IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2019, accepted December 23, 2019, date of publication December 30, 2019,
date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962873

Federated Learning-Based Cognitive Detection of Jamming Attack in Flying Ad-Hoc Network

NISHAT I. MOWLA[1], (Student Member, IEEE), NGUYEN H. TRAN[2], (Senior Member, IEEE), INSHIL DOH[3], AND KIJOON CHAE[1]

[1]Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea
[2]School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
[3]Department of Cyber Security, Ewha Womans University, Seoul 03760, South Korea

Corresponding author: Kijoon Chae (kjchae@ewha.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIP) under Grant 2019R1F1A1063194.

ABSTRACT Flying Ad-hoc Network (FANET) is a decentralized communication system solely formed by Unmanned Aerial Vehicles (UAVs). In FANET, the UAV clients are vulnerable to various malicious attacks such as the jamming attack. The aerial adversaries in the jamming attack disrupt the communication of the victim network through interference on the receiver side. Jamming attack detection in FANET poses new challenges for its key differences from other ad-hoc networks. First, because of the varying communication range and power consumption constraints, any centralized detection system becomes trivial in FANET. Second, the existing decentralized solutions, disregarding the unbalanced sensory data from new spatial environments, are unsuitable for the highly mobile and spatially heterogeneous UAVs in FANET. Third, given a huge number of UAV clients, the global model may need to choose a sub-group of UAV clients for providing a timely global update. Recently, federated learning has gained attention, as it addresses unbalanced data properties besides providing communication efficiency, thus making it a suitable choice for FANET. Therefore, we propose a federated learning-based on-device jamming attack detection security architecture for FANET. We enhance the proposed federated learning model with a client group prioritization technique leveraging the Dempster–Shafer theory. The proposed client group prioritization mechanism allows the aggregator node to identify better client groups for calculating the global update. We evaluated our mechanism with datasets from publicly available standardized jamming attack scenarios by CRAWDAD and the ns-3 simulated FANET architecture and showed that, in terms of accuracy, our proposed solution (82.01% for the CRAWDAD dataset and 89.73% for the ns-3 simulated FANET dataset) outperforms the traditional distributed solution (49.11% for the CRAWDAD dataset and 65.62% for the ns-3 simulated FANET dataset). Moreover, the Dempster–Shafer-based client group prioritization mechanism identifies the best client groups out of 56 client group combinations for efficient federated averaging.

INDEX TERMS Unmannad aerial vehicle, flying ad-hoc network, jamming attack, federated learning, on-device AI, Dempster–Shafer theory.

I. INTRODUCTION

The deployment of a group of Unmanned Aerial Vehicles (UAVs) is on the rise as they help to perform dangerous, dull, dirty and dumb tasks. Among the many design challenges for multi-UAV systems, communication remains crucial to provide cooperation and collaboration between the UAVs. However, the infrastructure-based communication architecture is not suitable for multi-UAV systems as UAV clients are highly mobile and undergoes rapid topology changes [1]. Thus, a Flying Ad-hoc Network (FANET) is considered to be the most efficient solution to address the challenges associated with a fully infrastructure-based UAV network [3]. The UAV clients in FANET transfer data to the base station independently within the communication range and infrastructure [1], as shown in Fig. 1.

Although FANET is an ad-hoc network, it has some distinct properties that strongly differentiate it from the

The associate editor coordinating the review of this manuscript and approving it for publication was Fangfei Li.

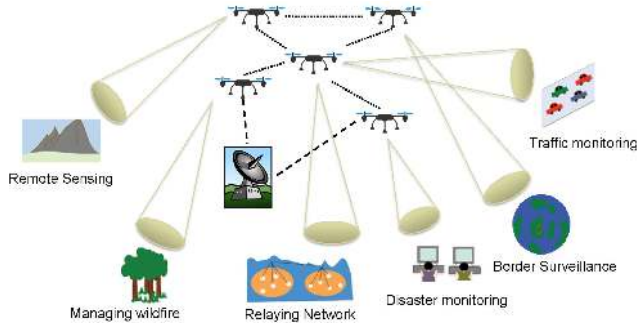


FIGURE 1. Flying Ad-hoc Network (FANET).

previously implemented ad-hoc networks such as Mobile Ad-hoc Network (MANET) and Vehicular Ad-hoc Network (VANET). For example, UAVs in the FANET can have rapid topology change and limited communication, unlike MANET or VANET. However, the computational capability of FANET is much higher than that of MANET and VANET [1], [2]. As the mode of communication in a FANET is wireless, the UAV nodes are vulnerable to malicious attacks by the aerial intruders. For example, aerial adversaries can cause a jamming attack in which the objective of the intruder is to disrupt the communication of the victim network intentionally by causing an interference or a collision on the receiver side [1], [4]. The jamming attack has been considered in previous research on various wireless networks [9]–[23]. In recent years, various solutions have also been proposed for the jamming attack in the UAVs [9], [13], [15]–[17], [20]. However, the proposed approaches of jamming attack detection for the UAVs are not well suited for FANET as jamming attack detection in the FANET faces three major challenges:

- First, it is to be noted that UAVs have a very high mobility and low density of nodes, unlike other networks, because of which, it is not always feasible to communicate with any centralized controller. This property of the nodes in FANET, thus, makes it more vulnerable to various attacks [1], [2]. In particular, there will be constraints for communicating to any centralized system whereas the UAV clients may require an *immediate response*. Furthermore, communications to any centralized system each time can cause severe *power consumption* [3]. Additionally, in centralized learning huge amounts of sensory data from the local devices need to be sent to the centralized controller which overlooks the critical *data privacy* issues of the local devices [26].
- Second, because of the high mobility [3], [4] of the UAV clients spanning very different spatial zones, the sensory data experienced by the UAVs can be very unbalanced. In this case, the traditional distributed solution to jamming attack detection may not be appropriate to learn these unbalanced sensory data resulting from the unprecedented environment of the specific UAV clients.
- Third, the global node will receive updates from a huge number of UAV clients. However, processing all the client updates will become infeasible at a certain point. Hence, the global node will need to select a sub-group of UAV clients to provide the global update timely. Moreover, the individual UAV clients will have varying contributions to the global model due to their individual feature collection. Therefore, selecting a representative sub-group of UAV clients can be additionally challenging.

Recently, on-device Artificial Intelligence (AI) has attracted considerable attention because of its communication efficiency [24], [25]. Solutions based on the federated averaging technique also address the unbalanced and non-IID data issues [25], [26]. Moreover, federated learning [25] is specially designed for device-level training for the lightweight devices (e.g., smart phones, UAVs, smart glass, and smart watches) [25]–[28]. With the help of federated learning, a neural network model can be trained locally (i.e., on-device) in any device by the help of a global model's weight updates. Moreover, federated learning allows low-level weight updates from the local devices to be sent and received from the global model [24], [25]. This property of the federated learning can help in extracting the fine-grained properties of the jamming data instances to reduce the effect of the imbalance in the data faced by the UAVs in FANET. This makes it a perfect fit for the jamming attack detection problem in FANET. Therefore, under the above mentioned circumstances, in this paper, we propose the first-of-its-kind a federated learning-based jamming attack detection approach for FANET. Furthermore, we propose a mechanism to identify UAV client groups with higher contributions to the global model, given that each UAV node trains its local model in very different spatial environment. In particular, we associate belief and plausibility as the lower and upper bounds of the trust measure with each UAV client group using the Dempster–Shafer theory [29]. Hence, we propose a UAV client group prioritization technique to identify better UAV client groups for the federated averaging computation. As a result, efficient on-device jamming attack detection can be performed in FANET while maintaining a limited amount of communication with the centralized system via federated learning. In essence, the main contributions of this paper can be summarized as follows:

- First, we investigate and discern the key issues of the jamming attack detection problem in FANET. In particular, we identify the challenges of using the existing centralized and distributed solutions which do not consider the communication efficiency and the unbalanced properties of the UAV sensory data respectively. Furthermore, we address the need to identify the sub-group of UAV clients from a huge pool of UAV clients to provide the timely global update. This becomes additionally challenging as the UAV clients make varying contributions to the global model because of varying feature collection and associated training.
- Second, we propose a security architecture of FANET, leveraging federated learning for cognitive jamming

attack detection. Thus, the detection can be achieved on-device while considering the unbalanced sensory data properties of the learning environment. Essentially, the UAV clients communicate with the centralized controller, when in range, to collect and incorporate the global weight updates into their own local model. The global weight update helps the local training models, but the final model is created locally in the UAV clients. Hence, the jamming attack detection is done locally in the devices using the on-device AI mechanism of federated learning [25]. Moreover, in the case of our proposal based on federated learning, only weight updates are sent to the global model; and this effectively preserves the privacy of the local sensory data of the UAVs [25], [26].

- Third, we propose a Dempster–Shafer theory-based client group prioritization technique to be conducted at the global node. The client group prioritization technique allows the evaluation of the UAV clients' contribution to the global model, given that the UAV clients encounter very different feature environments. Hence, the efficiency of the global model can be improved by enabling it to choose a better client group from a pool of clients, given that it needs to provide a global update periodically.
- Finally, we simulate the proposed architecture and apply our method to the publicly available CRAWDAD jamming attack dataset and the ns-3 [45] simulated FANET dataset. The simulation results show that the proposed model achieves a higher performance gain (82.01% for the CRAWDAD dataset and 89.73.22% for the ns-3 simulated FANET dataset) in terms of the average accuracy than the distributed model (49.11% for the CRAWDAD dataset and 65.62% for the ns-3 simulated FANET dataset). We also compute the best client groups by the proposed Demster–Shafer theory-based client group prioritization technique for a total of 56 client groups.

The rest of this paper is organized as follows. Section II present an overview of the related works of distributed machine learning and jamming attack detection techniques. Section III introduces the system model of the proposed security architecture. Section IV presents the performance of our system model. Finally, Section V concludes the paper with some remarks and possible future directions.

II. LITERATURE REVIEW

In this section, we will discuss the related works on distributed machine learning prediction systems. Then, we will review the various jamming attack solution approaches.

A. DISTRIBUTED MACHINE LEARNING

In [5], a cluster/data center-based distributed learning setting is proposed by applying distributed training with the iterative averaging of the locally trained models. However, they do not consider the unbalanced and non-IID data, which are essential properties for federated learning. In [6], an algorithm focused on communication efficiency with distributed optimization was proposed. However, this algorithm requires the number of clients to be smaller than the number of examples per client. Moreover, the data needs to be IID, balanced with each node having an equal number of data points. In [7], asynchronous distributed forms of SGD are proposed, which require a prohibitive number of updates in the federated setting. A distributed one-shot averaging technique was also proposed, but in the worst case, the global model produced no better performance than training a model on a single client [25]. In [27], high-level design, challenges, possible extensions, and some future directions of the federated learning are discussed. In [28], a decentralized learning framework for heterogeneous clients is proposed for federated learning. The paper focuses on the challenges for clients with limited computational resources which require longer update time or clients operating under poor wireless channel conditions resulting in longer upload time. In [8], a federated learning-based optimization model design and analysis are proposed for the wireless network.

B. JAMMING ATTACK SOLUTION APPROACHES

Jamming attack detection has long been a popular problem in wireless network settings. Some UAV-based solutions have been proposed recently, but they are not focused on the attack scenario in the case of the FANET architecture.

1) RULE-BASED JAMMING ATTACK SOLUTIONS

In [9], a rule-based attack detection mechanism for UAV networks is proposed to detect jamming, GPS spoofing, and other lethal types of cyber-attacks, as a hierarchical detection and response system. One main problem of rule-based detection mechanisms is that a large number of rules may be required for each variation of the attacks which makes it a limiting constraint, particularly in the era of fast-changing network attack types. In [10], frequency hopping strategies are derived from the optimal decision rules. The model evaluates the individual decision profiles of all the nodes and finally selects a rule that yields the maximum throughput. Various intrusion detection systems for jamming attacks in the wireless sensor network and the vehicular ad-hoc network are also discussed in [11]–[13]. In [14], a survey of attack and defense strategies is discussed involving spectral, spatial retreat and the competition strategies. Spectral retreat or frequency hopping and spatial retreat are the mechanisms of on-demand change in frequency and location respectively [14], [15]. In contrast, competition strategies work by competing against the jammer via adjusting the transmission power in the lower layers. However, they are resource-intensive and do not scale well for resource-constrained networks.

2) GAME THEORY-BASED ANALYSIS AND STRATEGIES FOR JAMMING ATTACK

In [16], a game-theoretic analysis of an aerial jamming attack on a UAV communication network is proposed using

TABLE 1. Summary of notations.

Notation	Description
$k \in K$	k-th client of K clients
w	Local weight
n	Total number of global instances
n_k	Toal number of instances in the k-th client
(x_i, y_i)	i-th feature x and i-th label y at the local client
$b \in \beta$	b mini batches of β
$e \in E$	e-th epoch of the total number of E epochs
η	Learning rate
$t = 1, 2, ...$	Number of rounds
$\frac{\Omega}{(\Omega-c)!c!}$	Generated number of client groups with c number of clients in each group
w_{t+1}^k	Local weight of client k at time $t+1$
$\sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$	Federated average
q_j	j-th client group
$m(K)$	Mass of K clients
$acc_K(w)$	Accuracy of K clients with weight w
$bel(q_j)$	Dempster–Shafer belief score of client q_j
$\sum_{K\|K\subseteq q_j} m(K)$	Sum of all the masses of client K in client group q_j
$pl(q_j)$	Plausibility of client group q_j
$\sum_{K\|K\cap q_j \neq \emptyset} m(K)$	Sum of all masses of client K that intersect with client group q_j
q_j^*	Client group with the maximum trust

a pursuit-evasion game. In [17], a Bayesian game-theoretic methodology is proposed for lethal attacks in UAV-aided networks. While game theory allows the computation of the optimal strategy, it remains a reactive mechanism that does not enable the proactive detection of an on-going attack. In [18], a survey of game-theoretic approaches for security requirements and threats mitigation in wireless sensor networks is discussed. In [19], a reputation-based coalition game was proposed to mitigate smart insider jamming attacks in MANET.

3) MACHINE LEARNING-BASED JAMMING ATTACK DETECTION

In [21], an unsupervised machine learning technique with the clustering technique was proposed for jamming attack detection in a pair of radio frequency communicating vehicles. In [22], malicious traffic detection in VANET was shown by using a statistical network traffic analysis with data mining method as a hybrid jamming detector. In [23], adversarial deep learning for cognitive jamming attack detection in a UAV network was proposed. The objective was to prevent the attacker from building a reliable classifier by balancing the conflicting effects of deceiving the attacker and making correct transmission decisions. In [20], smart jamming detection in a UAV network was proposed using reinforcement learning. Although these solutions leveraged machine learning to detect the jamming attack with cognition, the techniques remain centralized which makes it challenging to be suitable for a communication constrained network such as the FANET.

III. SYSTEM MODEL FOR FEDERATED LEARNING-BASED JAMMING ATTACK DETECTION

The proposed federated learning-based jamming attack detection model has five main steps: *A.* Parameter Estimation, *B.* Local Execution and Upload, *C.* Global Model Averaging, *D.* Dempster–Shafer Client Group Prioritization, and *E.* Local Download as shown in Figure 2.

A. PARAMETER ESTIMATION

In the federated learning architecture for the UAVs, each UAV client of the FANET conducts parameter estimation leveraging local sensory networking features such as Received Signal Strength Indicator (RSSI) and Packet Delivery Rate (PDR). Assuming that D_k is the set of the data points of client k and $n_k = |D_k|$, we get pairs of features and labels, (x_i, y_i), $i = 1, \ldots, n_k$ for local n_k data points as a fraction of global n data points (lines 2–3 in Alg. 1). This allows leveraging local learning from all the k clients each with n_k data points. Moreover, a shared weight update from the local devices sent to the global model allows a federated averaging technique, as will be discussed in the next subsection.

B. LOCAL EXECUTION AND UPLOAD

The local UAV client, k trains a local model with a finite sum objective of the form,

$$\min_{w \in R^d} L_k(w) \tag{1}$$

In (1), $L_k(w)$ is a loss function that is to be minimized with respect to w. $L_k(w)$ for an individual learner training over n_k data points stands for,

$$L_k(w) = \frac{1}{n_k} \sum_{i \in D_k} f_i(w) \tag{2}$$

where,

$$f_i(w) = f(x_i, y_i; w) \tag{3}$$

In (3), $f_i(w)$ is a function with the i-th feature x_i associated with label y_i and weight w. The local UAV, k splits n_k data points into B sized batches. Moreover, for each local epoch, a local weight, $w \in R^d$ is updated. A batch b out of a set of batches $\mathcal{B}$, each with size B, is trained to update w and η is the learning rate set by the local client. $\Delta(w; b)$ is the gradient of the local objective function of client k and is used to update the weight, w, over all the local batches before uploading them to a Muti-access Edge Computing (MEC) Server (lines 4–8 in Alg. 1). The *ClientUAVUpdate*(k, w) of Alg. 1 consists of the first two steps of the system model, i.e., parameter execution and local execution and upload.

Unlike the conventional computationally intensive approaches (e.g., Convolutional Neural Network (CNN), capsule networks, and other deep neural networks) [34]–[37], we propose a learning system model where the training process depends on both the local model (i.e., UAVs in the FANET) and the global model (i.e., MEC server). Moreover, the local model can converge to a suitable solution

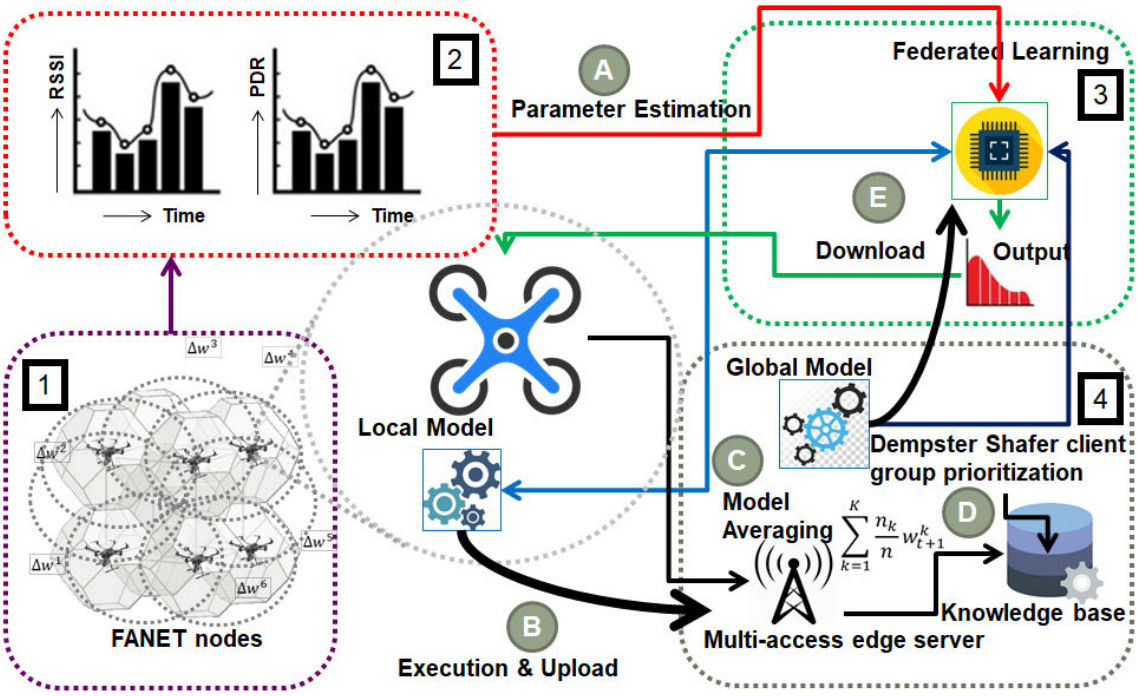


FIGURE 2. Proposed federated learning-based jamming attack detection mechanism.

relatively faster than that of the other distributed learning approaches [25], [26] with the aid of the global weight updates enabling fewer iterations for model training. As a result, the proposed hybrid trained model for jamming attack detection can be used for an efficient individual instance testing to perform on-device jamming attack detection.

C. GLOBAL MODEL AVERAGING

The global model aggregating node, i.e. the MEC server, has the global model objective,

$$\min_{w \in R^d} l(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \tag{4}$$

for n global data points. Because of the federated averaging, global $l(w)$ becomes,

$$l(w) = \sum_{k=1}^{K} \frac{n_k}{n} L_k(w) \tag{5}$$

In (5), $L_k(w)$ denotes the objective function of UAV client k. This yields a weighted average from all of the k UAV clients as the n_k data points will vary among the K clients. The global model of the security architecture initializes the weight and for each t round for each client k, w_{t+1}^k is updated by the *ClientUAVUpdate*(k, w) (lines 15–16 in Alg. 1). The local client updates w_{t+1}^k received at the MEC server are used to improvise the global model by using the federated averaging method by the weighted averaging of the aggregated client updates,

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{6}$$

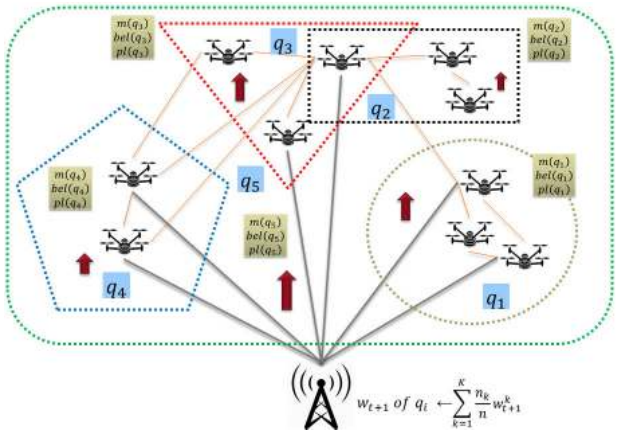


FIGURE 3. Dempster–Shafer-based client group prioritization.

In (6), w_{t+1} is the global weight at time $t+1$ for a total of K clients over a total of n data points (line 17 in Alg. 1).

D. DEMPSTER–SHAFER CLIENT GROUP PRIORITIZATION

The federated averaging technique considers all the UAV clients as one client group, (q_5) as shown in Fig. 3. However, in the FANET space, the clients may fall into different spatial zones encountering different feature accesses, thus affecting the weight that they calculate. For example, the client groups can be considered as a group of UAVs positioning in the local zones, q_1, q_2, q_3, q_4, and q_5, as shown in Fig. 3. Thus, some UAV clients in a specific zone may perform better than the UAVs in the other zones. Hence, the impact on the global model, computed from the local weight updates of the UAVs

Algorithm 1 Cognitive Jamming Attack Detection in FANET

1 **ClientUAVUpdate**(k, w)**:**
2 Pre-process n_k
3 Extract n_k feature set (x_i, y_i)
4 $b \leftarrow$ split data n_k into batches of size B
5 **for** *each local epoch e from 1 to E* **do**
6 | **for** $b \in \mathcal{B}$ **do**
7 | | $w \leftarrow w - \eta\Delta(w; b)$
8 return w to server
9 **BackboneUAVExecution:**
10 Initialize w_0
11 **for** *each round t = 1,2,...* **do**
12 | $c \leftarrow$ desired number of clients for client group q_j
13 | $j \leftarrow \frac{\Omega}{(\Omega-c)!c!}$
14 | $q_j \leftarrow j$-th client group
15 | **for** *each client* $k \in q_j$ **do**
16 | | $w^k_{t+1} \leftarrow$ ClientUAVUpdate(k, w)
17 | | $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1}$
18 | | $w \leftarrow w_{t+1}$
19 | **for** *each client group* $q_j \in \Omega$ **do**
20 | | $m(K) \leftarrow acc_K(w)$
21 | | $bel(q_j) = \sum_{K|K\subseteq q_j} m(K)$
22 | | $pl(q_j) = \sum_{K|K\cap q_j \neq \emptyset} m(K)$
23 | | Select the minimum value parameter between $bel(q_j)$ and $pl(q_j)$
24 | | Update q^*_j over q_j with highest $bel(q_j)$ or $pl(q_j)$ parameter value
25 | return w corresponding to q^*_j

in these zones, will also vary. Moreover, in the real-world applications, it is not always feasible to perform federated averaging over all the local client updates, mainly because of the bandwidth constraints and latency issues for the global update [27], [28]. In this regard, a client group prioritization can help to select a sub-group for the global model averaging. Essentially, the client group prioritization enables selecting a UAV client group that is contributing the most to improve the global weight update and use this sub-group to calculate the global update. Furthermore, it can indicate which clients are contributing the most to the global model averaging. We generate the total number of client groups, q_j, $j = 1, 2, 3..., j$ by computing the combination $\frac{\Omega}{(\Omega-c)!c!}$, where Ω is the total number of clients present and c is the number of desired clients forming a client group that the global aggregator will use to calculate the federated average, given $c \leq K$ (lines 12–14 in Alg. 1). The global weight updates for each client group q_j are saved in the knowledge base accordingly.

In order to perform the client group prioritization, we applied the Dempster–Shafer Theory [29] to derive the lower bound and the upper bound of the trust for the client group, q_j. At the global aggregator, q_j yields j-th global weight update from which we can derive a global average accuracy measure.

The Dempster–Shafer theory is a generalization of the Bayesian theory of subjective probability where the latter requires the probabilities for each question of interest, but belief functions are based on degrees of belief (i.e., trust) for one question on the probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities and how much they differ depends on how closely the two questions are related. In other words, it is a way of representing epistemic plausibilities but can yield answers that contradict those arrived at by using the probability theory [29]. As a result, the evidence in the Dempster–Shafer theory can be meaningful at a higher level of abstraction without having to resort to assumptions about the events within the evidential set. A key property of the Dempster–Shafer theory is that the model is designed to cope with varying levels of precision regarding the information and no further assumptions are needed to represent the information. The three main functions in the Dempster–Shafer theory are the basic probability assignment function or mass, the belief function and, the plausibility function. The building blocks of the Dempster–Shafer theory are associated with the basic probability assignment function called mass. The basic probability function, mass, defines the mapping of the power set to the interval between 0 and 1, where the mass of the null set is 0 and the summation of the masses of all the subsets of the power set is 1. The value of the mass for a given set K is represented as $m(K)$), expresses the proportion of all the relevant and available evidence that supports the claim that a particular element of Ω (the universal set) belongs to set A. In other words, given a domain Ω, a probability mass is assigned to each subset of Ω, as opposed to each element, as in the classical probability theory. Such an assignment is called the Basic Probability Assignment (BPA). A mass function on Ω is a function $m : 2^{\Omega} \leftarrow [0, 1]$ such that the following two conditions hold,

$$m(\varnothing) = 0 \tag{7}$$

$$\sum_{K\subseteq\Omega} m(K) = 1 \tag{8}$$

In (7), $\varnothing$ represents the null set and its mass is 0. In (8), $m(K)$ represents the mass of the subset K and the sum of all the subsets in the power set is 1. As such, the mass cannot be equated with a classical probability in general [30], [32].

For example, by assigning a probability mass to a subset in a BPA, we associate a level of trust in the subset but we cannot be any more specific. For instance, in the case of a wireless network of FANET consisted of six clients, $(\Omega = 1, 2, 3, 4, 5, 6)$, if we have no information, we have a BPA of $m(\Omega) = 1$. Because if there is no other information, there is a 100% certainty that one of the elements in Ω gives the correct output, but we cannot be more specific about which of these six clients will give the correct ouput. Now, if more information is given, for example, the subset of set Ω gives the correct output, where the subset is defined as 2, 4, 6

(i.e., composed of the second, fourth and the sixth client), the BPA would be $m(2, 4, 6) = 1$, but we still cannot distinguish between the performance of the second, fourth and the sixth clients. If BPA of $m(2, 4, 6) = 0.7$ and $m(1) = 0.3$, there is a 70% chance that the second, fourth and the sixth clients are giving the correct output and a 30% change that the first client is giving the correct output. The subsets of Ω that are assigned with a nonzero probability mass are called the focal elements of the BPA [31].

The functions and the combining rule of the Dempster–Shafer theory are well suited to associate certainty with the type of evidence and its aggregation as described above. Likewise, the combining rules of the Dempster–Shafer theory can be applied to the UAV clients of the wireless network of FANET when clients need to be aggregated and a subset needs to be selected with a higher chance of a better federated averaging computation. Here, the weight updates of the UAV clients resulting in the overall detection accuracy in FANET are equivalent to the evidence from clients which can then be aggregated by applying the Dempster–Shafer theory of evidence. The combination rule is independent of the order in which the evidence is gathered and requires that the hypotheses under consideration be mutually exclusive and exhaustive. From the BPA, the upper and lower bounds of an interval are defined with (9) and (10) respectively. This interval contains the precise probability of the set of interest (in the classical sense) and is bounded by two non-additive continuous measures called belief and plausibility [29], [30].

Definition 1: Let m be the mass on Ω. *Then for every client group (i.e, set)* $A \subseteq \Omega$, *the lower bound for a client group A is called the belief and is defined as the sum of all the basic probability assignments of the appropriate client groups (i.e., sets)* (B) *of the set of interest* (A).

Thus, for our client group q_j, $bel(q_j)$ is the sum of all the masses of clients belonging to a specific client group,

$$bel(q_j) = \sum_{K|K\subseteq q_j} m(K) \tag{9}$$

In (9), $m(K)$ is the basic probability assignment function of mass for K, $P(K) \rightarrow [0, 1]$.

Definition 2: Let m be the mass on Ω. *Then for every client group (i.e., set)* $A \subseteq \Omega$, *the lower bound for a client group A is called the plausibility and is defined as the sum of all the basic probability assignments of the client groups (i.e., sets)* (B) *that intersect the set of interest* (A).

Hence, in case of our client group q_j, plausibility is the sum of all the masses of the K clients that intersect the other q_j client groups as shown in (10),

$$pl(q_j) = \sum_{K|K\cap q_j\neq\emptyset} m(K) \tag{10}$$

Using these bounds, trust is associated with each client group q_j as a measure of belief and plausibility score. Hence, for our client group prioritization, a client group q_j with a higher belief and plausibility score is given more priority than a client group with a lower belief and plausibility score. The two measures, belief and plausibility are non-additive. This can be interpreted as is not required for the sum of all the belief measures to be 1 and similarly for the sum of the plausibility measures. It is possible to obtain the basic probability assignment function of mass from the belief measure with the following inverse function,

$$m(q_j) = \sum_{K|K\subseteq q_j} (-1)^{|q_j-K|} bel(K) \tag{11}$$

In (11), $|q_j - K|$ is the difference of the cardinality of the two sets. In addition to deriving these measures from the basic probability assignment function of mass (m), the belief and plausibility can be derived from each other. For example, plausibility can be derived from belief as,

$$pl(q_j) = 1 - bel(\bar{q_j}) \tag{12}$$

where $\bar{q_j}$ is the classical complement of q_j. This definition of plausibility in terms of belief comes from the fact that all the basic assignments must sum to 1. From the definitions of belief and plausibility, it follows that $pl(q_j) = 1-bel(\bar{q_j})$. As a consequence, it is possible to derive the values of the other two measures if any of these measures $(m(A), bel(A), pl(A))$ is given. The precise probability of an event (in the classical sense) lies within the lower and upper bounds of belief and plausibility, respectively. Thus, for our j-th client group q_j we get a measure of,

$$bel(q_j) \leq P(q_j) \leq pl(q_j) \tag{13}$$

where, $bel(q_j)$ is the belief, $P(q_j)$ is the probability of q_j and $pl(q_j)$ is the plausibility of client group q_j. The probability, $P(q_j)$, is uniquely determined if $bel(q_j) = pl(q_j)$. Otherwise, $bel(q_j)$ and $pl(q_j)$ may be viewed as the lower and upper bounds on the probabilities, respectively, where the actual probability is contained in the interval described by the bounds. For instance, in case of the wireless network of FANET, a client group q_j has an accuracy of 0.8 which means that it has a probability of 0.2 of being wrong in 20% of the cases. For an output set $O = \{$jammer, non-jammer$\}$, accuracy set $A = \{+acc, -acc\}$, and $m(+acc) = 0.8$, $m(-acc) = 0.2$, if the output by q_j is predicted to be jammer then,

$$bel(q_j) = \sum_{K|K\subseteq q_j} m(K) = 0.8 \tag{14}$$

and,

$$pl(q_j) = \sum_{K|K\cap q_j\neq\emptyset} m(K) = 1 \tag{15}$$

For the client group prioritization, the global model update for each client group initializes the BPA function of mass with the corresponding accuracy achieved (lines 19–20 in Alg. 1) as the accuracy can be expressed as a value between 0 and 1. This is used to calculate the belief and plausibility for each client group (lines 21–22 in Alg. 1) as discussed above. The minimum parameter value between the belief $(bel(q_j))$ and the plausibility $(pl(q_j))$ is selected for the best q_j value (q_j^*)

evaluation. Then, q_j over all the q_j client groups with the highest parameter value (i.e., $bel(q_j)$ or $pl(q_j)$) is selected as the q_j^*. Finally, the weight update of the selected q_j^* client group is returned to the local models (lines 23–25 in Alg. 1). The maximum parameter value between $bel(q_j)$ and $pl(q_j)$ for each client group can also be selected as the parameter for the q_j^* evaluation, as a second test, but mostly we used the minimum parameter value between $bel(q_j)$ and $pl(q_j)$ to consider the worst case.

The client group prioritization is practical, particularly, if the global node receives a huge number of local client updates. In such a case, the global aggregator can process a smaller number of local client updates by applying the client group prioritization technique to select subgroup q_j with a higher belief and plausibility in order to provide a better global weight update. Moreover, the Dempster–Shafer theory based client group prioritization allows one to associate higher trust with UAV client groups (i.e., subsets of the total number of UAV clients) providing better weight updates for the unbalanced data resulting from spatial heterogeneity, as the basic probability assignment function of mass is derived from the accuracy achieved by the global federated averaging technique.

Note that the solution method is based on the combinations of individual UAVs where the preference lists of the UAVs are mainly instantaneous and dependent on the received individual UAV performance. Moreover, the preference list sustains until the global model average accuracy requirements are not violated. In other words, the UAVs will be removed from the client group as the individual UAV performance degrades because of the individual sensory environment. In this regard, we solve the problem using the belief and plausibility measure [29], [30] of each individual UAV client and their combinations. Additionally, the prioritized client groups were maintained for a fixed amount of time. The reason behind this approach was to ensure the stability of the global averaging model. Therefore, the prioritization also changes after a fixed amount of time. In practice, the client group prioritization is done on the basis of the combination of UAVs in the range during the global averaging calculation and any UAV client updates that arrive in the range during the global averaging are kept at the backlog of the global node (i.e., MEC server).

E. LOCAL DOWNLOAD

The federated update from the global node is forwarded by the MEC server and downloaded by the local UAV clients to train their local models again (line 25 in Alg. 1). Thus, the downloaded federated update aids the local clients to improve the performance of their own local models. The updated local models can then be used by the UAV clients to perform on-device jamming attack detection. The last three steps of the system model, namely global model averaging, Dempster–Shafer client group prioritization and local download, construct the *BackboneUAVExecution* of Algorithm 1. Table 1 summarizes the notations used for the federated cognitive detection of the jamming attack in FANET. In the next section, we extensively evaluate the proposed mechanism and verify the benefits defined.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

A. EXPERIMENTAL SETTINGS

For our experimentation, we used a 64-bit, Intel i7-47900 CPU @ 3.60GHz processor and 8.00-GB RAM simulation environment. We used the standard public datasets of the jamming attack in a vehicular ad-hoc network [33] and pre-processed them to extract 3000 instances with 100 features each associated with Received Signal Strength Indicator (RSSI) and Packet Delivery Rate (PDR). The communication technologies that we consider for the dataset is wireless and therefore, is also applicable to FANET. Additionally, the features used are RSSI and PDR as in the case of FANET for interpreting the jamming scenarios. The dataset contains traces of the 802.11p packets collected in a rural area located in the periphery of Aachen (Germany) in 2012, with the presence of a Radio Frequency (RF) jamming signal with constant, periodic, and reactive jamming patterns. Moreover, according to the Federated Aviation Administration, Fact Sheet on Unmanned Aircraft Regulations [38], [39], the UAVs usually maintain a fixed altitude because of which the third dimension of the UAVs in the FANET can be considered fixed. Therefore, for the *proof-of-concept* for jamming detection in FANET, given that no other publicly available standard UAV jamming attack datasets are currently available, the jamming attack dataset in the vehicular ad-hoc network performs as the closest fit for the numerical analysis. However, note that the dataset is pre-processed to create a pathologically unbalanced dataset [25] by generating an imbalanced proportion of classes to particularly address the unbalanced sensory environment of the UAVs in FANET.

Apart from this, for further verification of our proposal, we also simulated an ns-3 [45] based Flying Ad-hoc Network topology with three-dimensional mobility model in an ad-hoc setting, communicating over the WiFi physical standard 802.11n [40]- [42]. A jammer node was introduced with constant, periodic and reactive RF jamming signals which interferes with the communication between three UAV nodes and a server node, in the three-dimensional UAV ad-hoc Gauss Markov mobility model [43], [44]. We extract 3000 instances with 8 features each consisting of PDR, RSSI, and Throughput.

By definition, an unbalanced dataset [25], [26] is a dataset where the number of instances available for an associated class is considerably smaller than the number of instances available for another associated class. This imbalance can affect the performance of the distributed learning if the fine-grained properties cannot be extracted. Thankfully, federated learning allows low-level weight updates from the local devices to be sent and received from the global model, which can help in extracting fine-grained properties of the instances to reduce the effect of the imbalance in the dataset.

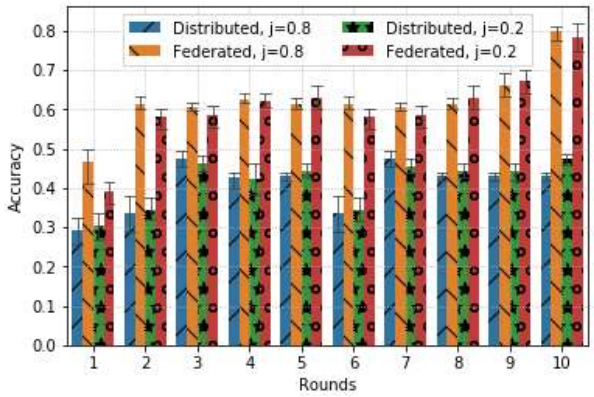


FIGURE 4. Distributed learning vs. federated learning for jamming attack detection in CRAWDAD vehicular ad-hoc network dataset.

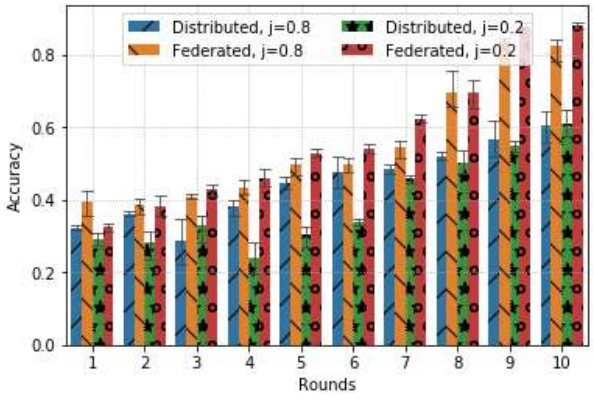


FIGURE 5. Distributed learning vs. federated learning for jamming attack detection in ns-3 simulated flying Ad-hoc network dataset.

Hence, we pre-process both the datasets to derive two unbalanced sub-datasets, which enable the comparison analysis between the proposed approach and the baseline approach in a more realistic scenario. The first sub-dataset contains a higher percentage of jamming instances (80%) and a lower percentage of non-jamming instances (20%). The second sub-dataset contains a lower percentage of jamming instances (20%) and a higher percentage of non-jamming instances (80%). For both the sub-datasets, a 50 : 50 ratio is maintained for the training and testing sets. As the jamming attack problem considered in this study is a two-class problem and the testing of the non-IID performance of the data requires more than two classes [25], the non-IID data problem does not apply to our specific scenario. Hence, the focus of this study was on the unbalanced sensory data collected by the UAVs, which we trained with the proposed federated learning model validated by using cross-validation. In our future work, we will consider incorporating some non-IID data scenarios for jamming detection in the FANET architecture.

A three-layer neural network model was simulated in Python 3.6.5 generated with one flatten layer (i.e., converting input features into a vector) and two fully connected dense layers with Rectified Linear Unit (ReLU) and softmax functions respectively. Six client instances and one MEC server instance were generated. The training data were distributed equally between the clients. Ten rounds of communication were used to update the global model by using the weight updates from the local models running on the six clients.

B. SIMULATION RESULT OF PROPOSED FEDERATED JAMMING ATTACK DETECTION MODEL

We present a comparison of the proposed federated learning model with the distributed model, which leverages a central server for averaging the updates of the average gradient collected from the local clients. In the distributed model, the local clients pull the newly updated global gradient from the centralized server to train their local model again.

Fig. 4 shows the results obtained over 10 rounds of communication for the distributed and federated learning models in the CRAWDAD dataset after 25 epochs. As shown in Fig. 4, Distributed, $j = 0.8$ and Federated, $j = 0.8$ indicate an unbalanced dataset of 80% jamming data instances and 20% non-jamming data instances for the distributed and the federated learning model, respectively. Alternatively, Distributed, $j = 0.2$ and Federated, $j = 0.2$ indicate an unbalanced dataset of 20% jamming data instances and 80% non-jamming data instances for the distributed and the federated learning model, respectively. As can be seen from Fig. 4, the distributed learner is significantly outperformed by the proposed federated learning model for both the 80% and the 20% jamming data instances in the unbalanced jamming attack dataset. In terms of accuracy, the distributed model's performance with 80% jamming instances remained around 0.3259 and 0.4375. In contrast, the federated learning model's performance with 80% jamming instances increases from 0.5 to 0.8101 over the 10 rounds. The distributed model's performance with 80% non-jamming instances remains around 0.3125 and 0.4911. In contrast, the federated learning model's performance with 80% jamming instances increased from 0.4259 to 0.8201 over the 10 rounds. The measurement analysis clearly indicates the effectiveness of the proposed approach over the baseline approach. In fact, the gap in performance between the two models, namely the distributed and federated learning models, increases with an increase in the number of rounds.

Fig. 5 shows the results obtained over the 10 rounds of communication for the distributed and federated learning models for the ns-3 simulated Flying Ad-hoc Network dataset after 25 epochs. As can be seen from Fig. 5, for the unbalanced jamming attack dataset of the ns-3 simulated FANET, the distributed learning model is again significantly outperformed by the proposed federated learning model for both the 80% and 20% jamming data instances. In terms of accuracy, the distributed learning model's performance with 80% jamming instances remained around 0.3362 and 0.6562. In contrast, the federated learning model's performance with 80% jamming instances increased from 0.433 to 0.8438 over the 10 rounds. The distributed learning model's performance

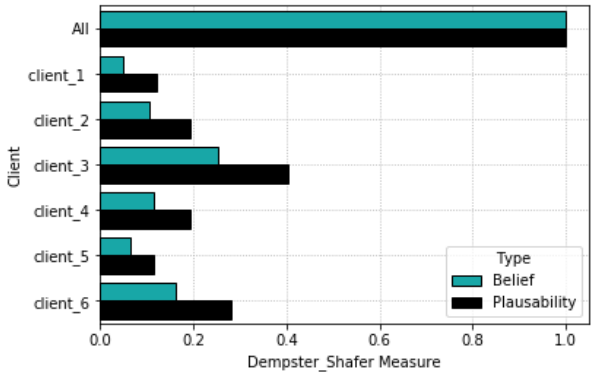
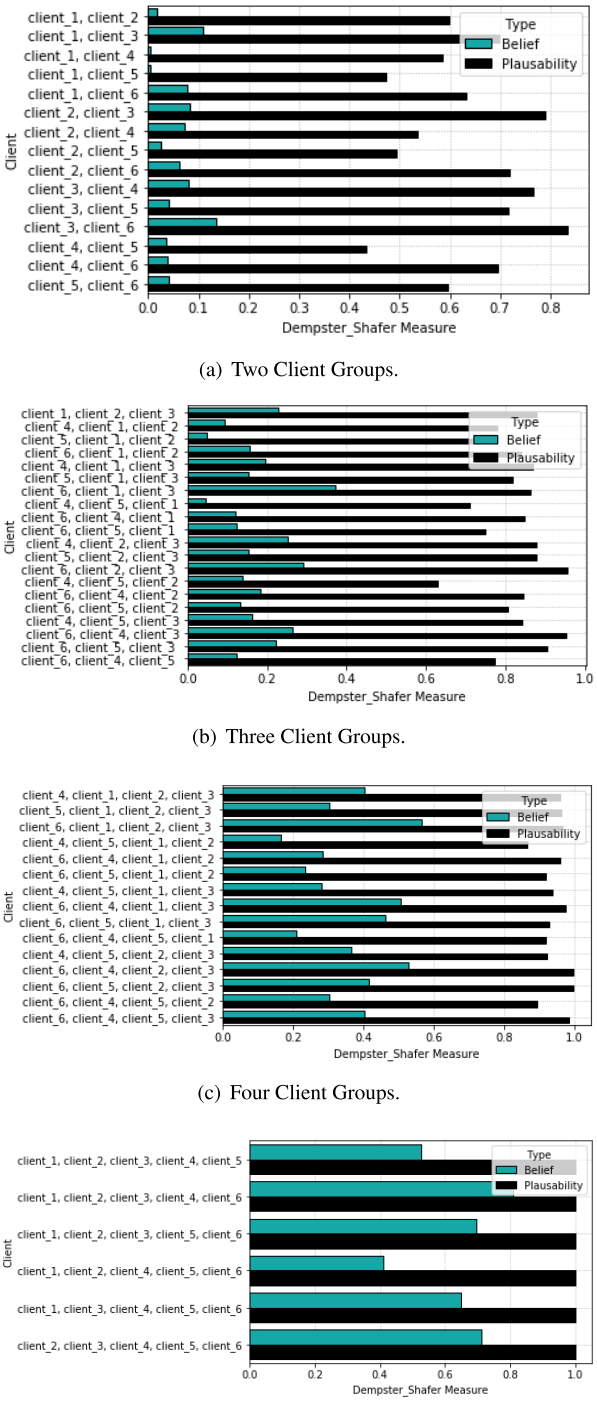


FIGURE 6. Dempster–Shafer measure for individual clients and all.

with 80% non-jamming instances remains around 0.3162 and 0.625. In contrast, the federated learning model's performance with 80% jamming instances increases from 0.3504 to 0.8973 over the 10 rounds. Similar to the previous case, the gap in performance between the two models, the distributed learning and federated learning model, increases with an increase in the number of rounds. The reason behind this phenomena is that the federated learning model's weights help to recognize each instance by fine-tuning the weight updates at a considerably lower level than a more general average gradient calculated from the distributed learning models. Furthermore, from both Fig. 4 and Fig. 5, it can be inferred that a federated learning model can detect jamming attack better than the traditional distributed learning model in an unbalanced data environment.

C. SIMULATION RESULT OF THE DEMPSTER–SHAFER BASED CLIENT GROUP PRIORITIZATION

In the global node of our federated learning architecture, the Dempster–Shafer algorithm is implemented to evaluate the client groups' updates, unlike the traditional federated learning model. The masses of the six clients are associated with the lower average accuracy from the two datasets attained by each client to consider the worst case scenario. Fig. 6 depicts the belief and the plausibility achieved for all the clients together and for each of the clients individually. The results allowed us to confirm two crucial facts in our experimental scenario. First, each of the six clients has a different belief and plausibility associated with them. As a result, the performance of the client groups varied on the basis of which client falls into which group. Here, client 3 has higher belief and plausibility (0.254, 0.4027) bounds than that of the other individual clients, and client 6 has the second-highest belief and plausibility (0.1613, 0.2806) bounds. Client 1 has the lowest belief and plausibility bounds (0.0476, 0.1199). Second, all of the six clients achieve better Dempster–Shafer evidence bounds (i.e., 1, 1) than that of the individual clients alone which allow to verify and associate more trust with an aggregation model such as federated learning for jamming attack detection.



(a) Two Client Groups.

(b) Three Client Groups.

(c) Four Client Groups.

(d) Five Client Groups.

FIGURE 7. Dempster–Shafer based client group evaluation.

As discussed earlier, it is not always feasible to perform federated averaging over all the client local updates because of the latency and bandwidth constraints for the global update for real-world applications [27], [28]. Therefore, a subset of clients have to be selected that would provide a fair performance of the global model. Fig. 7 shows the Dempster–Shafer based client group evaluation in terms of

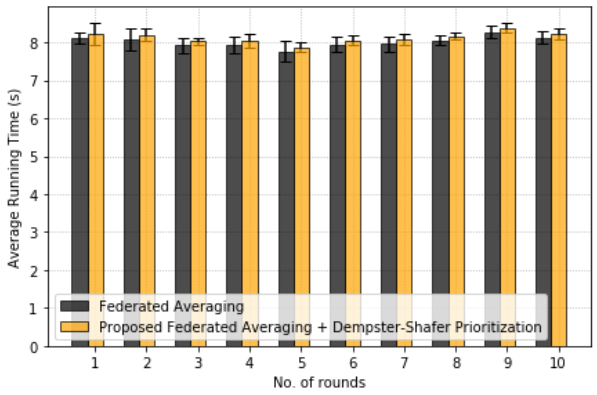


FIGURE 8. Average running time for federated averaging and proposed federated averaging with Dempster–Shafer client group prioritization.

belief and plausibility. The subfigures, Fig. 7a, Fig. 7b, Fig. 7c, and Fig. 7d illustrate the associated belief and plausibility of client groups with 2, 3, 4 and 5 clients comprised of 15, 20, 15 and 6 possible client group combinations, respectively, resulting in a total of 56 client group combinations. The belief and plausibility scores for the client group are obtained by applying the algorithm of the Dempster–Shafer theory on the average accuracy achieved by the client group combinations, as discussed earlier. Among two client groups, the client group composed of client 3 and client 6 achieves the highest belief and plausibility bounds (0.1351, 0.8352). Among three client groups, the client group composed of client 2, client 3, and client 6 achieves the highest belief and plausibility bounds (0.2905, 0.9547). Among four client groups, the client group composed of client 2, client 3, client 4 and client 6 achieves the highest belief and plausibility bounds (0.5268, 0.9957). Among five client groups, the client group composed of client 1, client 2, client 3, client 4 and client 6 achieves the highest belief and plausibility bounds (0.8089, 1). It can be observed that in all of the combinations of the four cases of client group combinations, client groups having client 3 and client 6 perform better than the other client groups. This could be attributed either to the fact that client 3 and client 6 encountered more representative features, i.e., sensory information such as the RSSI and PDR collected by the UAV client or to a better local learning. Thus, the proposed Dempster–Shafer-based client group prioritization technique enables the selection of a specific client group, given that a constrained number of clients need to be selected to calculate the federated average of the federated learning model on a timely basis.

D. SIMULATION RESULT OF AVERAGE RUNNING TIME COMPARISON

In order to confirm that the Dempster–Shafer client group prioritization doesn't incur considerable delay for the overall federated learning global model update, we evaluated the average running time for the models. Fig. 8 shows the performance of the average running time of the federated averaging model and the Dempster–Shafer implementation for the three client groups (i.e the client groups with the largest number of combinations) over 10 rounds of communication. The average running time of the federated averaging algorithm for the jamming attack detection in the federated learning model is 8.0191s. In contrast, the average running time of the proposed federated averaging with the Dempster–Shafer client group prioritization is 8.0303s. This shows that the added Dempster-Shafer client group prioritization incurs very little added running time (0.02 seconds on average) to the federated averaging technique, while enabling a subset of the client groups to be selected to perform the scaled federated learning when all of the clients cannot be accommodated.

V. CONCLUSION

In this paper, we proposed a federated learning-based security architecture for the on-device detection of jamming attack in FANET. For doing so, we integrate an on-device jamming attack detection federated learning model devised in the UAV clients of the FANET. We also proposed a client group prioritization technique using the Dempster–Shafer theory to perform client group selection for a constrained global model. We showed that the proposed mechanism could achieve promising performance in jamming attack detection with the proposed federated learning approach, given that there is unbalanced sensory data. We also showed that the federated averaging mechanism of the federated learning model could be further scaled to select better client groups with the Dempster-Shafer based client group prioritization technique. In the future, we will consider a decentralized global model to improve the reliability of the global updates in the UAV-FANET architecture.

REFERENCES

[1] İ. Bekmezci, O. K. Sahingoz, and . Temel, "Flying ad–hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013.
[2] A. Guillen-Perez and M.-D. Cano, "Flying ad hoc networks: A new domain for network communications," *Sensors*, vol. 18, no. 10, p. 3571, Oct. 2018.
[3] O. K. Sahingoz, "Networking models in flying ad–hoc networks (FANETs): Concepts and challenges," *J. Intell. Robot. Syst.*, vol. 74, nos. 1–2, pp. 513–527, Apr. 2014.
[4] I. Bekmezci, E. Senturk, and T. Turker, "Security issues in flying ad-hoc networks (FANETS)," *J. Aeronaut. Space Technol.*, vol. 9, no. 2, pp. 13–21, 2016.
[5] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 685–693.
[6] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2328–2336.
[7] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, and A. Senior, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1223–1231.
[8] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[9] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal Cyber–attacks in UAV networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1594–1606, Sep. 2018.

[10] P. Bhavathankar, S. Sarkar, and S. Misra, "Optimal decision rule-based ex-ante frequency hopping for jamming avoidance in wireless sensor networks," *Comput. Netw.*, vol. 128, pp. 172–185, Dec. 2017.

[11] O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *Proc. 6th Int. Conf. Modeling Simulation Appl. Optim. (ICMSAO)*, May 2015, pp. 1–6.

[12] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure Internet of Things," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 84–90.

[13] S. Tanwar, J. Vora, S. Tyagi, N. Kumar, and M. S. Obaidat, "A systematic review on security issues in vehicular ad hoc network," *Secur. Privacy*, vol. 1, no. 5, p. e39, Sep. 2018.

[14] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," *IEEE Netw.*, vol. 20, no. 3, pp. 41–47, May 2006.

[15] P. Cevik, I. Kocaman, A. S. Akgul, and B. Akca, "The small and silent force multiplier: A swarm UAV—Electronic attack," *J. Intell. Robotic Syst.*, vol. 70, nos. 1–4, pp. 595–608, 2013.

[16] S. Bhattacharya and T. Basar, "Game-theoretic analysis of an aerial jamming attack on a UAV communication network," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 818–823.

[17] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Intrusion detection and ejection framework against lethal attacks in UAV–aided networks: A Bayesian game–theoretic methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1143–1153, May 2017.

[18] M. Abdalzaher, K. Seddik, M. Elsabrouty, O. Muta, H. Furukawa, and A. Abdel-Rahman, "Game theory meets wireless sensor networks security requirements and threats mitigation: A survey," *Sensors*, vol. 16, no. 7, p. 1003, Jun. 2016.

[19] T. Oyedare, A. A. Sharah, and S. Shetty, "A reputation-based coalition game to prevent smart insider jamming attacks in MANETs," in *Proc. Int. Conf. Wired/Wireless Internet Commun.* Cham, Switzerland, Springer, 2016, pp. 241–253.

[20] L. Xiao, W. Zhuang, S. Zhou, and C. Chen, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[21] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning," *Veh. Commun.*, vol. 13, pp. 56–63, Jul. 2018.

[22] N. Lyamin, D. Kleyko, Q. Delooz, and A. Vinel, "AI–based malicious network traffic detection in VANETs," *IEEE Netw.*, vol. 32, no. 6, pp. 15–21, Nov. 2018.

[23] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2018, pp. 1–6.

[24] N. I. Mowla, I. Doh, and K. Chae, "On–device ai–based cognitive detection of bio–modality spoofing in medical Cyber physical system," *IEEE Access*, vol. 7, pp. 2126–2137, 2019.

[25] H. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*. [Online]. Available: https://arxiv.org/abs/1602.05629

[26] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*. [Online]. Available: https://arxiv.org/abs/1610.05492

[27] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, and C. Kiddon, "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*. [Online]. Available: https://arxiv.org/abs/1902.01046

[28] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.(ICC)*, May 2019, pp. 1–7.

[29] K. Sentz and S. Ferson, "Combination of evidence in dempster Shafer theory," Sandia Nat. Laboratories, Albuquerque, CA, USA, Tech. Rep. 4015, 2002.

[30] J. Kohlas and P. Monney, *A Mathematical Theory of Hints: An Approach to the Dempster-Shafer Theory of Evidence*, vol. 425. Springer, 2013.

[31] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, and R. P. Goldman, Eds, "Plan, activity, and intent recognition: Theory and practice," in *Newnes*. 2014.

[32] Y. Lu and J. He, "Dempster-Shafer evidence theory and study of some key problems," in *Infinite Study*. 2017.

[33] O. Punal, C. Pereira, A. Aguiar, J. Gross. (May 2014). *CRAWDAD Dataset Uportorwthaachen/vanetjamming Downloaded From*. [Online]. Available: https://crawdad.org/uportorwthaachen/vanetjamming2014/20140512, doi: 10.15783/C7Q306.

[34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[36] K. Ovtcharov, O. Ruwase, J. Kim, J. Fowers, K. Strauss, and E. S. Chung, "Accelerating deep convolutional neural networks using specialized hardware," Microsoft Res., Washington, DC, USA, White Paper 211, 2015, pp. 1–4.

[37] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2016, pp. 354–370.

[38] *Federated Aviation Administration, Fact Sheet—Small Unmanned Aircraft Regulations (Part)*. [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615

[39] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A Tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart. 2019.

[40] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Heimlicher, "Characterizing 802.11 n aerial communication," in *Proc. 2nd MobiHoc Workshop Airborne Netw. Commun.*, 2013, pp. 7–12.

[41] S. Rosati, K. Kruzelecki, L. Traynard, and B. Rimoldi, "Speed-aware routing for UAV ad-hoc networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 1367–1373.

[42] N. Goddemeier, S. Rohde, and C. Wietfeld, "Experimental validation of RSS driven UAV mobility behaviors in IEEE 802.11 s networks," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1550–1555.

[43] D. Broyles, A. Jabbar, and J. P. G. Sterbenz, "Design and analyis of a 3-d Gauss-Markov mobility model for highly-dynamic air bourne networks," in *Proc. Int. Telemetering Conf.*, ITC, San Deigo, CA, USA. Oct. 2010.

[44] J. P. Rohrer, E. K. Cetinkaya, H. Narra, D. Broyles, K. Peters, and J. P. G. Sterbenz, "Aero RP performance in highly dynamic airborne networks using 3D Gauss Markov mobility model," in *Proc. MILCOM*, 2011.

[45] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*. Berlin, Germany: Springer, 2010, pp. 15–34.

NISHAT I. MOWLA (Student Member, IEEE) received the B.S. degree in computer science from Asian University for Women, Chittagong, Bangladesh, in 2013, and the M.S. degree in computer science and engineering from Ewha Womans University, Seoul, South Korea, in 2016, where she is currently pursuing the Ph.D. degree. She worked as a Senior Teaching Fellow with Asian University for Women. Her research interests include next-generation network security, the IoT network security, machine intelligence, and network traffic analysis. She received the Best Paper Award at the Qualcomm paper awards 2017, Ewha Womans University, and Korea Paper Competition.

NGUYEN H. TRAN (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the Ho Chi Minh City University of Technology, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University, South Korea, in 2011. He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University, from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, where he is currently a Senior Lecturer. His research interest is to apply the analytical techniques of optimization, game theory, and stochastic modeling to cutting-edge applications, such as cloud and mobile edge computing, data centers, heterogeneous wireless networks, and big data for networks. He received the Best KHU Thesis Award in engineering, in 2011, and the Best Paper Award at IEEE ICC 2016. He has served as an Editor for the 2017 Newsletter of Technical Committee on Cognitive Networks on Internet of Things. He has been an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, since 2016.

INSHIL DOH received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Ewha Womans University, Seoul, South Korea, in 1993, 1995, and 2007, respectively.

From 1995 to 1998, she worked with Samsung SDS of Korea. She was a Research Professor with Ewha Womans University and Sungkyunkwan University. She is currently an Associate Professor with the Department of Cyber Security, Ewha Womans University. Her research interests include wired/wireless network security, sensor network security, and the IoT network security.

KIJOON CHAE received the B.S. degree in mathematics from Yonsei University, in 1982, the M.S. degree in computer science from Syracuse University, in 1984, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, in 1990.

He is currently a Professor with the Department of Computer Science and Engineering, Ewha Womans University, Seoul, South Korea. His research interests include blockchain, security of FANET, sensor networks, smart grid, CDN, SDN and the IoT, network protocol design, and performance evaluation.