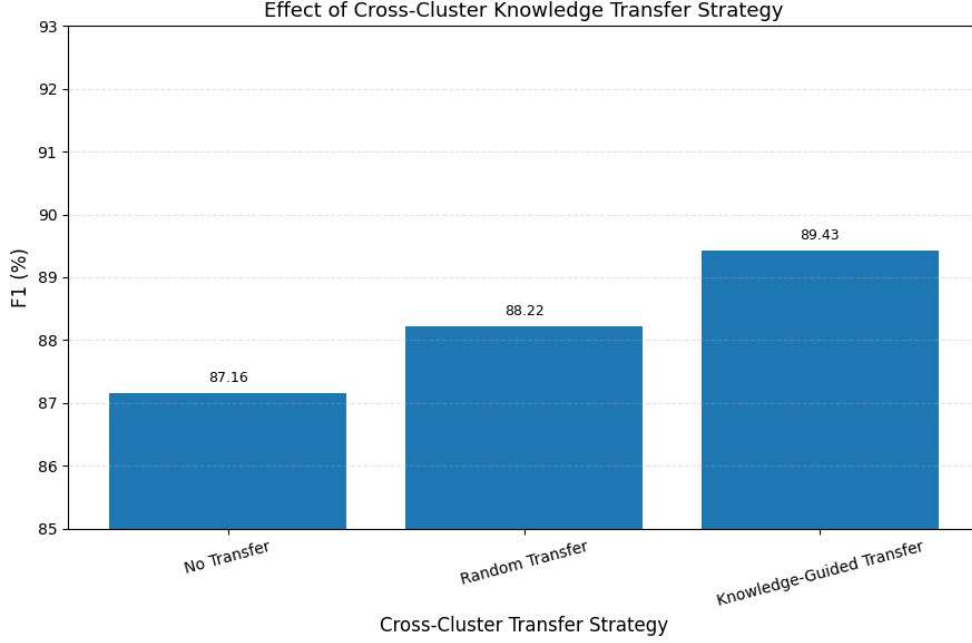


Fig. 5 F1-score comparison of different cross-cluster knowledge transfer strategies on the UNSW-NB15 dataset.

Table 5 Per-Attack-Category F1-Score Comparison on CIC-IoT-2023

Attack Category	IFCA	FedProto	ACP + KG
DDoS	84.12	89.37	92.61
DoS	81.46	87.92	90.84
Recon	76.35	84.18	88.93
Botnet	73.21	82.47	87.65
Spoofing	70.84	79.63	84.91

5.0.4 Per-Attack-Category Analysis

To provide a more detailed assessment of the proposed approach, we analyze the performance at the individual attack-category level. Table 5 reports the F1-score obtained for representative attack categories.

The per-category results reveal that the benefit of knowledge-guided transfer is particularly pronounced for attack categories that are not uniformly represented across clients. For instance, the F1-score for the Botnet category increases from 82.47% with ACP-based clustering to 87.65% after introducing knowledge-guided transfer. A similar improvement is observed for Spoofing, whose F1-score increases from 79.63% to 84.91%. These results provide evidence that the proposed cross-cluster mechanism is particularly useful when knowledge about a specific attack category is concentrated in

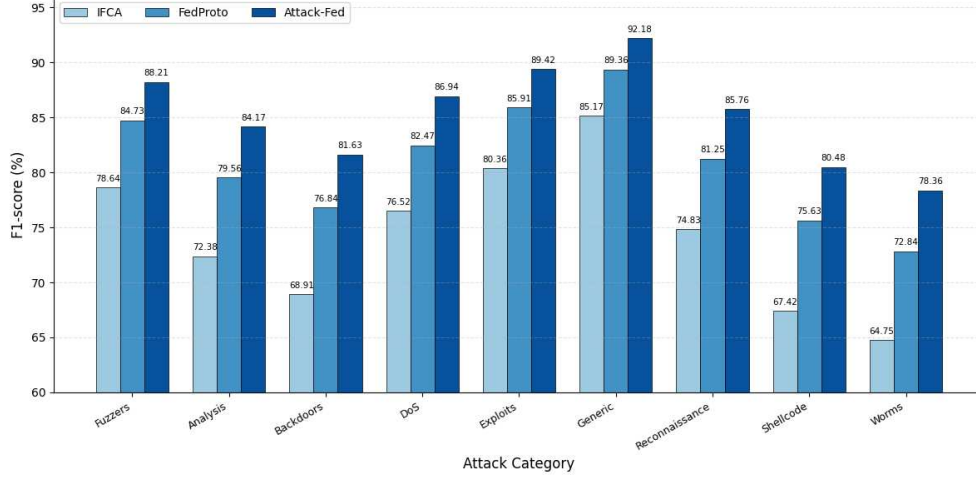


Fig. 6 Per-Attack-Category F1-Score Comparison on UNSW-NB15.

a limited subset of clients. Instead of requiring every client to observe the same category locally, the framework allows relevant knowledge to propagate from clients or clusters possessing that knowledge towards other compatible parts of the federation.

As illustrated in Fig 6, Attack-Fed consistently achieves the highest F1-score across all attack categories on UNSW-NB15, outperforming both IFCA and FedProto. The improvement is particularly noticeable for challenging categories such as Worms, Shellcode, and Backdoors. These results demonstrate that knowledge-guided cross-cluster transfer improves the detection of diverse attack types and provides more effective knowledge sharing under label heterogeneity.

5.0.5 Discussion

The experimental results provide three main observations. First, ACP-based clustering substantially improves federated learning compared with global aggregation, confirming that label-aware client grouping is beneficial when clients observe different attack categories. Second, the additional knowledge-guided transfer consistently improves the performance of ACP-based clustering, demonstrating that intra-cluster collaboration alone does not fully exploit the knowledge available in the federation. Third, the comparison with random transfer shows that the improvement cannot be attributed solely to additional communication. These findings support the complementary roles of the two components of the proposed framework. ACP-based clustering establishes a compatible collaboration structure by grouping clients with related attack-category spaces. Knowledge-guided transfer then selectively bridges these clusters when complementary attack knowledge can be beneficial. Consequently, the framework avoids both extremes, indiscriminate global aggregation, which may introduce conflicting updates, and completely isolated clusters, which may prevent useful knowledge sharing.

Table 6 Ablation study on CIC-IoT-2023 and UNSW-NB15 with 20 clients.

Dataset	Configuration	Accuracy	Precision	Recall	F1-score
CIC-IoT-2023	Attack-Fed w/o Clustering	86.42	85.17	83.61	84.73
	Attack-Fed w/o Cross-Cluster Transfer	90.18	89.37	87.94	88.56
	Attack-Fed	92.47	91.83	90.76	91.28
UNSW-NB15	Attack-Fed w/o Clustering	82.47	81.36	79.84	80.42
	Attack-Fed w/o Cross-Cluster Transfer	86.31	85.42	83.76	84.51
	Attack-Fed	89.24	88.37	86.92	87.58

Overall, the results demonstrate that the proposed approach provides a more effective balance between knowledge isolation and knowledge sharing. The former limits the negative impact of incompatible local objectives, while the latter enables complementary knowledge to propagate across clusters when supported by relationships between attack categories.

5.0.6 Ablation Study

To further investigate the contribution of the two main components of Attack-Fed, we conduct an ablation study by selectively removing the label-aware clustering mechanism and the cross-cluster knowledge transfer mechanism. Three configurations are considered: Attack-Fed w/o Clustering, Attack-Fed w/o Cross-Cluster Transfer, and the complete Attack-Fed framework. All configurations are evaluated under identical experimental conditions on CIC-IoT-2023 and UNSW-NB15 with 20 clients.

The results in Table 6 show that both components contribute to the performance of Attack-Fed on the two datasets. Removing the clustering mechanism produces the largest degradation, while disabling cross-cluster transfer results in a smaller but consistent decrease in performance. The complete framework achieves the highest F1-score on both datasets, reaching 91.28% on CIC-IoT-2023 and 87.58% on UNSW-NB15. These results confirm that attack-aware clustering and selective cross-cluster knowledge transfer provide complementary contributions to handling label-space heterogeneity. We further evaluate the considered methods under three levels of label-space heterogeneity: low, medium, and high. These settings progressively increase the differences between the local attack-category spaces of participating clients, while the remaining experimental parameters are kept unchanged.

As shown in Table 7, all methods experience a performance degradation as label-space heterogeneity increases. However, Attack-Fed consistently achieves the highest F1-score across all three heterogeneity levels on both datasets. The performance gap is particularly visible under high heterogeneity, where Attack-Fed reaches 89.76% on CIC-IoT-2023 and 86.71% on UNSW-NB15. These results indicate that the proposed combination of attack-aware clustering and selective cross-cluster transfer remains effective as the differences between local attack spaces become more pronounced.

While the above analysis focuses on final predictive performance under different levels of label-space heterogeneity, it does not show how the considered methods behave throughout the federated training process. We therefore further analyze their convergence behavior to assess training stability and communication efficiency.

Table 7 F1-score (%) under different levels of label-space heterogeneity.

Dataset	Method	Low	Medium	High
CIC-IoT-2023	FedAvg	88.74	84.21	76.35
	FedProx	89.31	85.07	77.42
	IFCA	90.26	86.41	79.63
	FedProto	92.17	90.18	85.94
	Attack-Fed	94.21	92.47	89.76
UNSW-NB15	FedAvg	86.41	81.73	73.62
	FedProx	87.18	82.46	74.85
	IFCA	88.36	83.94	77.12
	FedProto	90.27	87.16	82.38
	Attack-Fed	92.63	89.94	86.71

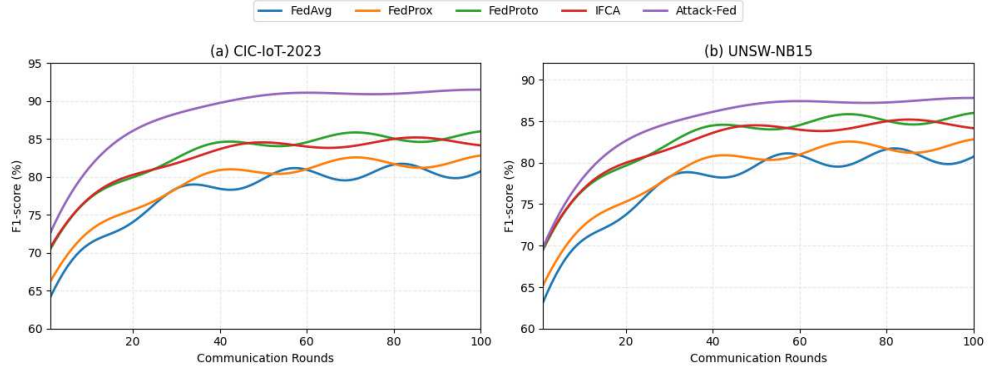


Fig. 7 F1-score evolution over communication rounds on (a) CIC-IoT-2023 and (b) UNSW-NB15.

Convergence Analysis and Communication Cost

In addition to the final predictive performance, we analyze the convergence behavior of the considered methods throughout the federated training process. The objective is to assess whether Attack-Fed provides stable optimization under heterogeneous label spaces and reaches its final performance within a reasonable number of communication rounds. The convergence behavior is evaluated on both CIC-IoT-2023 and UNSW-NB15.

Fig 7 show the evolution of the F1-score over the communication rounds on the two datasets. Attack-Fed exhibits a stable improvement during training and reaches a higher performance level than the considered baselines. The same overall trend is observed on both datasets, indicating that the attack-aware clustering and selective cross-cluster transfer contribute to a more stable optimization under label-space heterogeneity. The convergence results are complemented by an analysis of the computational overhead introduced by Attack-Fed. The proposed framework requires

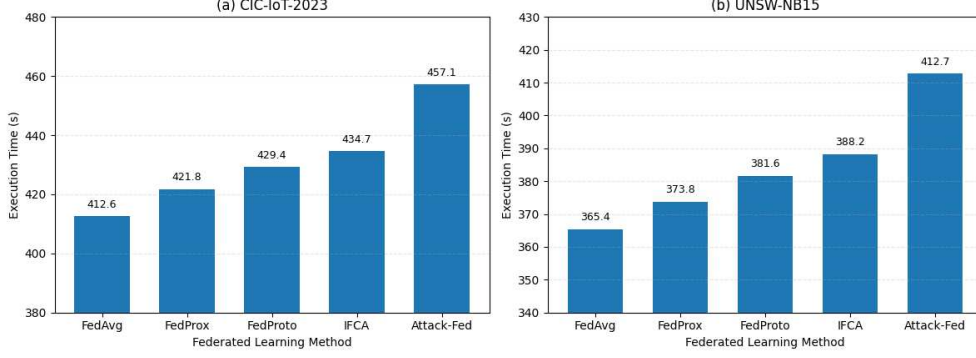


Fig. 8 Execution time of different federated learning methods on (a) CIC-IoT-2023 and (b) UNSW-NB15.

additional operations for constructing attack-category profiles, performing client clustering, and executing cross-cluster knowledge transfer. However, these operations rely on compact information rather than raw network traffic.

The convergence results further highlight the stable training behavior of Attack-Fed under label-space heterogeneity. This behavior is supported by the attack-aware client clustering and selective cross-cluster knowledge transfer, which enable clients with heterogeneous label spaces to benefit from relevant knowledge during federated training. To further assess the practical efficiency of the proposed method, Fig. 8 compares the execution time of the considered methods on both datasets. On CIC-IoT-2023, Attack-Fed completes the training process in 457.1 seconds, compared with 412.6 seconds for FedAvg. The additional processing is mainly associated with the attack-aware clustering and selective knowledge-transfer stages, which require approximately 18.6 and 24.3 seconds, respectively. A similar trend is observed on UNSW-NB15. Despite incorporating these additional attack-aware mechanisms, Attack-Fed maintains a practical execution time while providing improved performance under label-space heterogeneity.

Moreover, the proposed coordination relies on compact attack profiles and selected prototypes rather than raw network traffic. This design enables Attack-Fed to exploit cross-cluster knowledge while preserving the decentralized nature of federated training. Overall, the execution-time results demonstrate that the proposed framework can incorporate attack-aware clustering and selective knowledge transfer with a practical computational cost.

6 Conclusion and future work

In this paper, we addressed the challenge of label-space heterogeneity in federated intrusion detection systems, where participating clients may observe different subsets of attack families due to the diversity of local network environments and traffic patterns. To address this challenge, we proposed Attack-Fed, a two-level federated learning method that explicitly exploits the structure of clients local attack knowledge. At the first level, each client constructs an ACP that summarizes its local attack expertise

through complementary statistical descriptors. These profiles are used to quantify the compatibility between clients and construct an attack-aware similarity matrix. Based on this matrix, affinity propagation is employed to automatically identify groups of clients with compatible attack spaces, after which model knowledge is aggregated within each cluster. This attack-aware organization reduces the negative impact of indiscriminate collaboration between clients with substantially different label spaces. At the second level, we address the knowledge isolation introduced by cluster-wise aggregation. Although attack-aware clustering improves the compatibility of intra-cluster collaboration, independently trained clusters may still possess complementary knowledge about attack families that are absent from their local members. To overcome this limitation, we construct an attack relationship graph that captures semantic relationships between the knowledge maintained by different clusters. This graph supports selective cross-cluster knowledge propagation through prototype-guided knowledge transfer, allowing complementary attack knowledge to be exchanged without requiring direct aggregation of all cluster models. A global knowledge completion mechanism subsequently integrates the transferred knowledge to improve the overall detection capability while preserving the specialization achieved through attack-aware clustering. Experimental evaluation on heterogeneous federated intrusion detection settings demonstrates the effectiveness of the proposed method against conventional federated learning and clustered federated learning baselines. The results show that explicitly modeling the attack spaces of clients provides meaningful benefits, particularly as the degree of label-space heterogeneity increases. The ablation and robustness analyses further highlight the respective contributions of attack-aware clustering and cross-cluster knowledge transfer to the final detection performance.

Future work will investigate several directions. First, the proposed attack relationship graph could be extended to incorporate richer temporal and structural relationships between evolving attack patterns. Second, more adaptive knowledge-transfer mechanisms could be explored to determine the amount and direction of knowledge exchanged between clusters according to their evolving compatibility. Third, extending the method to dynamic federated environments, where clients and attack categories can appear or disappear over time, represents an important direction for practical deployment. In particular, future work will investigate mechanisms for detecting and incorporating newly emerging attack categories while limiting the forgetting of previously learned attacks. Such an extension would allow the method to continuously adapt its knowledge to evolving attack landscapes. Finally, evaluating the proposed method in additional intrusion detection environments would provide further insight into its scalability, communication efficiency, and robustness against evolving cybersecurity threats.

References

- [1] Ji, C., He, S., Dai, W.: A federated learning based intrusion detection method with multi-scale parallel convolution and adaptive soft prediction clustering. *Electronics* **14**(23), 4705 (2025)
- [2] Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods

- for cyber security intrusion detection. *IEEE Communications surveys & tutorials* **18**(2), 1153–1176 (2015)
- [3] Wang, Z., Liu, J., Sun, L.: Efs-dnn: An ensemble feature selection-based deep learning approach to network intrusion detection system. *Security and Communication Networks* **2022**(1), 2693948 (2022)
 - [4] Aldweesh, A., Derhab, A., Emam, A.Z.: Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems* **189**, 105124 (2020)
 - [5] Zhang, Y., Suleiman, B., Alibasa, M.J., Farid, F.: Privacy-aware anomaly detection in iot environments using fedgroup: A group-based federated learning approach. *Journal of network and systems management* **32**(1), 20 (2024)
 - [6] Truong, N., Sun, K., Wang, S., Guitton, F., Guo, Y.: Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security* **110**, 102402 (2021)
 - [7] Farnaaz, N., Jabbar, M.: Random forest modeling for network intrusion detection system. *Procedia Computer Science* **89**, 213–217 (2016)
 - [8] Badrouni, M., Inoubli, W., Katar, C., Kodja, Z.: Privacy meets personalization: a systematic literature review of federated recommender systems. *Knowledge and Information Systems* **68**(1), 146 (2026)
 - [9] Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G.: A survey on security and privacy of federated learning. *Future Generation Computer Systems* **115**, 619–640 (2021)
 - [10] Kairouz, P., McMahan, H.B.: Advances and open problems in federated learning. *Foundations and trends in machine learning* **14**(1-2), 1–210 (2021)
 - [11] Nascimento, L., Awaysheh, F.M., Alawadi, S.: Data skew in federated learning: An experimental evaluation on aggregation algorithms. In: *2024 2nd International Conference on Federated Learning Technologies and Applications (FLTA)*, pp. 162–170 (2024). IEEE
 - [12] Pei, J., Liu, W., Li, J., Wang, L., Liu, C.: A review of federated learning methods in heterogeneous scenarios. *IEEE Transactions on Consumer Electronics* **70**(3), 5983–5999 (2024)
 - [13] Chen, J., Li, M., He, X.: Fedck: addressing label distribution skew in federated learning via clustering-efficient and knowledge distillation: J. chen et al. *The Journal of Supercomputing* **81**(14), 1318 (2025)
 - [14] Kang, M., Kim, S., Jin, K.H., Adeli, E., Pohl, K.M., Park, S.H.: Fednn: Federated

- learning on concept drift data using weight and adaptive group normalizations. *Pattern Recognition* **149**, 110230 (2024)
- [15] Deng, M., Sun, C., Kan, Y., Xu, H., Zhou, X., Fan, S.: Network intrusion detection based on deep belief network broad equalization learning system. *Electronics* **13**(15), 3014 (2024)
 - [16] Han, W., Peng, J., Yu, J., Kang, J., Lu, J., Niyato, D.: Heterogeneous data-aware federated learning for intrusion detection systems via meta-sampling in artificial intelligence of things. *IEEE Internet of Things Journal* **11**(8), 13340–13354 (2023)
 - [17] Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection. *Journal of Network and Systems Management* **31**(1), 3 (2023)
 - [18] Zhao, R., Yin, Y., Shi, Y., Xue, Z.: Intelligent intrusion detection based on federated learning aided long short-term memory. *Physical Communication* **42**, 101157 (2020)
 - [19] Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K., Pan, W.: Intrusion detection for wireless edge networks based on federated learning. *Ieee Access* **8**, 217463–217472 (2020)
 - [20] Man, D., Zeng, F., Yang, W., Yu, M., Lv, J., Wang, Y.: Intelligent intrusion detection based on federated learning for edge-assisted internet of things. *Security and Communication Networks* **2021**(1), 9361348 (2021)
 - [21] Campos, E.M., Saura, P.F., González-Vidal, A., Hernández-Ramos, J.L., Bernabe, J.B., Baldini, G., Skarmeta, A.: Evaluating federated learning for intrusion detection in internet of things: Review and challenges. *Computer Networks* **203**, 108661 (2022)
 - [22] Agrawal, S., Sarkar, S., Aouedi, O., Yenduri, G., Piamrat, K., Alazab, M., Bhattacharya, S., Maddikunta, P.K.R., Gadekallu, T.R.: Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications* **195**, 346–361 (2022)
 - [23] Idrissi, M.J., Alami, H., El Mahdaouy, A., El Mekki, A., Oualil, S., Yartaoui, Z., Berrada, I.: Fed-anids: Federated learning for anomaly-based network intrusion detection systems. *Expert Systems with Applications* **234**, 121000 (2023)
 - [24] Sun, S., Sharma, P., Nwodo, K., Stavrou, A., Wang, H.: Fedmade: Robust federated learning for intrusion detection in iot networks using a dynamic aggregation method. In: *International Conference on Information Security*, pp. 286–306 (2024). Springer

- [25] Bierbrauer, D.A., Coffey, S.M., Willeke, M.R., Beggs, J.D., Bastian, N.D.: Data-efficient federated learning for edge network intrusion detection. *Engineering Applications of Artificial Intelligence* **150**, 110685 (2025)
- [26] Jullian, O., Otero, B., Rodriguez, E., Gutierrez, N., Antona, H., Canal, R.: Deep-learning based detection for cyber-attacks in iot networks: A distributed attack detection framework. *Journal of Network and Systems Management* **31**(2), 33 (2023)
- [27] Sattler, F., Müller, K.-R., Samek, W.: Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* **32**(8), 3710–3722 (2020)
- [28] Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. *Advances in neural information processing systems* **33**, 19586–19597 (2020)
- [29] Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9 (2020). IEEE
- [30] Duan, M., Liu, D., Ji, X., Liu, R., Liang, L., Chen, X., Tan, Y.: Fedgroup: Efficient clustered federated learning via decomposed data-driven measure. *arXiv preprint arXiv:2010.06870* (2020)
- [31] Islam, M.S., Javaherian, S., Xu, F., Yuan, X., Chen, L., Tzeng, N.-F.: Fed-clust: Optimizing federated learning on non-iid data through weight-driven client clustering. In: 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1184–1186 (2024). IEEE
- [32] Irtiza, S., Khan, L., Hamlen, K.W., Chan, K.S., De Lucia, M., Kantarcioglu, M., Mehrotra, S., Swami, A., Thuraisingham, B.: Clusterfed: Self-supervised federated network intrusion detection using clustering. In: MILCOM 2025-2025 IEEE Military Communications Conference (MILCOM), pp. 502–508 (2025). IEEE
- [33] Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8432–8440 (2022)
- [34] Xu, J., Tong, X., Huang, S.-L.: Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867* (2023)
- [35] Shiimoto, K.: Network intrusion detection system based on an adversarial auto-encoder with few labeled training samples: K. shimoto. *Journal of Network and Systems Management* **31**(1), 5 (2023)
- [36] Huang, C., Chen, X., Zhang, Y., Wang, H.: Fedcrl: Personalized federated learning

with contrastive shared representations for label heterogeneity in non-iid data. arXiv e-prints, 2404 (2024)

- [37] Wu, S., Chen, J., Nie, X., Wang, Y., Zhou, X., Lu, L., Peng, W., Nie, Y., Menhaj, W.: Global prototype distillation for heterogeneous federated learning. *Scientific reports* **14**(1), 12057 (2024)
- [38] Chen, L., Zhang, W., Dong, C., Zhao, D., Zeng, X., Qiao, S., Zhu, Y., Tan, C.W.: Fedtkd: A trustworthy heterogeneous federated learning based on adaptive knowledge distillation. *Entropy* **26**(1) (2024)
- [39] Breitholtz, A., Zec, E.L., Johansson, F.D.: Federated learning with heterogeneous and private label sets. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 472–491 (2025). Springer
- [40] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282 (2017)
- [41] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450 (2020)
- [42] A federated learning based intrusion detection method with multi-scale parallel convolution and adaptive soft prediction clustering. *Electronics* **14**(23), 4705 (2025)
- [43] Xu, J., Tong, X., Huang, S.-L.: Personalized federated learning with feature alignment and classifier collaboration. arXiv preprint arXiv:2306.11867 (2023)
- [44] Ay, M., Özbakır, L., Kulluk, S., Gülmez, B., Öztürk, G., Özer, S.: Fc-kmeans: Fixed-centered k-means algorithm. *Expert Systems with Applications* **211**, 118656 (2023)
- [45] Rajendran, L.K.K.: Hematological malignancy identification via k-means based roi extraction. *International Journal of Clinical Research and Medical Sciences* **1**(2) (2026)
- [46] Zhang, Y., Zheng, K., Yan, S., Wang, X., Cai, Z.: Efficient tensorized multi-view anchor graph clustering with affinity propagation for remote sensing data. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, pp. 28501–28509 (2026)
- [47] Tseng, S.-M., Wang, Y.-Q., Wang, Y.-C.: Multi-class intrusion detection based on transformer for iot networks using cic-iot-2023 dataset. *Future Internet* **16**(8), 284 (2024)

- [48] Kasongo, S.M., Sun, Y.: Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *Journal of Big Data* **7**(1), 105 (2020)
- [49] Elkhadir, Z., Begdouri, M.A.: Enhancing iot security: A comparative analysis of preprocessing techniques and classifier performance on iot23 and cic iot 2023 datasets. *IAENG International Journal of Computer Science* **52**(4) (2025)
- [50] Kumar, V., Sinha, D., Das, A.K., Pandey, S.C., Goswami, R.T.: An integrated rule based intrusion detection system: analysis on unsw-nb15 data set and the real time online dataset. *Cluster Computing* **23**(2), 1397–1418 (2020)
- [51] Fan, Z., Yao, J.: Federated learning with partially class-disjoint data. *Federated Learning-A Systematic Review: A Systematic Review*, 65 (2025)
- [52] Huang, W., Li, T., Wang, D., Du, S., Zhang, J., Huang, T.: Fairness and accuracy in horizontal federated learning. *Information Sciences* **589**, 170–185 (2022)
- [53] Wang, Z., Zhang, Y., Xu, X., Fu, Z., Yang, H., Du, W.: Federated probability memory recall for federated continual learning. *Information Sciences* **629**, 551–565 (2023)
- [54] Bakopoulou, E., Tillman, B., Markopoulou, A.: Fedpacket: A federated learning approach to mobile packet classification. *IEEE Transactions on Mobile Computing* **21**(10), 3609–3628 (2021)
- [55] Zhou, Y., Ye, Q., Lv, J.: Communication-efficient federated learning with compensated overlap-fedavg. *IEEE Transactions on Parallel and Distributed Systems* **33**(1), 192–205 (2021)
- [56] Mathew, C., Asha, P.: Fedprox: Fedsplit algorithm based federated learning for statistical and system heterogeneity in medical data communication. *J. Internet Serv. Inf. Secur.* **14**(3), 353–370 (2024)