Federated Threat-Hunting Approach for Microservice-Based Industrial Cyber-Physical System

Mohamed Abdel-Basset, Hossam Hawash, Karam Sallam

Supervisor: Prof. Jong Hyuk Park

Presented By : Bhagyashree Kakde

2022.11.14

Seoul National University of Science and Technology, Seoul, South Korea

Table of content

Abstract

- This Paper presented the novel federated deep learning (DL) model (Fed-TH) for hunting cyber threats against industrial cyber-physical systems (ICPSs) that captures the temporal and spatial representations of network data.

- The container-based industrial edge computing framework is designed to deploy this model as a threat-hunting microservice on suitable edge servers.

- An exploratory microservice placement method is introduced to enable better microservice deployment to tackle the latency issue.

1. Introduction

- Cyber-physical systems (ICPS) consist of different IoT technologies, such as software-defined networking, fifth-generation (5G) and beyond 5G (B5G) networks, cloud computing, mobile edge computing (MEC), and standard industrial control systems.

- Heterogeneous, wide-scale, distributed, and intricate nature of an ICPS causes many common and application-related vulnerabilities that can be abused by attackers to execute malicious actions.

- For Increasing data transfer models, MEC intends to bring computations from the cloud to a network's edges to minimize the network's latency and bandwidth congestion.

- Deep learning (DL) has shown promising performances in developing efficient threat intelligence (TI) in autonomous industrial (IIoT) networks. It is challenging to develop a distributed DL at edge nodes owing to resource constraints and privacy matters, including data eavesdropping and leakage.

- Federated learning (FL) is an encouraging approach that has achieved great success in delivering intelligence to the edge layers of IoT networks with participating devices using local data. Eliminates privacy issues by sharing local updates instead of raw data.

1. Introduction(cont..)

A. Main Challenges:

a) Security: The combination of ICPS and 5G network allows cyber threat which affects on the industrial process.

b) Heterogeneity: microservice design is prone to large slowdowns and/or run-time collisions because of the heterogeneity of ESs stemming from high variations in transient network interrupts memory footprints and the frequencies of central processing unit (CPU) cycles.

c) Privacy: Current FL cannot fully guarantee the preservation of the privacy of participating entities as the central authority (cloud) is often presumed to be completely trustworthy for managing training.

d) Latency: ICPS often entails time-critical tasks, rapid responses and low latency values are necessary.

B. Primary Contributions :

To overcome Mention challenges author proposed a new threat-hunting approach for IIoT networks is briefly discussed in the following.

2) A DL model (Deep-TH) for detecting cyber threats in an ICPS is presented.

3) Container-based industrial edge computing (CIEC) framework is introduced to incorporate the Deep-TH in a differentially private FL framework referred to as Fed-TH.

2. Literature Review

The methodology for conducting the experiments in this study is discussed under three different aspects including the empirical settings, evaluation datasets, and performance measures.

A. Cyber-Threat Intelligence for ICPSs

- In this context, author Li et al. introduced a new DL approach for recognizing intrusions in ICPSs, with a gated recurrent unit (GRU) and convolutional neural network (CNN)to show different classes of attacks.
- Zhou et al. [10] addressed intrusion detection with a variational LSTM model that employs an encoder-decoder architecture accompanied by a variational method to learn the low-dimensional patterns from high-dimensional IoT data. following are shortcoming:
  - They ignored the data's heterogeneity during training which is common for ICPSs.
  - The majority of ICPS-related TI methods were designed to hypothesise that enough excellent samples of cyber threats on ICPSs were constantly obtainable for designing intelligent detection models. realistically, one ICPS terminal has few cyber-threat samples causes the development of a DL model challenging.
  - ICPS holders are customarily unwilling to disclose their data due to the sensitivity and privacy of data hence developing a reliable DL model for cyber-threat hunting or detection in ICPSs is a complex challenge.

B. Microservice-Based IoT Applications

- Microservice-based solutions have been developed to improve the flexibility of IoT networks.
- Many Papers were researched with different technology but Unfortunately, none of these studies considered a microservice design for TI applications.
- The development of smart industrial services consisting of multiple microservices with intrinsically complicated dependencies still requires careful planning to achieve the efficient latency of TI services on the edge side of a network.

3. System Design

The framework of the proposed system consists of three primary components, i.e., containerized edge nodes (e.g.,edge services(ESs) or mobile devices), a cloud server, and an IoT network.

A. Cloud Backend

The cloud backend is responsible for constructing a final threat-hunting model by federating the parameters of an edge-trained one. It manages the distributed training of Fed-TH.

Consist of four main distinct modules

1. Resource Manager (R): Retain the resource profile of each Industrial agent involved in federated training. The system takes into account edge resources.
2. Service Manager (S): Retain the profile of every active service/microservice. microservice reliance, resource obligations.
3. Scheduler (S): Liable for deciding on the strategy for deploying threat-hunting microservices.
4. Deployer (D): Accountable for deploying and initiating all threat-hunting microservices.

B. Microservice-Based IoT Applications

- Microservice-based solutions have been developed to improve the flexibility of IoT networks. many studies are carried out but Unfortunately, none of these studies considered a microservice design for TI applications.
- Development of smart industrial services consisting of multiple microservices with intrinsically complicated dependencies still requires careful planning to achieve the efficient latency of TI services on the edge side of a network.

C. Containerized Edge Tier

- Edge Services entities are computational entities of CIEC responsible for delivering remote edge-based microservices. The ESs communicate with the IPs through a direct channel to decrease broadcast dormancy.

D. Microservice Placement

- Exploratory microservice placement (EMP) method (Algorithm 1) is introduced to compare the computational delays and local and remote discharging. This process consists of two procedures
  - An offloading policy determination whereby
  - A microservice placement whereby

Algorithm 2: Federated Training of Fed-TH.

Input: N subsets of database $\{D^1, D^2, \cdots, D^N\}$, batch size B.

Output: Federated optimal parameters (**w**) of Fed-TH.

Set initial parameters w^0 for all participants.
$t = 0$.
ESs perform.
    For $i = 1$ *to* N **do:**
        The ith ES request gradients (gg^{t-1}) from cloud.
        The local parameters (w_i^t) updated by (21).
        Calculate the noised parameter ($\overline{\overline{w}}_i^t$) using (24) to achieve LDP.
        The $i$thES compute the local gradients lg_i^t by (20).
        Upload the filtered gradient by (23).
        $t = t + 1$
Cloud performs.
Aggregate gradients form ESs.
    Calculate the mean gradient (gg^t) by (22).
    Broadcast filtered gradient (gg^t) by (23) to the ESs.
    Repeat from lines 3 to 14 until the Fed-TH converges.
Return the final parameters ($\boldsymbol{w}$) of Fed-TH.

Algorithm 1: EMP Method

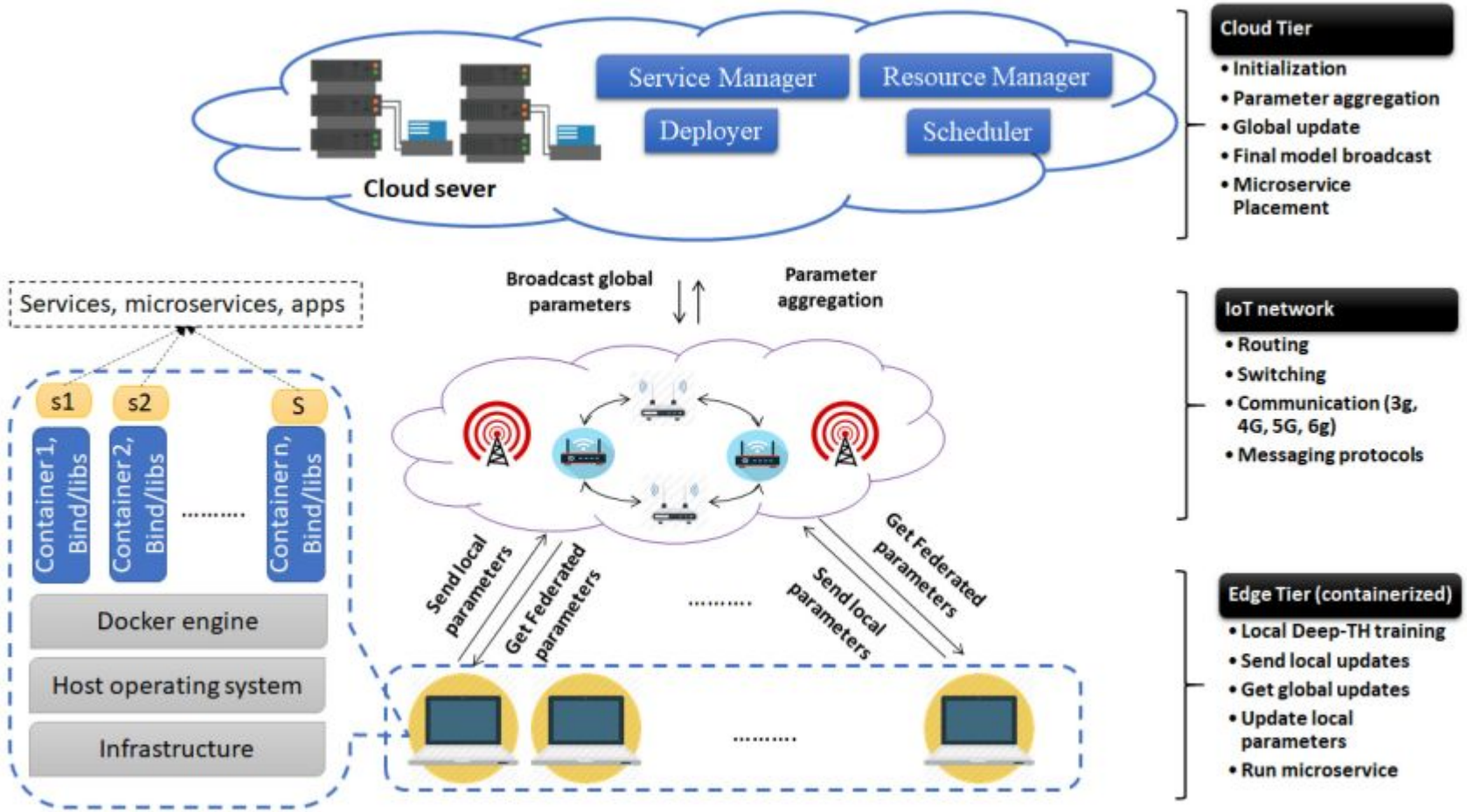

Fig. 1 Systematic diagram indicating the system model of proposed for federated threat/attack detection in microservice-based ICPS

4.Proposed FED-TH

In this section design of the author's proposed model for cyber threat hunting in an edge-based ICPS is described. In Fig. 2, a systematic diagram of its structural design is comprised of three main blocks.

A. Multiscale Spatial Representation Block

- The main role of this block is to learn efficient spatial representations in ICPS data sequences.
- Finds a high-level spatial representation which causes it to miss or leak information across layers.
- In the proposed model two MSR blocks are stacked to learn spatial representations in the data.

B. Temporal Learning (TL) Block

- This is a different enhanced TL block that, for the first time, involves an AE and GRU (AE-GRU) inspired by a Recurrent Neural Network-based AE.

C. Decision Block

- This block takes the extracted representations and processes them to obtain the final classification decision using FCLs and SoftMax.

D. Federated Training

- Each ES trains its own Deep-TH locally on its local dataset using the adaptive stochastic gradient descent algorithm.

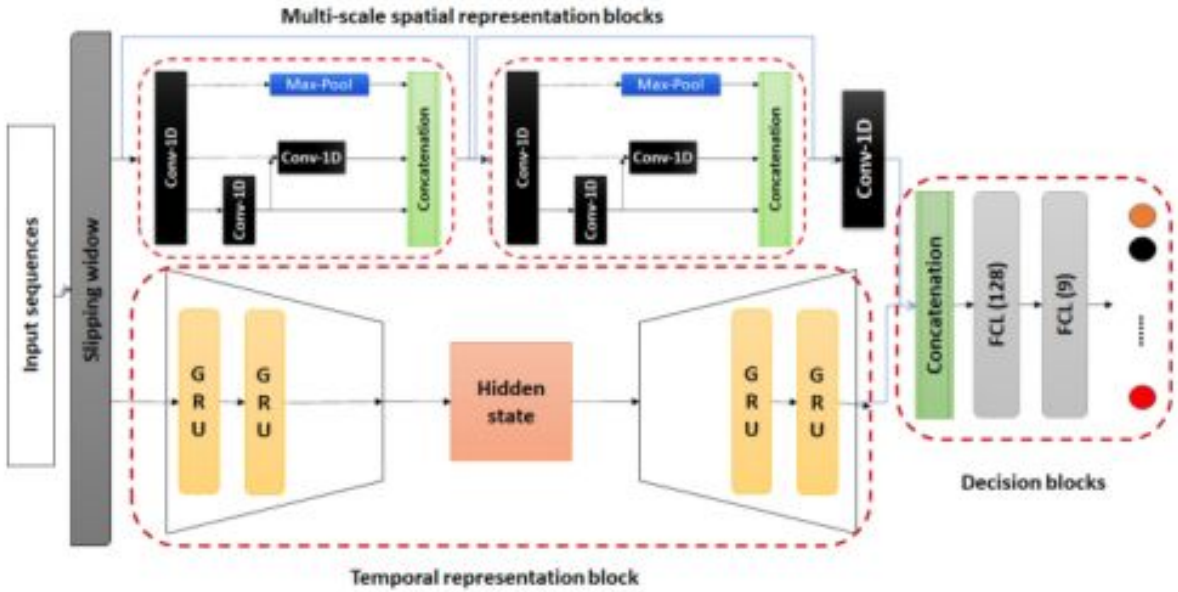

Fig 2. Architecture of proposed Deep-TH model

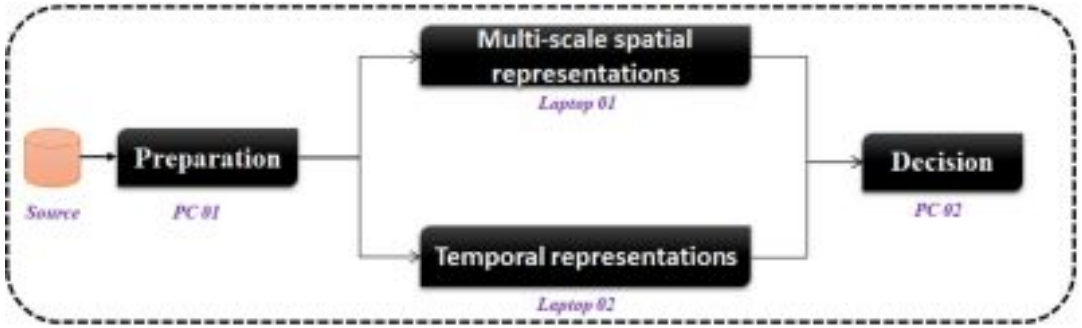

Fig 3. DAG representations of Fed-TH service

5. Experimentation Strategy

The methodology for conducting the experiments in this study is discussed under three different aspects including the empirical settings, evaluation datasets, and performance measures.

A. Empirical Settings

The implementation process for the models is coded in a Python 3.7 environment using a PyTorch library. Optimal hyperparameters of the proposed Fed-TH are determined based on grid-search experiments in Table 1.

B. Descriptions of Datasets

The proposed model is evaluated using the ToN_IoT [8] and LITNET-2020 datasets. contains aggregated labelled IoT/IIoT data including heterogeneous data sources of IoT traffic. Its distribution information is presented in table 2. its main characteristics and distributions are shown in table 3.

C. Data Prepreparation

some preprocessing steps are applied to the raw data. to avoid the effect of large discrepancies between the values of different features, min–max normalization is applied to rescale them into the range of [01] as

$$x_{\text{normalized}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Parameters represent the current instances. To mitigate the impact of class imbalance, the adaptive synthetic sampling technique is employed to adaptively generate more synthetic samples.

FL experiments, the training data are equally distributed across the containerized edge nodes.

Table 1. Hyperparameters of proposed FED-TH

Hyperparameter	Value
Batch size	128
No. of epochs	150
No.r of rounds	20
Optimizer	Adam
Initial learning rate	0.007
No. of GRU layers	2
No. of MSR blocks	2

Table 2. Characteristics of ToN_IoT Dataset

Device	No. of samples	Features	Class	Distribution
Fridge	587076	Ts,Date,Time,Fridge_tempreature, Temp_condition,Date,Time, Door_state,Sphone_signal, Latitude,Longitude, Motion status, Signal status, FC1_Read_Input_Register,FC2_Read_Descret_ Value, FC3_Read_Holding_Register, FC4_Read_Coil,Current_temperature, Thermostat_status, Temperature, Humidity, Pressure, Label, Type	Backdoor	246136
Garage	591446		Ddos	53992
GPS	595686		Injection	50319
Modbus	287194		Normal	3086973
Motion	452261		Password	142674
Thermostat	442228		Ransomware	16030
Weather	650242		Scanning	3973
/	/		Xss	6037

Table 4. Characteristics of LITNET-2020 Dataset

Exporter	Features	Class	Distribution
Kaunas–Vytautas Magnus University and Kaunas Technological University (KTU), Vilnius Gediminas Technical University, Klaipeda University, Siauliai University, KTU Panevezys Faculty of Technologies and Business Kaunas University of Technology	ts_year, ts_month, ts_day, ts_hour, ts_min, ts_second, te_year, te_montht, _day, te_hour, te_min, te_second, td, sa, da,sp, dp, pr, _flag1, __flag2, _flag3,_flag4, _flag5,_flag6, fwd, stos, ipkt, ibyt, opkt, obyt, _in, out, sasdas, smk,dmk, dtos, dir, nh, nhb, svln, dvln, ismc, odmc, idmc, osmc, mpls1, mpls2, mpls3, mpls4, mpls5, mpls6, mpls7, mpls8, mpls9, mpls10, cl, sl, al, ra, eng, exid, tr, icmp_dst_ip_b,icmp_src_ip udp_dst_p, tcp_f_s, tcp_f_n_a, tcp_f_n_f, tcp_f_n_r, tcp_f_n_p, tcp_f_n_u, tcp_dst_p, tcp_src_dst_f_s, tcp_src_tftp, tcp_src_kerb, tcp_src_rpc,tcp_dst_p_src, smtp_dst, dp_p_r_range, p_range_dst, udp_src_p.	Packet fragmentation attack (A1)	477
		Spam bot's detection (A2)	747
		Reaper Worm(A3)	1176
		Scanning/Spread(A4)	6232
		ICMP-flood(A5)	11,628
		HTTP-flood (A6)	22,959
		Blaster Worm(A7)	24,291
		LAND attack(A8)	52,417
		Smurf(A9)	59,479
		UDP-flood(A10)	93,583
		Code Red Worm(A11)	1,255,702
		TCP SYN-flood (A12)	3,725,838

6. Experiments and Analysis

A. Results :

- experiments conducted in the paper study evaluate the proposed framework under a multiclass scenario.
- In table 4 it can be observed that the normal and XSS classes have high precision values of 94.88% and 92.10%, respectively.
- In Table 5, it is clear that the code red worm and TCP SYN-flood classes achieve high precision values of 93.96% and 93.59% .representing the capabilities of Fed-TH to efficiently recognize different forms of cyber-physical attacks.

B. Comparative Analyses:

- With the ToN_IoT dataset, Fed-TH achieves a performance improvement (accuracy 2.6% and F1-score 1.6%).
- With LITNET-2020 dataset, , Fed-TH achieves a performance improvement (accuracy 2.2% and F1-score 1.9%).

Table 4. Confusion Matrix of FED-TH on TON_IOT Dataset

		Predicted Classes									
		Backdoor	DDos	Injection	Normal	Password	Ransomware	Scanning	XSS	**R** (%)	**F1** (%)
Actual Classes	Backdoor	44476	301	101	3433	289	131	287	209	90.35%	90.91%
	DDos	211	27905	305	1647	167	227	199	137	90.61%	90.29%
	Injection	105	222	27898	1123	188	234	193	101	92.80%	91.95%
	Normal	2871	1891	1245	206427	1134	1453	1131	1243	94.95%	94.92%
	Password	305	103	271	1156	26094	159	249	198	91.45%	91.27%
	Ransomware	183	129	323	1417	354	26163	313	324	89.58%	90.03%
	Scanning	243	209	206	1139	205	341	28209	243	91.60%	91.62%
	XSS	228	254	266	1213	211	207	201	28627	91.73%	91.92%
	P (%)	91.47%	89.98%	91.13%	94.88%	91.10%	90.48%	91.64%	92.10%		

Table 5. Confusion Matrix of FED-TH on F LITNET-2020 Dataset

		Predicted Classes													
		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	**R**(%)	**F1**(%)
Actual Classes	A1	10095	15	8	24	5	7	46	11	16	2	573	493	89.38	89.58
	A2	5	10276	6	3	11	7	0	6	18	2	502	513	90.55	90.10
	A3	7	22	12940	39	18	31	29	103	44	16	502	484	90.90	90.48
	A4	71	89	62	14483	33	47	28	27	31	51	511	613	90.26	90.54
	A5	24	59	18	15	14968	11	9	6	7	8	599	602	91.68	90.76
	A6	113	84	23	4	8	13409	3	14	7	5	488	434	91.89	91.12
	A7	89	21	21	2	5	15	15282	11	3	0	613	796	90.65	90.49
	A8	34	54	88	11	19	0	57	14608	33	55	822	702	88.62	89.84
	A9	43	97	31	24	0	10	23	5	15884	7	819	953	88.76	88.33
	A10	16	29	14	13	18	7	8	16	28	17612	1109	1847	85.01	86.94
	A11	323	302	415	717	774	516	813	615	913	1039	236442	8271	94.15	94.06
	A12	424	413	743	612	798	779	621	614	1084	999	8648	229433	93.58	93.59
P (%)		89.78	89.66	90.05	90.82	89.86	90.36	90.32	91.10	87.91	88.97	93.96	93.59		

6. Experiments and Analysis (cont..)

D. Federated Versus Central Learning

- The performance of the proposed Fed-TH is with local variants built locally with a centrally built version of the Fed-TH trained on all data samples as shown in fig 5.
- A threat-intelligence solution suitable for all ICPS owners because of its high efficiency, awareness of heterogeneity and privacy-preserving characteristics.

E. Number of Communication Rounds

- In fig.6 graph analyze the number of communication rounds required for convergence during the training procedures. 15 and 12 rounds which shows efficient communication overhead.

F. Tradeoff Between Privacy and Accuracy

- The proposed Fed-TH is evaluated to analyze its variations in accuracy under various privacy budgets. Obtaining greater privacy necessitates a smaller value which implies that more Gaussian noise must be added. From fig 7 shows it is found satisfactory.

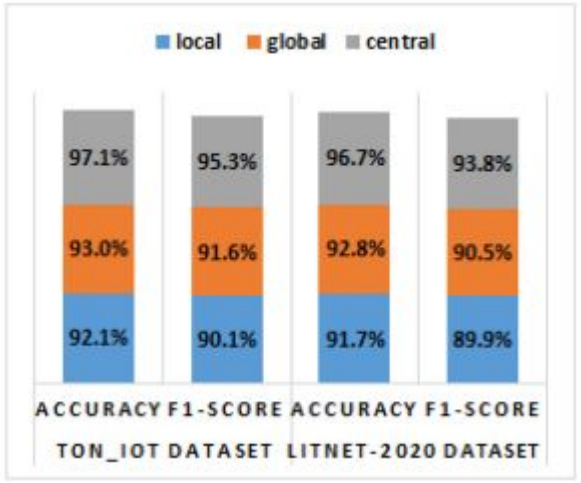

Fig 5.Classification performance under federated, local and central Fed-TH

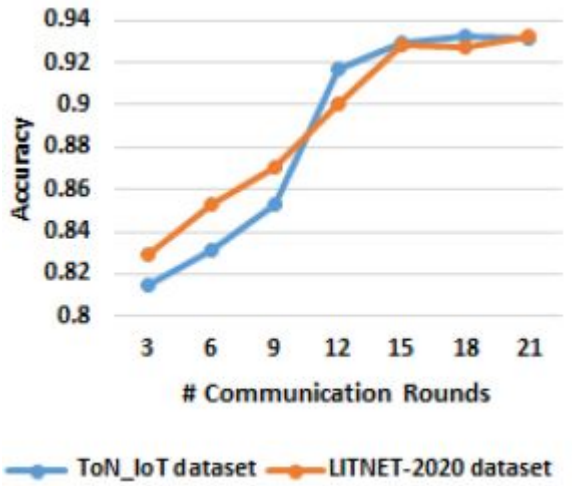

Fig 6. Accuracy under different numbers of communication rounds

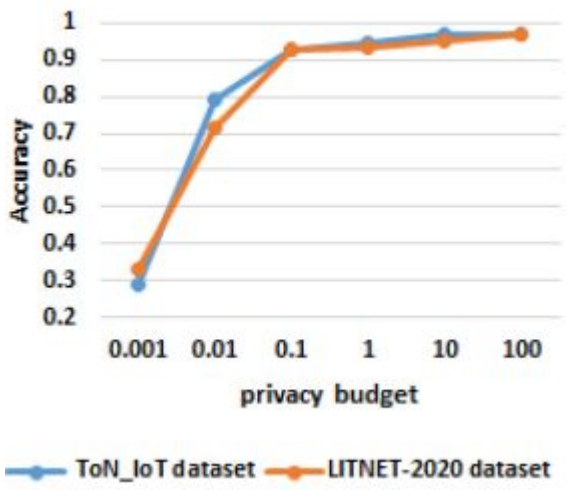

Fig7.Classification accuracy under different privacy budgets

6. Experiments and Analysis (cont..)

G. Microservice Placement in CIEC Framework:

Performances of microservice placements are evaluated in terms of average latency.

1) Comparative Analysis: It can be seen that in fig. 8 a cloud placement obtains the worst latency owing to its communication delay. The random and FIFO methods achieve high latency values in the CIEC framework since unaware of network information and resource consumption.

There are various graphs comparing with latency

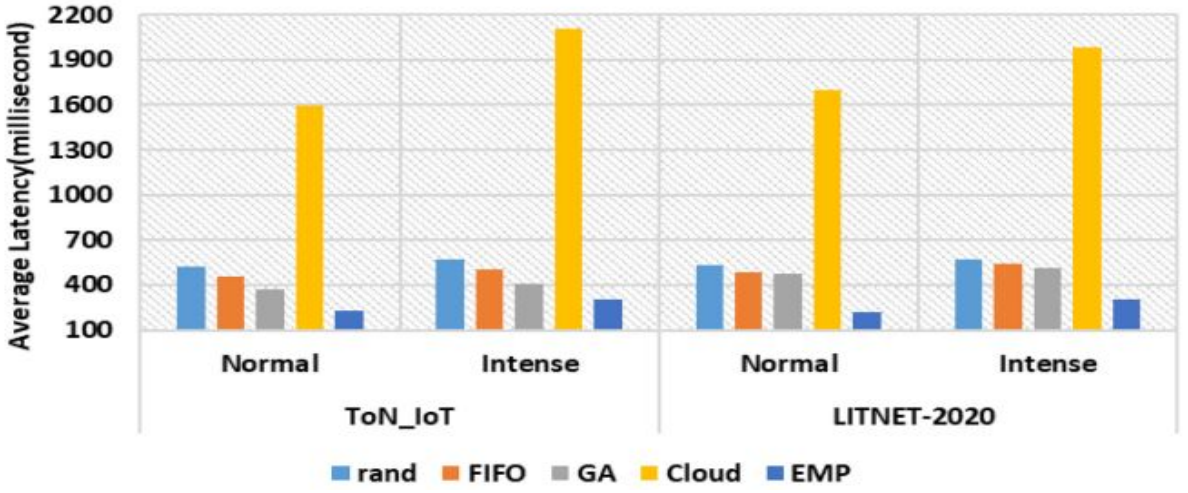

Fig. 8 Comparison of average latency performances of different placement methods

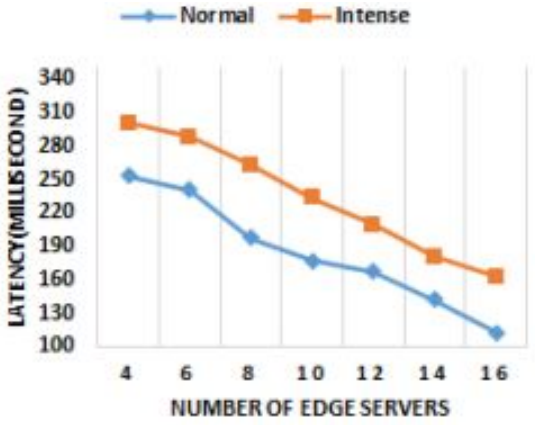

Fig. 9 Average latency value of different number of ESs under normal and intense flows obtained from ToN_IoT,

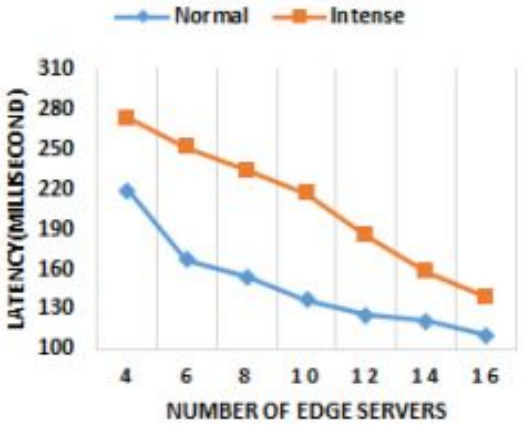

Fig. 12 Average latency value of different number of ESs under normal and intense flows obtained from LITNET-2020 data

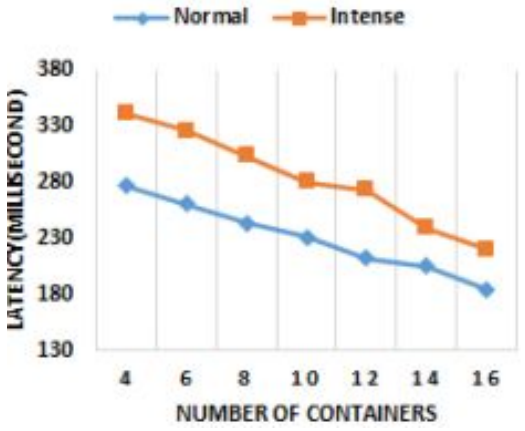

Fig. 10 Average latency value of different number of containers under normal and intense flows obtained from ToN_IoT

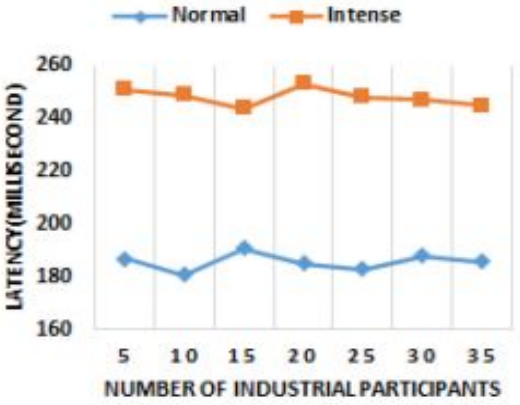

Fig. 11 Average latency value of different number of IPs under normal and intense flows obtained from ToN_IoT

7. Advantages and Disadvantages of proposed Federated Threat-Hunting approach

- Advantages
  I. Use of Proposed model for complex and heterogeneous data.
  II. It improves the real-time performance of threat detection in an ICPS as the first attempt to consider deploying a federated model as a threat-hunting microservice.
  III. This is the first study to introduce a differentially private FL model for detecting cyber attacks in containerized heterogeneous edge computing devices.

- Disadvantages
  I. Limitation on the amount of data heterogeneity due to widely available unlabeled data is still unable to be exploited during federated training.
  II. Detection performances might be negatively impacted due to differential privacy.
  III. malicious or improper ones can cause catastrophic consequences since the proposed system presumes that all the participants are trusted and suitable to engage in the training.

8. Conclusion

- This paper shows the efficiency of collaboration in the edge-cloud environment for Preserving the privacy of participants.
- For latency-effective placements a CIEC framework is introduced to integrate the Fed-TH in a microservice-based architecture.
- Extensive analysis reveal the accuracy and latency of the framework.
- Load balancing in multicloud IIoT environments can be explored for Future improvements.
- Severance deployment strategy for installing a single TI microservice on various ESs might be an expansion of this paper.

Thank you for your Attention

Bhagyashree Kakde
bhagyashreekakde27@gmail.com