IEEE INTERNATIONAL CONFERENCE ON

COMMUNICATIONS

9–13 June 2024 // Denver, CO, USA

Scaling the Peaks of Global Communications

IEEE ICC®

IEEE ComSoc
IEEE Communications Society

IEEE

Effective Intrusion Detection in Heterogeneous Internet-of-Things Networks via Ensemble Knowledge Distillation-based Federated Learning

Jiyuan Shen, Wenzhuo Yang, Zhaowei Chu, Jiani Fan, Dusit Niyato, Kwok-Yan Lam

College of Computing and Data Science

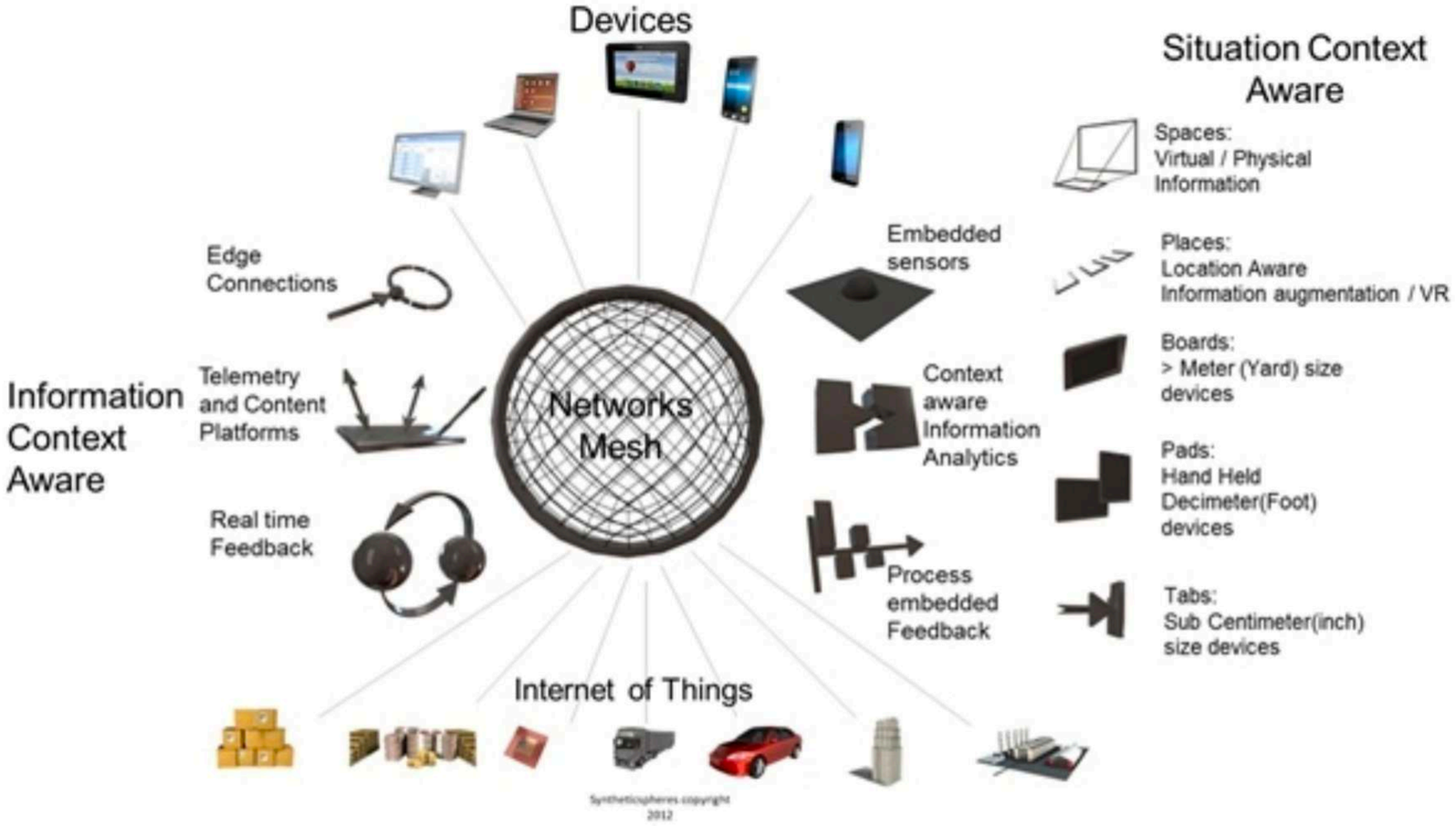
Nanyang Technological University, Singapore

Outline

- 1. Introduction
- 2. Proposed Scheme
- 3. Experiment
- 4. Conclusion

Intro-»Ubiquitous IoT Devices

- IoT devices appear in nearly everywhere of our daily life. For example, smart devices, various sensors, edge device, actuators, gadgets and hardware appliances…
- These devices record and upload our personal information from day to night, thus generating exponential data every day for every person.

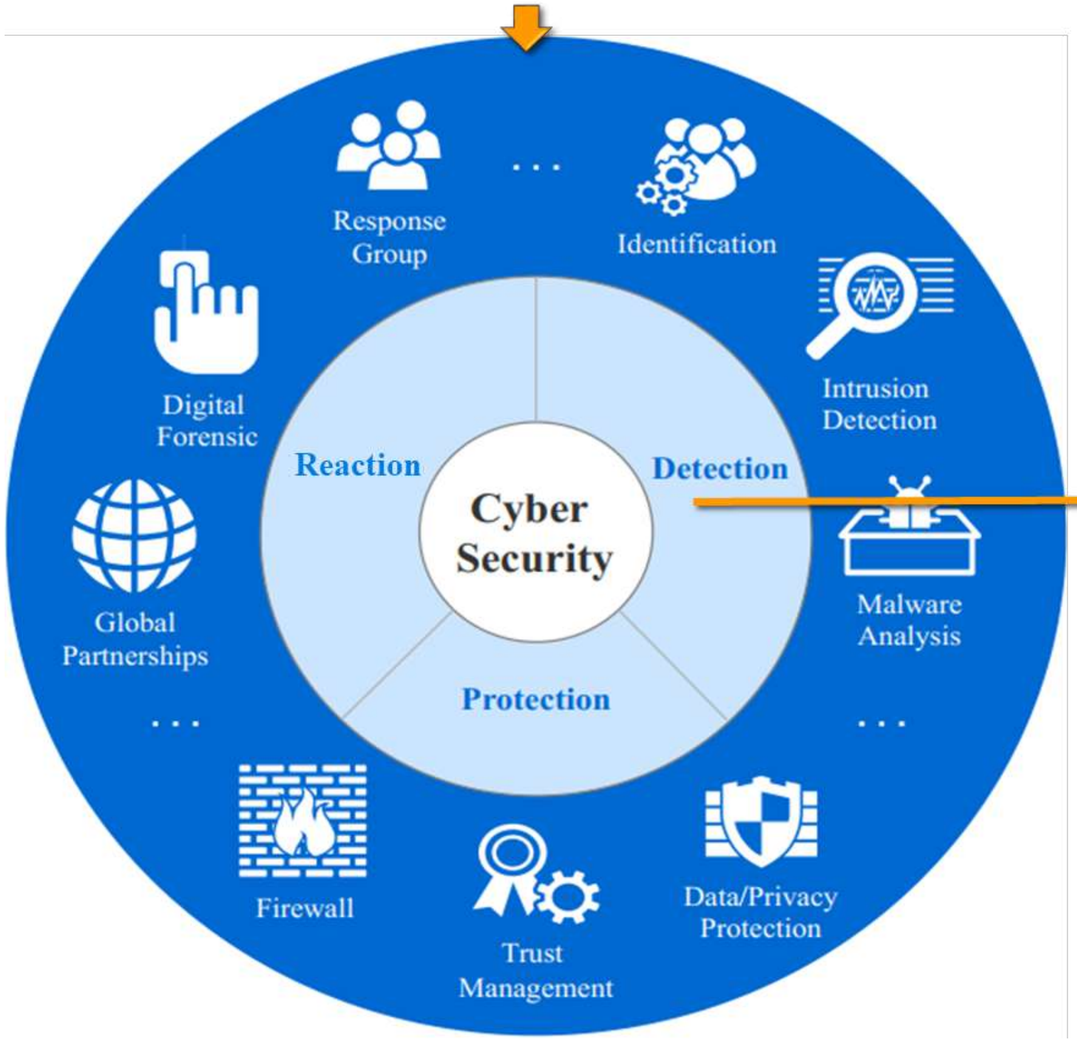


Does our data really safe?

Is it susceptible to cyber attacks?

Intro-»Cyber Attack Detection

Cybersecurity puts a lot of emphasis on detection, reaction, and protection measures such as …



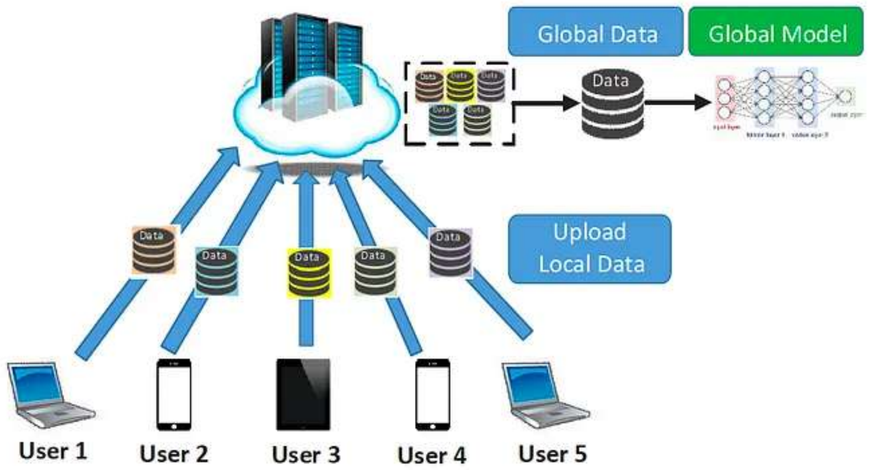
Advanced Intrusion Detection System (IDS)

- Efficient
- Effective
- Practical

Intro-»Centralized ML Detection

- Some traditional IDS use Centralized Learning:
  - All collected data will be transferred to global.
  - Training data is centralized in a machine.
  - The centralized entity trains and hosts the model.

- Cons 1: Operators have access to sensitive training data including network identity and privacy address information.
- Cons 2: The communication overhead is large and will cause privacy concerns to transferring raw data of all devices to a central server for collaborative training.



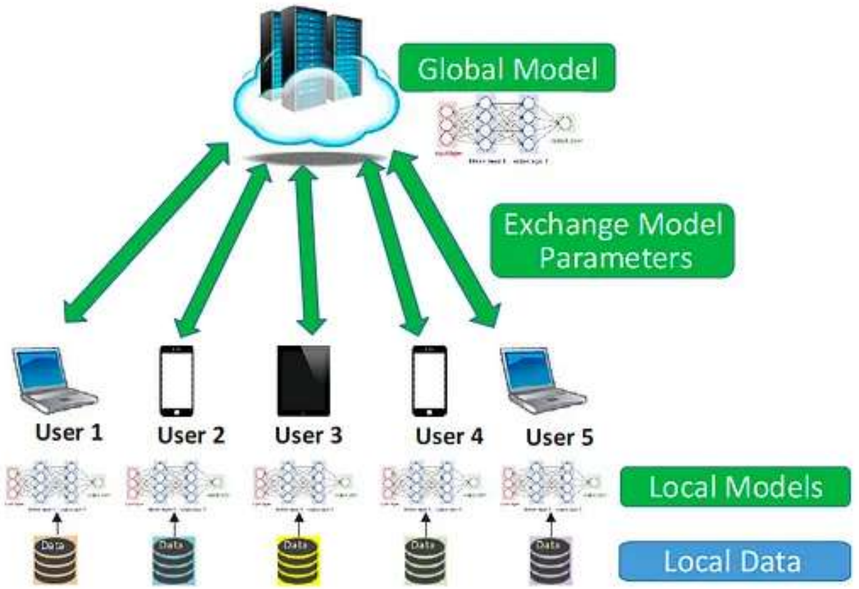
(a) centralized learning (can be outsourced learning).

Challenges for Intrusion Detection

- First, training models by local devices themselves will lead to **poor performance** because of the constrained computation ability and insufficient data.
- Second, if directly use centralized learning, the **communication overhead** is heavy since large volume of data need to be transferred.
- Third, storing user-side data in global server will cause privacy concern.
- Another main challenge is handling the **heterogeneous data** collected by different devices.
  - The data distribution, number of samples, and collection time on each edge device may be different.
  - Besides, due to system updates or different device functions, the feature dimensions and attack types of data collected from different devices/clients may also be inconsistent.

Proposed Scheme-»Decentralized Training

- We introduce federated learning as a privacy-preserving collaborative training paradigm for the IDS model.
- The conventional FL algorithm commonly involves three steps:
  - First, synchronizing the current global model parameters to maintain consistency among each client.
  - Second, updating the local model parameters on private data using Adam or SGD as optimizer.
  - Third, transmitting the models of each client to the server side and integrating them by using a specific aggregation algorithm.
- These three steps constitute a loop until the global model converges



(b) distributed learning

Proposed Scheme-»Challenges in Aggregation

- Usually, average or weighted average algorithms are used for server-side aggregation.
- However, this may not be the best aggregation mode, especially on heterogeneous data.
  - Simple averaging algorithms result in the **loss of a lot of useful information** from client models.
  - Specifically, due to the imbalance in data distribution, **some clients may be better at detecting certain attack patterns but not others**.
  - After aggregation, it is highly likely that the classification boundaries that were originally clear for certain categories **become fuzzy**, which affects the overall detection performance.

- This work aims to **better utilize the effective information** (through ensemble knowledge distillation) from each client to further improve the generalization and performance of the server model after aggregation.

Knowledge Distillation

- First proposed by Hinton, allows transferring the knowledge of a large, complex model (known as the teacher) to a smaller, simpler model (known as the student).
- The process of knowledge transfer usually needs a proxy dataset as the medium.
- Loss typically consists of two terms: a standard cross-entropy loss term and a distillation loss term. The former uses the hard label as the target while the latter uses a soft target.
- For the federated learning scenario, we usually do not have the publicly labelled dataset.
- Therefore, the combined loss is reduced to only the distillation loss term. The distillation loss term is often formulated as the Kullback-Leibler (KL) divergence between the teacher and student's softmax output probabilities.

$$\sigma(z_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

$$L_{KL}(S\|T) = \sum S_i(\sigma(z)) \log\left(\frac{S_i(\sigma(z))}{T_i(\sigma(z))}\right)$$

Proposed Scheme-»Flexible Aggregation

- We transfer the original knowledge distillation to the idea of **one-to-many knowledge distillation** within federated learning scenario.
  - ➢ It uses the server-side model obtained by simple aggregation as the student model, and client models as the teacher models.
  - ➢ The student model acquires knowledge from logits ensembled overall the received teacher models, and thus mutually beneficial information can be shared.
- Besides, we propose a **dynamic weight ensemble method** to compress the knowledge of the teacher model.
  - ➢ We first test each client model to obtain its score on the test set.
  - ➢ Then, we perform a Softmax operation on the test scores, with the addition of a deterministic temperature to enlarge the difference between clients and increase the teacher knowledge's reliance on higher-scoring clients. Therefore, the dynamic weight for each client at this round is generated.

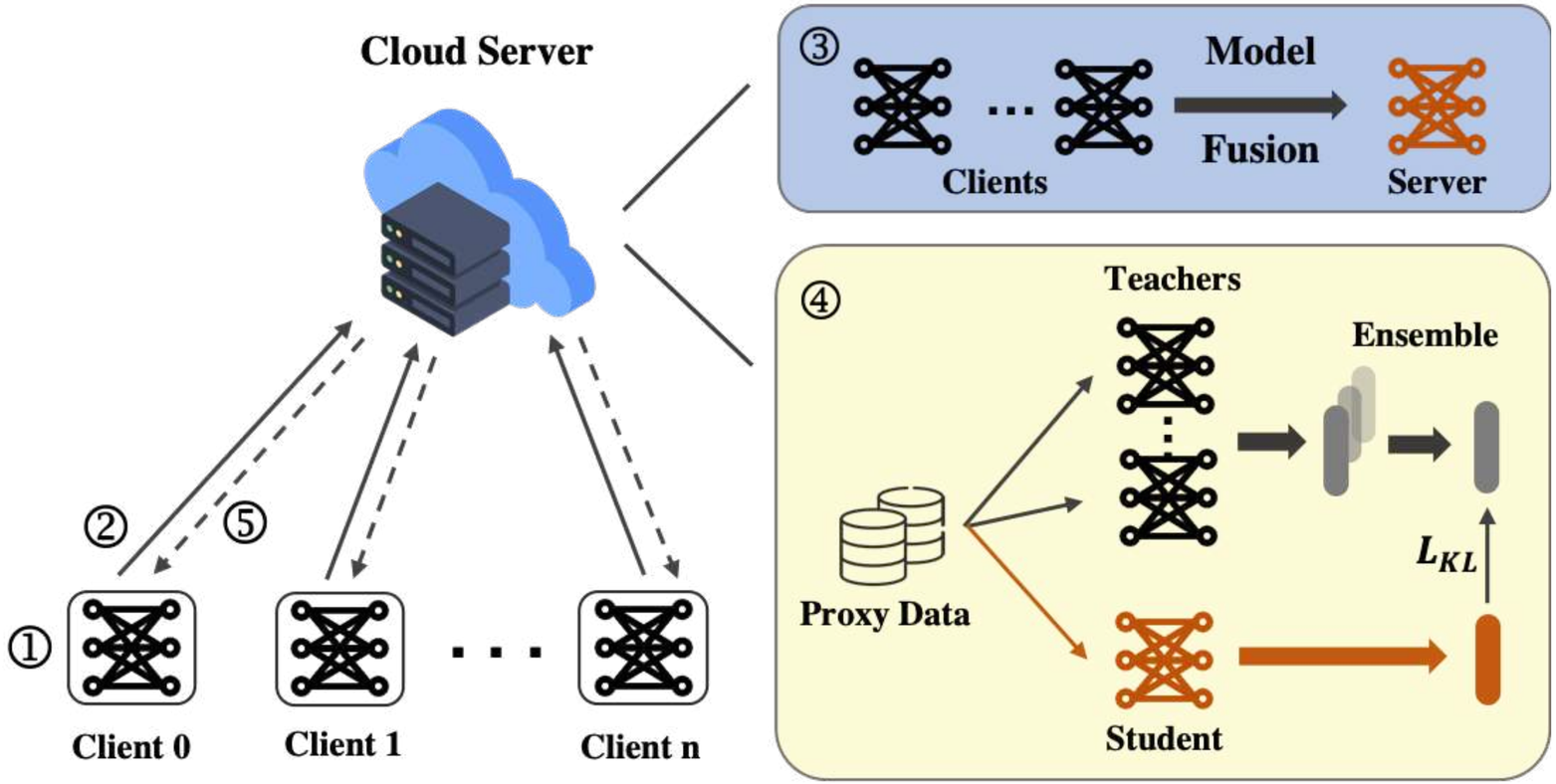
$$Client_i = \frac{\exp\left(acc_i/DT\right)}{\sum_j \exp\left(acc_j/DT\right)}$$

$$EKD = Client \times logits$$

  - ➢ Finally, we perform a matrix multiplication between the dynamic weights and the logits from each client to obtain the final ensemble knowledge. We then use KL divergence to fine-tune the server-side model.

Proposed Scheme-»Flexible Aggregation

- The overview of the FLEKD-IDS framework.
  1. Train local model.
  2. Transmit local models to server.
  3. Fuse clients' models.
  4. Ensemble knowledge distillation and fine-tune the global model.
  5. Distribute the latest global model.



Proposed Scheme-»Put it Together

Algorithm 1: Federated Learning with Ensemble Knowledge Distillation

Input: Proxy IDS dataset (unlabeled) $\mathcal{D}_0$, private IDS datasets (labeled) D_K, initialize clients models $\mathcal{W}_i$, the number of data points per client n_i.

Output: Trained server model $\mathcal{W}_G$

for *each communication round* $t = 1$ *to* T **do**
- Select a subset of clients C_t to participate in the round;
- **for** *each client* i *in* C_t **do**
  - **Synchronize** the current global model $\mathcal{W}_G$ to client i;
  - **Update** a local model $\mathcal{W}_i$ using client i's private dataset D_K;
  - **Transmit** the client models $\mathcal{W}_i$ to the central server;
- **end**
- **Model Fusion:** The server computes an updated consensus, which is an average of client models parameters $\mathcal{W}_G = \sum_{i \in C_t} \frac{n_i}{\sum_{i \in C_t} n_i} W_i$;
- **Ensemble Knowledge Distillation:**
  - Student: Calculate the logit vectors of server model $\mathbf{x}_t^s$ based on the proxy dataset $\mathcal{D}_0$;
  - Teacher: Calculate the logit vectors of client models based on the proxy dataset $\mathcal{D}_0$. Use ensemble algorithm based on Equation (3) to get the final teacher knowledge $\hat{\mathbf{x}}_t^k$;
  - **Fine-tune** the global model $\mathcal{W}_G$ using Kullback-Leibler divergence $\mathcal{L} = KL(\mathbf{x}_t^s, \hat{\mathbf{x}}_t^k)$;
- **end**

end

Return $\mathcal{W}_G$

Experiments-»Dataset & Parameters Setting

- We choose CIC-IDS2019 to evaluate and compare the performance of the proposed IDS model with other baseline methods.
  - ➢ It contains normal and the latest common DDoS attack events, similar to real data (PCAPs).
  - ➢ It also includes network traffic analysis results using CICFlowMeter-V3, which is based on timestamp, source and destination IP, source and destination port, protocol and attacking tag stream.
- Client number is set to 9.
- Adam optimization algorithm
- Learning rate: 0.001
- Weight decay: 0.0005
- Train every IDS model several times to get the average performance and time cost values.
- Use Dirichlet distribution to simulate the non-iid distribution case:
  - ➢ We set α=10, 1, 0.5.
  - ➢ Smaller values will result in more heterogeneous data distributions.

Experiments-»Evaluation Metrics

- **Precision**: represents how many positive samples out of all positive predictive samples are really positive.
- **Recall**: is also known as true positive rate (TPR) or detection rate (DR). It reflects the sensitivity of the model to identify positive (anomaly) samples from all real positive samples.
- **F1 score (F1s)**: is used to evaluate a binary classification model. It is the harmonic mean of recall and precision, which can give a more comprehensive assessment for the evaluated network intrusion detection model.

		Predicted Class	
		Anomaly	Normal
Actual Class	Anomaly	True Positive(TP)	False Negative(FN)
	Normal	False Positive(FP)	True Negative(TN)

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F\ Score = 2\frac{Recall \times Precision}{Recall + Precision}$$

Experiments-»Experiment Settings

- F1 score combines the Precision and Recall score, which is an important indicator to measure the detection performance of the IDS.
- We use F1 score in the following experiments to evaluate our proposed method.

Object	Properties
CPU	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz, 8 Cores
Operating System	CentOS Linux 7
RAM	128G
Python Version	3.8
PyTorch Version	1.9.0
scikit-learn Version	0.23.2

The experimental environment details and used software information.

Experiments-»Main Results

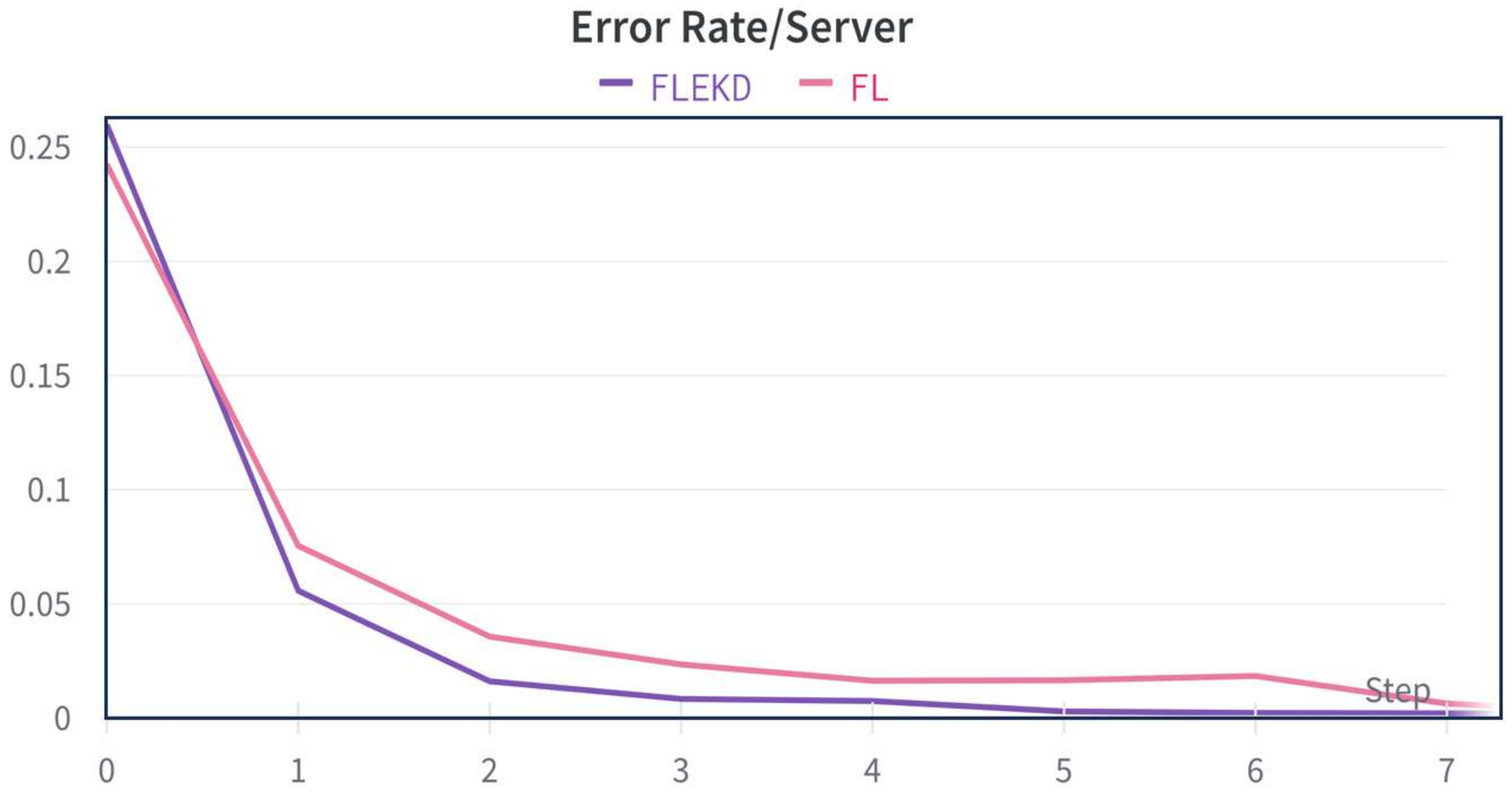
The comparison between local and distributed training under different Dirichlet distribution.

TABLE I
MAIN RESULTS. CLIENTS 0-8 CONDUCT LOCAL TRAINING AND ALL THE SCORES ARE CALCULATED BY F1-SCORE.

Local	**Portmap**			**LDAP**			**MSSQL**			**NetBIOS**			**Syn**			**UDP**			**UDPLag**		
Training	α =10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5
Client 0	99.89%	99.74%	70.08%	99.83%	99.78%	0.00%	99.96%	99.85%	99.97%	99.84%	99.69%	99.54%	99.91%	99.90%	99.92%	99.98%	99.96%	92.73%	97.27%	96.77%	97.99%
Client 1	99.88%	99.89%	99.51%	99.86%	99.80%	96.59%	99.96%	99.97%	95.78%	99.88%	99.83%	99.72%	99.93%	99.94%	98.78%	99.98%	99.92%	99.96%	97.52%	98.42%	0.00%
Client 2	99.92%	99.82%	99.66%	99.94%	99.92%	99.83%	99.93%	99.93%	99.71%	99.94%	99.87%	99.81%	99.93%	99.90%	99.87%	99.99%	100.00%	99.46%	97.30%	97.03%	76.48%
Client 3	99.81%	99.60%	98.96%	99.17%	99.76%	96.76%	99.89%	99.95%	99.92%	99.80%	99.51%	98.70%	99.92%	99.89%	99.93%	99.29%	99.90%	99.94%	97.01%	97.11%	56.63%
Client 4	99.84%	2.30%	64.93%	99.92%	99.86%	99.77%	99.90%	66.47%	94.97%	99.89%	77.55%	0.00%	99.92%	99.90%	99.11%	99.99%	99.86%	99.73%	97.40%	94.12%	67.13%
Client 5	99.83%	99.97%	98.91%	99.90%	99.90%	83.64%	99.93%	99.98%	98.77%	99.89%	99.90%	99.52%	99.87%	99.92%	99.86%	99.96%	99.99%	75.67%	97.03%	97.63%	89.07%
Client 6	90.65%	95.13%	8.94%	99.78%	99.84%	99.29%	80.23%	67.99%	2.42%	93.57%	39.35%	52.28%	99.83%	99.84%	98.95%	99.99%	99.99%	99.91%	93.55%	93.28%	46.65%
Client 7	94.86%	14.50%	32.67%	99.84%	99.89%	99.71%	89.36%	21.53%	13.57%	95.19%	70.78%	53.99%	99.81%	99.90%	99.05%	99.91%	99.99%	99.99%	94.22%	93.05%	54.84%
Client 8	97.19%	91.88%	96.53%	99.82%	99.58%	94.26%	93.11%	84.77%	0.00%	95.94%	95.06%	68.50%	99.89%	99.92%	99.82%	99.94%	99.90%	97.11%	93.31%	95.59%	49.36%
Distributed	**Portmap**			**LDAP**			**MSSQL**			**NetBIOS**			**Syn**			**UDP**			**UDPLag**		
Training	α =10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5	10	1	0.5
FL	99.00%	99.45%	99.54%	99.67%	99.64%	99.83%	99.98%	99.86%	99.62%	98.97%	99.57%	99.73%	99.93%	99.93%	99.88%	99.87%	99.78%	99.99%	97.55%	97.21%	95.40%
FLEKD	**99.80%**	**99.73%**	**99.77%**	**99.84%**	**99.70%**	**99.91%**	**99.99%**	**99.98%**	**99.90%**	**99.82%**	**99.73%**	**99.86%**	**99.94%**	**99.93%**	**99.92%**	**99.91%**	**99.88%**	**99.99%**	**98.34%**	**97.64%**	**96.62%**

Experiments-»Main Results

Our proposed ensemble knowledge distillation-based fine-tuning further **enhances the performance** of our global model and **accelerates the convergence rate**.



Experiments-»Practical Scenarios

The impact of different dimensions:

- We consider the possibility that data collected from different time periods may have varying dimensions.
- For instance, the CICIDS2017 dataset contains only 24 feature dimensions compared to CICIDS2019 with 82 dimensions.
- In real-world application scenarios, it is necessary to combine data collected from different IoT devices for training.
- We compare the results of training with datasets of different dimensions locally and with federated learning.

TABLE II
THE IMPACT OF DIFFERENT DIMENSIONS.

	Precision	Recall	F1-score
82 dim	95.93%	96.39%	96.02%
79 dim	86.31%	86.75%	86.17%
24 dim	84.17%	85.48%	84.71%
FL	99.37%	99.36%	99.37%
FLEKD	**99.80%**	**99.80%**	**99.80%**

Experiments-»Practical Scenarios

The impact of different sample sizes:

- Due to the characteristics of network attacks, it may occur in short periods with high frequency and target certain vulnerable devices.
- There may be significant differences in the number of samples between IoT devices.
- We divided the dataset into three groups of clients with different sample sizes. Specifically, the number of samples in each group differs by a factor of ten, with the group containing the least number of samples represented as ‘Base’.

TABLE III
THE IMPACT OF DIFFERENT SAMPLE SIZES.

	Precision	Recall	F1-score
Base	84.17%	85.48%	84.71%
10*Base	92.97%	88.55%	87.25%
100*Base	99.51%	99.50%	99.50%
FL	99.37%	99.36%	99.37%
FLEKD	**99.80%**	**99.80%**	**99.80%**

Experiments-»Practical Scenarios

The impact of different attack category distributions:

- It is common to encounter unknown attacks but traditional centralized IDSs unable to identify novel attacks effectively. However, through using proposed FLEKD, the unknow targeted attacks can be identified since other clients may encounter these and transfer the knowledge by flexible aggregation.
- For example, we find that even the most difficult-to-detect ‘UDPLag’ can reach 80.86%, while other corresponding attack categories can be improved to about 99%.

TABLE IV
THE IMPACT OF DIFFERENT ATTACK CATEGORY DISTRIBUTIONS.

	0: Portmap	1: LDAP	2:MSSQL	3:NetBIOS	4: Syn	5: UDP	6: UDPLag
Drop Label 0	0.00%	98.72%	59.09%	95.62%	99.67%	95.10%	0.00%
Drop Label 1	99.30%	0.00%	66.45%	99.62%	99.91%	99.98%	94.95%
Drop Label 2	66.77%	99.96%	0.00%	99.75%	99.95%	99.99%	96.81%
Drop Label 3	91.06%	97.03%	54.03%	0.00%	99.68%	95.84%	0.00%
Drop Label 4	75.76%	99.69%	94.54%	95.72%	0.00%	99.90%	11.05%
Drop Label 5	99.74%	66.62%	99.78%	99.83%	99.92%	0.00%	97.47%
Drop Label 6	91.21%	97.93%	0.00%	72.35%	99.65%	94.46%	0.00%
FL	99.05%	99.80%	94.37%	97.31%	99.92%	98.85%	75.28%
FLEKD	**99.66%**	**99.88%**	**98.97%**	**99.80%**	**99.94%**	**99.97%**	**80.86%**

Conclusion

- We introduce an FL framework to **develop an on-device collaborative deep intrusion detection model** for edge devices in IoT networks.

- We propose a **dynamic weight ensemble knowledge distillation scheme** (FLEKD) to assist in mitigating the negative influence of clients' heterogeneity without violating users' personal privacy.

- We conduct extensive experiments on the public dataset CICIDS2019 to demonstrate better detection performance and improved ability to identify unknown attacks over local training models and naive FL global models.

- We assess the performance of our proposed framework in **three possible real-world scenarios**, namely, **diverse data features, sample quantity, and missing certain classes**. Our experimental results demonstrate that FLEKD exhibits clear and strong advantages.