https://doi.org/10.7840/kics.2023.48.6.648

DATA-FedAVG: Delay-Aware Truncated Accuracy-Based Federated Averaging for Intrusion Detection in UAV Network

Vivian Ukamaka Ihekoronye◆, Cosmas Ifeanyi Nwakanma*, Dong-Seong Kim*, Jae-Min Lee°

ABSTRACT

In federated learning, the synchronous approach employed by the aggregating algorithm in the federal server, such as federated averaging (FedAVG), introduces high network communication costs, thus rendering it unsuitable for securing a network of unmanned aerial vehicles. This approach impedes the convergence speed of the global model and degrades its performance by increasing the number of participants. This study proposes a novel optimized aggregating algorithm called delay-aware truncated accuracy-(DATA)-based FedAVG. DATA-FedAVG is robust to the contingencies of straggling edge servers/clients (owing to network connectivity issues and system heterogeneity) and adaptively selects the fraction of clients whose model parameters are to be utilized in building the global model, thus optimally detecting intrusions in the network. In addition, the truncated client selection mechanism applied by DATA-FedAVG allows only clients with high-accuracy contributions to participate in both local training and federal updates. Extensive simulation experiments performed with a cybersecurity dataset validate the high performance of the proposed model and its reliability in accurately detecting attacks within an almost 75% reduced communication cost, while improving the performance of the intrusion detection model in terms of the average accuracy, recall, precision, and F1-score by 2%, 3%, 3%, and 3%, respectively.

Key Words : Anomaly Detection, Cybersecurity, Edge Computing, Federated Learning, UAV Network

Ⅰ. Introduction

In the last decade, an emerging and ubiquitous Internet of Things (IoT) device called an unmanned aerial vehicle (UAV), often referred to as a drone, has been extensively employed in both military and civilian applications. In a UAV network, several UAVs are connected via wireless technologies (such as WiFi, MAVlink, and Internet) to provide real-time information during operation. However, the resource constraints of UAVs (energy and computing limitations) impede their performance in achieving low latency, high data rates, and reliable services, which are the major requirements of wireless networks[1].

Incorporating edge computing (EC) technology into a UAV network can help mitigate the bottleneck of resource limitations experienced by the nodes in the network, thereby improving the network performance. EC is an innovative approach to alleviate the high data transmission costs and network traffic delays incurred when a data center in the cloud is utilized for data storage and analysis; additionally, it uses distributed servers to process time-sensitive data close to the periphery of the network. Consequently, the addition of EC servers to

※ This research was supported by Kumoh National Institute of Technology 2021
◆ First Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering; vivian@kumoh.ac.kr, 학생회원
° Corresponding Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering; ljmpaul@kumoh.ac.kr, 종신회원
* Kumoh National Institute of Technology, Department of IT Convergence Engineering, dskim@kumoh.ac.kr
논문번호 : 202304-073-B-RU, Received April 7, 2023; Revised May 17, 2023; Accepted May 18, 2023

the UAV network drastically reduces network outages due to insufficient battery energy. In addition, bandwidth limitations are overcome as telemetry data and network traffic are stored, cached, and processed in dedicated EC servers. Similarly, latency is minimized because applications and computing services are processed close to the UAVs where the data are generated[2].

However, UAV networks still suffer from aerial adversarial intrusions because of their operation in hostile environments and dependency on wireless communication technologies[3]. Cyberattacks and physical attacks are the two major categories of attacks that can be launched in UAV networks. Whereas cyberattacks involve data manipulation that can be launched remotely, physical attacks involve physical damage to devices in the network. As the UAVs in a network constantly communicate with each other utilizing wireless communication technologies (such as Wi-Fi, Bluetooth, and 5G) and open-source software (such as MAVLink), the susceptibility of the network to cyberattacks is very high. Off-the-shelf UAVs have been designed using limited functional mechanisms[4].

For instance, a malicious UAV can transmit superfluous data to overflood the limited bandwidths of legitimate UAVs in a network, thus causing a denial of service (DoS) or jamming attacks[5]. Hence, to realize the complete potential and global adoption of UAVs in the aerospace industry, an intelligent intrusion detection model must be incorporated into the network for robust network security and optimal performance to guard against the pervasive effects of cyberattacks in this time-critical network[6]. Although conventional machine learning (ML) and deep learning (DL) models have displayed renowned success for intrusion detection tasks, these schemes are cloud-centric with high communication overheads because of the large volume of data used in training the models[1].

Moreover, the central server has access to all network data generated and communicated by the UAVs during operations, thus leading to privacy issues[7]. Consequently, the centralized ML and DL models are unsuitable for UAV networks. Additionally, these models introduce high complexities that overwhelm the computing capacity of the UAVs during real-time critical mission operations. In addition, ML and DL models have low convergence speeds and low accuracy capabilities when trained with nonindependent and identically distributed (nonIID) data, that is, dissimilar data distributions[8].

Federated learning (FL) is a promising technique proposed by Google for promoting edge device learning and foster data privacy preservation[9]. Most FL research aims to optimize aggregating algorithms, such as federated averaging (FedAVG). The need to implement FedAVG is owing to the inherently high communication cost[10] and model degradation[11] of FedAVG when aggregating the model parameters of clients, owing to its synchronous approach and random selection of clients. For instance, it solves uncertainty and randomness in the selection of clients in an industrial IoT environment.

The authors of [10] proposed an accuracy-based client-selection mechanism. Additionally, to accelerate the convergence of the global model, only the parameter contributions from high-performance clients were used to update the global model. Similarly, in [12], FedAVG was modified by incorporating a dynamic learning rate to adapt to a fading channel in a wireless data aggregation scheme utilized for over-the-air computation. Although the private data of clients are preserved when utilizing the enhanced FedAVG in these works, the challenge of straggling clients in the network was not considered.

Therefore, this study focused on two phases. First, our proposed algorithm is robust to straggling clients in the network while still achieving a high performance the global models in an efficient manner. Second, the delay-aware truncated accuracy (DATA) FedAVG designed in this study was employed to provide maximum security to the UAV networks against diverse types of attacks. Nonetheless, this study made the following contributions.

1. A decentralized privacy-preserving method was proposed for data analytics and artificial modeling within a federated learning framework. A sophisticated intrusion detection model was developed through the collaborative participation of UAV-edge servers in real-world network contingencies.
2. An efficient lightweight multilayer perceptron that serves as the principal AI attack detection model for both the client and federal servers in the federated network was implemented.
3. Owing to the stringent and adaptive strategy applied by DATA-FedAVG during the selection of the fraction of clients participating in both local training and general updating, which is crucial for a security system, straggling clients were efficiently accommodated, even at a reduced communication cost, without affecting the network's performance.
4. Several simulation experiments were conducted to investigate the impact of the straggler effect owing to an increased fraction of client participation and the reliability, effectiveness, and efficiency of the proposed algorithm in terms of network scalability. Thus, DATA-FedAVG was subjected to different hyperparameters and variable compliance with the UAV network and FL settings based on the average performance.
5. The performance of the proposed algorithm was validated by comparing it with other state-of-the-art aggregating algorithms to determine the detection accuracy, processing time, and rationality in terms of other essential metrics when trained and tested with a cybersecurity dataset.

The remainder of this paper is structured as follow: The background of the study and extant literature are discussed in Section Ⅱ. The proposed intelligent framework is extensively discussed in Section Ⅲ. The experimental setup and results, as well as, a comparison of existing frameworks, are presented in Section Ⅳ. Finally, the conclusions n of this study are presented in Section Ⅴ.

Ⅱ. Background of Study and Related Works

The bulk research on extant literature considered in this article is focused on two key strategies for providing security mechanisms to UAV networks: 1) Centralized-based Intrusion detection frameworks; 2) Federated learning-based Intrusion detection frameworks.

2.1 Centralized-Based Intrusion Detection Model for UAV Network

Basically, intrusion detection models (IDMs) are hardware/software integrated into the UAV network to monitor anomalous traffic data and to alert the network of any deviations in forms of intrusions/attacks. The ultimatum of every IDM is to guarantee resilient protection over various attacks. Network-based IDM (NIDM) and host-based IDM (HIDM) are the two major types of IDMs, categorized based on their data source[2]. While HIDM monitors specific operating system applications to discover malicious data, NIDM investigates the traffic patterns of nodes in the network to discover unusual traffic targeted to impede the overall performance of the network[13].

According to the detection approach, IDMs can be classified as either signature-based or anomaly-based. With a database of predefined patterns/rules, signature-based IDM can detect previously known attacks in the network. On the other hand, anomaly-based IDM is widely used to detect known and novel attacks. ML/DL models utilized for attack detection and prediction tasks are trained with network traffic data containing benign and anomalous network behaviors. Hence, network traffic with deviation from models' thresholds is subsequently classified as intrusive actions from attackers[3].

Both ML/DL models such as random forest classifier, support vector classifier, decision tree, dDeep reinforcement learning, long short-term memory (LSTM), convolution neural network

(CNN), etc., have been widely used as anomaly-based IDM to provide security to the drone network. Autonomous detection of attacks by UAVs in the network was enabled by the adoption of a deep-Q-learning model designed by [14]. To secure the communications of drones in a software-defined environment, authors in [15] proposed a CNN model for the extraction of data features, a deep auto-encoder model for data dimensionality reduction, and an attention mechanism that was used to improve the features of important data for fast convergence of the model. In addition, a real-time data analytics framework was designed in [16] based on the LSTM model to investigate intrusions launched at the drone network. In [17], an optimized random forest model was used as the baseline algorithm for the IDM. We envisaged the integration of the IDM in dedicated edge servers for securing the communication link of the nodes in the UAV network from DoS and other malware attacks.

Despite this plethora of works, some limitations still exist. Firstly, modeling ML algorithms for intrusion detection tasks with complex datasets having multidimensional features (feasible in real-world scenarios) can be tedious and time expensive, due to the manual preprocessing steps needed to build highperformative models. The expensive time cost no doubt contradicts the time-criticality attribute of IDM essential during real-time robust protection against invaders. Moreover, in the centralized scheme, no preservation of data privacy as data generated by the different IoT devices in the network are sent to the central server for the training of the AI model.

2.2 Federated Learning (FL) Techniques

The general concept of FL is to promote on-device training of AI models without compromising data privacy. Edge servers make use of the local private data from IoT devices to collaboratively train a shared global model following a decentralized approach in a federated network. Specifically, in an FL network, the aggregator publicizes the parameter of the global model to a random set of clients (edge servers) in the network. After which, the global model is trained individually by a fraction of selected clients utilizing their private data. At the completion of the local training by the participating clients, only their model parameters (excluding local data) are synchronously uploaded to the federal server where the aggregation is done and a new global model emerges. Subsequently, other rounds of training commence, involving selected clients downloading the new global model and again locally training this model with their private dataset. This process of downloading, updating, and creating a global model is continued in the FL network until the global model converges given the a desired s performance or a pre-defined number of federal rounds.

According to [18], clients' data can be partitioned following the horizontal, vertical, and transfer federated learning formats. Thus, in this work, the horizontal FL approach is adopted in an FL scenario where the cybersecurity datasets comprise different clients having similar features with different sampling spaces reflecting the data generated by each fleet of UAVs deployed to perform surveillance operations in different environments. Therefore, the data generated by the UAVs in a particular cluster is peculiar to the cluster, like one-to-one mapping.

On the other hand, the vertical FL approach is set up in situations where the data features are dissimilar but captured in a similar sample space. Whereas, the transfer FL scheme is a combination of both different data features and sampling space. Furthermore, characterized by the resource capacities of clients in an FL network, the cross-silo and cross-device settings are the two types of FL schemes[19]. In a crosssilo FL setting, the clients' data are generated from organizations/entities having a wide data distribution and few participating clients. Also, these clients have substantial networking and computational resources. Whereas the cross-device setting considered in this study and applicable to most wireless IoT networks, has numerous participating clients with limited network resources and fluctuating communication connectivity, thus, displaying the tendency of

training latencies during the updating process due to intermittent availability of the devices.

Considering the spatial heterogeneity, high mobility, and susceptibility of the UAV network to jamming attacks, authors in [5] proposed an enhanced FL that prioritized the clients' groups leveraging the Dempster-Shafer theory. This theory enables the aggregator to select high-performing client groups to participate in updating the global model. However, no specific structure of the ML/DL model was stated as the algorithm used for detecting the jamming attacks in the network for reproducibility sake.

To provide security for wireless edge networks, both [2], [11] designed FL-based attention-gated recurrent unit (FedAGRU) and FedACNN respectively. Both works tried to improve FedAVG by integrating attention mechanisms. The idea is to utilize only local clients with increased weights for federal aggregation to reduce communication delays during network attack detection. To curb the use of rogue drones, [20] proposed a drone authentication model based on FL while securing the local models' parameters with homomorphic encryption during federal aggregation.

Authors in [21] enhanced the efficiency and reliability of data sharing in UAV-enabled networks, by developing an asynchronous FL scheme due to the heterogeneous resource capacity of the devices in the network. Moreover, a selection strategy was introduced to utilize only essential devices with a certain accuracy threshold to participate during local training, to speed up the convergence of the global model. Other researchers have also investigated different security frameworks to combat intrusions in the UAV network[5] and other wireless networks[2,11,22-25], leveraging FL techniques.

However, most of these intrusion detection frameworks are deployed at the core network which does not measure the actual security needs of dynamic networks[2]. Nevertheless, although contributions were made in the above research works by providing security defense in various networks, improvement is paramount since the design of intrusion detection algorithms in FL networks is still in the developmental stage. Most of the works that employed FL techniques in the UAV network neglected the security aspects. It is essential to state that only a minuscule amount of research has been done employing the FL technique for intrusion detection tasks in the UAV network.

In an attack scenario, the time needed to detect a possible attack could become a crucial factor in halting the pervasive occurrence of the attack given the volatility of the UAV network. In such cases, neglecting the straggler effect of FL can exacerbate the situation. The straggler effect in FL is a situation that occurs when some of the edge servers in the network expend a long time to send local parameters. In such a situation, the overall learning process can be impeded, leading to the slow convergence of the global model[26]. Hence, to mitigate the straggler effect, this work proposes a stringent mechanism that aids the aggregator to perform global aggregation given the contributions of clients that uploaded their model parameters within a specific time stamp. Furthermore, to speed up the convergence of the global model client selection of each particular round is based on the accuracy contributions of participating clients with respect to the accuracy threshold, evaluated by the aggregator based on the validation dataset.

Ⅲ. System Model and Implementation

An illustration of the edge-assisted UAV network is given in this section, as well as the proposed intelligent aggregating algorithm employed for intrusion detection tasks in the federated network.

3.1 Problem Definition

An edge-aided UAV network designed to detect attacks in a federated learning setting; consisting of a global cloud server, K flying edge servers, and P UAVs belonging to k number of clusters, as captured in Fig. 1 is considered in this research. Each K_{th} edge server during network authentication is designated as master controlling specific k_i cluster. If p is the summation of data samples generated by the UAVs during a mission-critical operation, and

the ith UAV has its custom-generated dataset depicted as U_i, with p_i samples of data, then the objective of the FL system is to minimize the global loss function when predicting anomalous data from normal traffic data, using Equation 1. Thus, taking cognizance of the weighted sum of the local loss from each K_{th} edge server during local training.

$$F_i(\theta) = \frac{1}{p_i} \sum_{q \in U_i} f_k(\theta; p_{i,q}, r_{i,q}), \tag{1}$$

where θ represents the parameter vector, ($p_{i,q}$, $r_{i,q}$) depicts the input and the predicted output respectively, and $F_i(\theta)$ is a specific loss function, in this work, the sparse-categorical.

As envisioned in this research, the UAV network is exploited for reconnaissance operations involving spatiotemporal activities embarked on by the fleet of UAVs. Before embarking on the network's mission, the drones are authenticated in the different clusters with the objective to navigate and communicate network traffic data with each other and their designated edge servers in a multi-hop manner given an established routing protocol. The drones equipped with adequate sensors such as camera, lidar, and other sensors are configured to survey their operational environments whilst sending information (images, and live video feeds) to the dedicated edge servers for real-time data analysis. That is, the heterogenous data generated by each cluster is handled by the edge servers as communication between drones in different clusters is restricted, to limit the risks of unauthorized access to sensitive network data.

In an FL security network, the server (also called the cloud server) instantiates the global model and broadcasts its parameters to all participating clients (UAV edge servers) servers in the network. The goal is to promote data privacy in each cluster by allowing the clients to utilize their private data and collaboratively train the shared model in a distributed manner. At the end of each local training, the clients send their model parameters to the federal server for aggregation and global model updating as detailed in Fig. 1. Clients' updates are ephemeral and aggregated as soon as possible at every communication round. After aggregation, the updated global model parameter is sent back to the

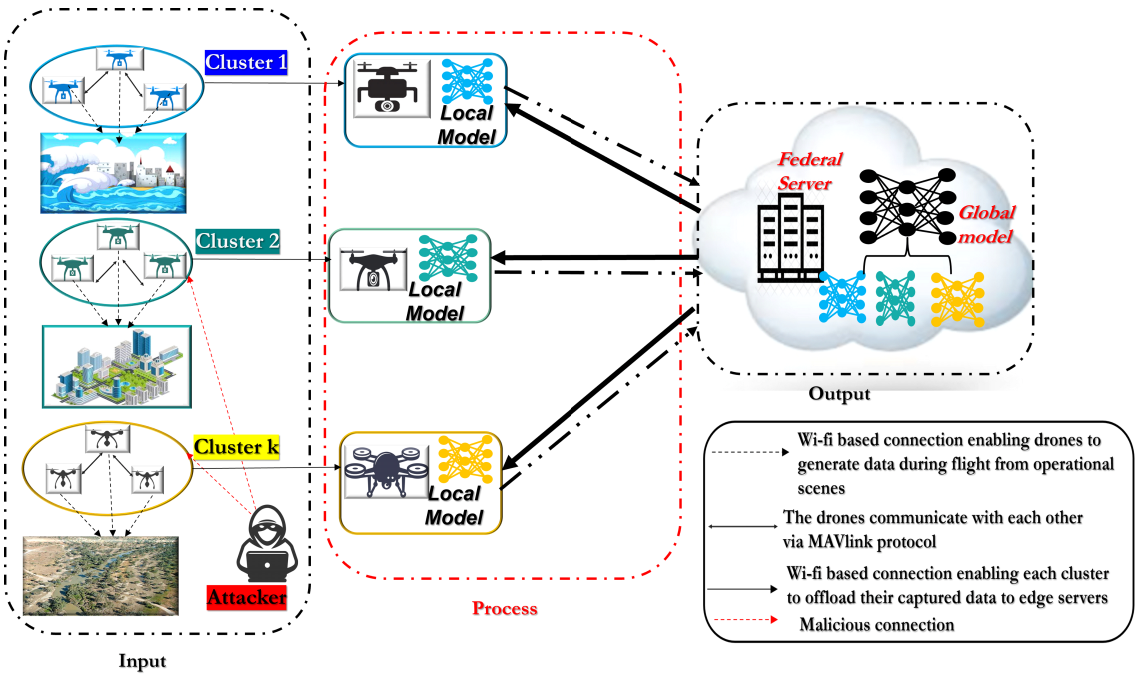

Fig. 1. Intrusion Detection Model Based on Federated Learning Scheme

client servers for another round of training and parameter optimization involving a different batch of the dataset. Parallel communications between the federal server and the client servers are repeated synchronously until the convergence of the global model at a desired accuracy.

The aggregation function is an optimization algorithm that enables the federal server to aggregate the parameters sent by decentralized and heterogeneous clients. Since each client has a specific version of the global model, the aggregation function coordinates each client's contribution (parameter) based on its in-trinsic setup to build an updated global model without having access to the client's private data. The goal is to create a robust model that is representative of the individual clients' models. Several aggregation functions can be utilized by the federal server for aggregating the model parameters from the collaborating clients, with FedAVG and federated stochastic gradient descent (FedSGD) being the most widely used.

FedAVG algorithm is based on the traditional stochastic gradient descent (SGD) used in optimizing DL models. It performs optimization by sending the current state of the global model to a subset F of clients that execute local training by running ϵ epoch (the number of forward and backward passes of training samples, also called local iterations) of mini-batch β (the number of training samples in a single forward and backward pass, also known as batch-size) SGD, after which the clients' gradients will be sent to the server for averaging. The averaging and updating processes are repeated over a number of communication rounds R until convergence of the global model or certain criteria is achieved.

FedAVG is beneficial to reduce the communication bottlenecks in the FL network because it applies two key strategies; performs multiple local SGD updates and communicates with only a subset of the clients. Following these strategies, the global model converges within fewer communication rounds when the batch SGD update for each communication round is adequately chosen[27]. That is, if the parameters (ϵ, β, R) facilitating the optimization operation of FedAVG are not effectively selected, it can potentially lead to the performance degradation of the global model. Contrarily, FedSGD updates the parameter of the global model by concatenating the gradient computed by each local client and performing traditional SGD as its optimization technique. Although FedSGD is computationally effective, its communication cost is expensive[11]. Nonetheless, when ϵ =1 and β = 8, invariably FedAvg " FedSGD.

However, the limitations of both averaging functions in terms of suboptimal model performance when subjected to non-IID data[10,28] and expensive communication costs[29] respectively, necessitate the improvement. Considering the critical importance of the UAV network, especially when deployed for reconnaissance and surveillance operations, and the targeted sophisticated types of attacks experienced in this network, a security layer is integral. Therefore, a security mechanism that is responsive, resilient, reliable, and robust is expedient.

3.2 Delay-Aware Segmentation against Stragglers

As earlier stated, in the R_{th} federal round of a standardized FL scheme, the weight of the initial global model W_g is sent to randomly selected F_t clients amongst all K clients in the network. For each round of local training, E performed by F_t clients with their own private data, parameters of the local models are sent to the server.

While handling straggling clients, we assume a semi-asynchronous approach; a periodic server aggregation. At each federal round R, the optimized global model alongside a time stamp (W_{g+1}, t) is sent to F_t clients. That is, given a hard deadline T_d, the optimized model W_{g+1} is built based on the parameter aggregation of the clients who updated the server within the deadline period. However, the parameter contributions of the straggling clients are not neglected but are used in the next federal round R + 1 or in subsequent rounds, depending on the delay.

Assuming $Q_t^{(R)}$ is the set of clients that uploaded their model parameters within the given time threshold at the federal round r for $r \geqslant R$. Therefore,

$F_t = Q_i^{(\emptyset)} + Q_i^{(R)}$, where $Q_i^{(\emptyset)}$ is considered as the set of clients randomly selected at federal round R but whose model parameters were not uploaded to the federal server due to time delay. Consequently, the clients whose model parameters arrived within the time deadline at federal round R are categorized into one of the $R+1$ sets: $Q_R^{(0)}$, $Q_R^{(1)}$, $Q_R^{(2)} \cdots Q_R^{(R)}$.

Moreover, the model parameter sent from client $k \in Q_R^{(r)}$ after E local updates with weight w_r is denoted as $w_r(k)$. Since we aim to optimize FedAVG without the straggling clients influencing the emergence of the global model for every round r, the weighted average is performed on the clients' model parameters using equation 2:

$$z_{R+1}^{(r)} = \sum_{k \in Q_R{}^{(r)}} \frac{p_i}{\sum_{k \in Q_R{}^{(r)}} p_i} w_r(k), \tag{2}$$

where $z_{R+1}^{(r)}$ is the local weighted result of a group of representative models received by the federal server at round R with lag $r - R + 1$.

Furthermore, the weighted average of the straggling clients is considered for the same federal round R according to the level of the time lag: $\sum_{r=0}^{R} \alpha_R^{(r)}(\lambda) z_{R+1}^{(r)}$. Where $\alpha_R^{(r)}(\lambda) \propto \frac{\sum_{k \in Q_R^{(r)}} p_i}{(R-r+1)^{\lambda}}$ is a regularized coefficient that is directly proportional to the amount of dataset in Q_R^r and inversely proportional to $(R - r + 1)^{\lambda}$, for every lag exponent $\lambda \geqslant 0$. The goal is to have a larger weight assigned to $z_{R+1}^{(r)}$ (clients with the updated optimized global model at round R) with a least lag $(R - r + 1)$. Given the weighted sum $\sum_{r=0}^{R} \alpha_R^{(r)}(\lambda) z_{R+1}^{(r)}$, w_{R+1} can be calculated as

$$w_{R+1} = (1 - \delta) w_R + \delta \sum_{r=0}^{R} \alpha_R^{(r)}(\lambda) z_{R+1}^{(r)} \tag{3}$$

where λ is the average time coefficient. For subsequent round $R+1$, a new subset of clients C_{R+1} is selected by the server, and the new global model with timestamp w_{R+1}, $t+1$ is sent to the clients for another round of local training. In essence, the proposed model is robust to straggling clients, as the updating of the global model is not affected by the delay of stragglers. The contributions of straggling clients are not neglected but utilized in future federal rounds.

3.3 Truncated Accuracy-Based Federated Averaging

While the global model is being optimized by the aggregation of clients' model parameters, a validation dataset is available at the federal server to evaluate the attack detection performance of each participating client. Let N_{val} be the number of validation datasets containing the different anomalous and benign datasets in the server. During client parameter updating, the server evaluates the accuracy performance of each client using the validation dataset, with the given equation:

$$Acc(k) = \frac{\sum_i (acc_i * p_i)}{N_{val}} \tag{4}$$

where acc_i is the accuracy of the i_{th} client's model on its local dataset, p_i is the number of samples in the i_{th} client's local dataset and N_{val} is the validation dataset. Therefore, with an accuracy threshold of $\geqslant$ 90%, a list of client selections at each round of global training is achieved, to truncate clients with attack detection < 90.

To the best of our knowledge, the proposed algorithm in this research is the first work that provides a security solution to a cyber-physical network like that of the UAV, with stringent averaging approaches. While considering the real-world challenges of network connectivity issues and clients' resource limitations resulting in straggling clients, the delay aware truncated accuracy (DATA) federated averaging algorithm helps to mitigate the high communication overhead predominant to FL scheme, at the same time enhance the security resiliency of the UAV network.

3.4 Deep Learning Model for Intrusion Detection

The principal model that is continuously trained and improved by the collaborative participation of

clients in an FL network to perform a specific task, is either an ML or DL model. In this study, a simplified neural network known as the multilayer perceptron (MLP) is employed for attack detection and prediction task. In the architectural structure of an MLP, the network traffic of the UAV network is received by the input layer, while the processing of the input data is handled by the hidden layers, and the various anomalous and benign labels are classified by the output layer, with the neurons as the basic building blocks of the network.

Consequently, the number of hidden layers and neurons are influential parameters in determining the performance of the MLP model. However, as a rule of thumb, a complex MLP is designed using a large number of hidden layers and neurons, resulting in an inevitable trade-off between the complexity and overfitting of the model. In this regard, employing MLP for intrusion detection in the UAV network requires careful consideration in the selection of the number of hidden layers and the number of neurons constituting these hidden layers, as both the input and output layers are dependent on the number of features (independent variables) and labels (dependent variables) respectively in the dataset.

The MLP model design captured in Fig. 2, displays a simplified architecture comprising the input layer, five hidden layers, and the output layer with 71 neurons stacked together. The choice of a lightweight model for detecting attacks in the UAV network is to reduce computational complexity during local training and global modeling and to ensure speedy intrusion detection during eventual implementation. In addition, sixteen neurons representing the number of features in the datasets constitute the input layer, which is densely connected to the ten neurons in the first hidden layer, as well as the subsequent ones until the output layer containing 5 neurons depicting the 5 classes of normal and anomalous network traffic. Intuitively, each neuron in a given layer (except in the input layer) receives inputs from the previous layer which is a weighted sum mapped with a nonlinear activation function, to produce outputs that are transferred to neurons of subsequent layers in a similar format.

In essence, each layer starting from the input feeds the next layer with their computational results until the classification task is completed by the output layer. To introduce nonlinearity to the model (allowing the model to learn more abstract representations of the input data), aiding in the alleviation of the vanishing gradient problem during backpropagation for effective and efficient model learning, and fast convergence of the model, the rectified linear (ReLU) activation function was used in each of the five hidden layers, mathematically expressed in equation 5 as:

$$z = \sum(w_i * x_i) + b \rightarrow f(z) = max(0, z) \quad (5)$$

where z is the linear combination of inputs represented by the weighted sum of inputs, x_i, w_i, and b are the input of the *ith* neuron, associated weight, and bias respectively. Whereas $f(z)$ is the activation function that performs a simple thresholding operation on each weighted neuron so that if $z \geqslant 0$ the exact value is given as output and if $z < 0$ the

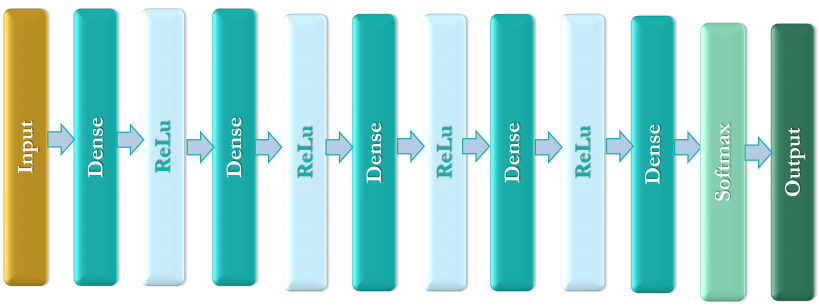

Fig. 2. Lightweight Sequential Multilayer Perceprron for Clients and Federal Server

value is set to 0.

Thus, setting the output of the ReLU function from 0 to positive 8. While the vector of real numbers from the output layer is mapped to a probability distribution over the five classes using the softmax activation function given in equation 6, the sparse categorical cross entropy was employed to optimize the model during training using the loss function in equation 7. Hence, calculating the difference between the predicted probabilities and ground truth.

$$\sigma(x^{\rightarrow})_i = \frac{e^{x_1}}{\sum_{j=1}^{K} e^{x_j}} \tag{6}$$

where σ is the softmax function, $x_i^{\rightarrow}$ is the input vector of the i_{th} neuron, K is the number of class labels and e^{y_j} is the standard exponential function of the output vector. In addition, a detailed description of the architectural layout and other hyperparameters used in the training of the proposed MLP model is highlighted in Table 1 and Table 2 respectively.

Table 1. Structural Layout of the Proposed Lightweight Multilayer Perceptron

Layer Type	Output Shape	No. of Parameter
dense_31 (Dense)	(None, 10)	170
dense_32 (Dense)	(None, 10)	110
dense_33 (Dense)	(None, 10)	110
dense_34 (Dense)	(None, 10)	110
dense_35 (Dense)	(None, 10)	110
dense_36 (Dense)	(None, 10)	110
dense_37 (Dense)	(None, 5)	55

Table 2. Hyperparameters Used for Model Training

S/N	Parameter	Value
1	Batch_size	32
2	Learning_rate	0.001
3	Optimization function	Adams
4	Activation function (Hidden layers)	ReLU
5	Activation function (Output layer)	Softmax
6	Loss function	Sparse_categorical_crossentropy

$$L(y, \hat{y}) = \frac{-1}{N} \sum_{i=1}^{N} y_i log(\hat{y_i}) \tag{7}$$

where N is the total number of samples in the dataset y_i represents the ground truth label for the i_{th} sample, and $\hat{y}_i$ is the predicted label of the same sample.

3.5 Experimental Setup

In this section, we succinctly described how the FL experiment was conducted, the dataset used for evaluating the proposed model, the preprocessing steps involved, and the simulation processes.

3.6 Simulation Setup

The implementation of the proposed truncated averaging algorithm was done using *Flower*[30] as the FL framework. As an agnostic framework, *Flower* can be used with Pytorch and Tensorflow (employed in this study) for building the DL framework. Also, Pandas and Numpy enabled the federated analytics. Google Colaboratory offered the computing resource for running the simulations, while Python 3.9.7 was used as the programming language, supported by *Flower* (1.1.0) and Tensorflow(2.9.1). Lastly, the Windows 10 operating system with the configuration of Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz, 8GB RAM, and GPU Tesla K80 make up the hardware specifications of the system used for the simulation.

3.7 Dataset Preprocessing

Since the UAV network is a type of wireless sensor network (WSN) where the sensor nodes are airborne UAVs, the WSN dataset (WSN-DS) a cybersecurity dataset[31], created for detecting the variants of DoS attacks commonly experienced in a WSN was used to evaluate the performance of the proposed model. Blackhole, Grayhole, Flooding, and Scheduling attacks constitute the four different types of attacks in the WSN-DS, containing 374661 samples in addition to benign (normal) samples reflecting a real-world attack scenario, with the description given in Table 3.

Although the WSN-DS has no missing and

'NAN' (not a number) values, feature scaling and label encoding were performed on the independent and dependent features respectively. To normalize the input features of the dataset to have a uniform distribution so that each feature contributes equally to the task of attack detection, the feature scaling technique for data normalization was employed. The standard scaler (Z-score normalization) feature scaling technique was used to independently standardize each 16 feature in the training set to a mean of 0 and a standard deviation of 1, achieved by subtracting the mean and dividing by the standard deviation using the formula:

$$X_{scaled} = \frac{X - \mu}{\sigma} \tag{8}$$

where X is the initial feature, μ is the mean of the feature, σ is the standard deviation and $Xscaled$ is the resultant scaled feature. On the other hand, the categorical target classes were label encoded by assigning an arbitrary integer value (0 to 4) to each unique category representing the five classes in the dataset.

Furthermore, to achieve a real-world FL scenario where all the clients generate and individually train local models, depicting non-iid data compliance, the dataset was divided into various local datasets for training. A closer look at Table 3 indicates the highly imbalanced dataset distribution used in this experiment, which is a reflection of a feasible real-world attack situation. To ascertain the efficacy of the proposed model when deployed in the real world, applying an oversampling technique to balance the class distribution was completely ignored. However, to avoid the model from overfitting, the dataset was split into 70:20:10 for training, testing, and evaluation statistically analyzed also in Table 3.

To design an FL network consisting of a server and multiple clients, *Flower* implements the *'get-parameters'*, *fit*, and *evaluate* methods. While the *'get-parameters'* uploads the local parameters from the clients to the server, the *fit* method enables each instance of the client to download, locally train and upload optimized parameters to the server, whereas the *evaluate* method evaluates model parameters received from the server on the local dataset and

Table 3. Statistical Distribution and Description of WSN-DS

S/N	Traffic Type	Description	Sample Size	Training Set	Test Set	Val.Set
1	Blackhole	A type of DoS attack where the attacker advertises himself to the drones in a cluster as the master controlling the cluster. Thus, intercepting and discarding all data packets forwarded, with the aim of disrupting communication.	10049	7034	2010	1005
2	Grayhole	Similar to the blackhole attack, but instead of discarding all the data packets, the attacker selectively forwards some of the data after modifications, thereby, compromising data communication.	14596	10217	2920	1459
3	Flooding	This DoS attack is aimed at overwhelming the network bandwidth with a high volume of traffic, so as to degrade the network performance.	3312	2318	662	332
4	Scheduling	The attacker aims at manipulating the scheduling tasks of the UAVs, leading to disorientation or a worst-case scenario complete idleness of the UAVs during time-critical missions.	6638	4646	1328	664
5	Normal	Benign telemetry data and real-time commands and control signals in the network.	340066	238046	68014	34006

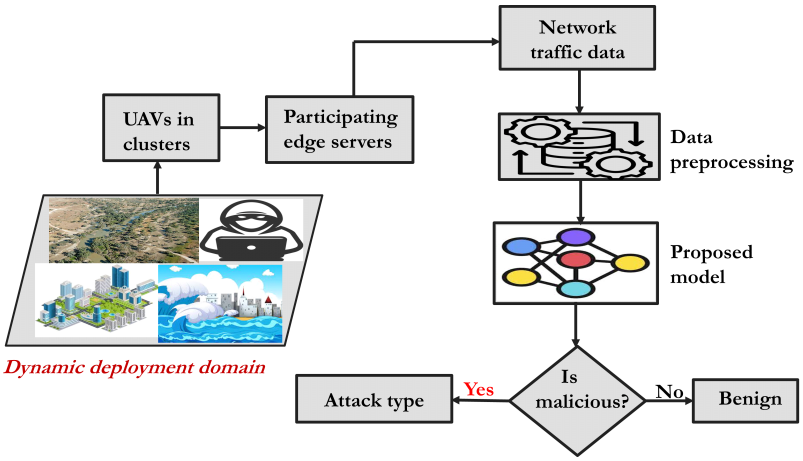

Fig. 3. Overall Attack Detection Flow of the Federated Intrusion Detection Model For a Secured Edge-Assisted UAV Network

sends the evaluation result to the server. It is worth noting that the lightweight MLP model was utilized during experimentation. Although *Flower* provides (FedAVG) and accommodates (FedSGD) averaging functions and other aggregating algorithms, the proposed truncated averaging function designed in this study served as the aggregating algorithm in the global server. Lastly, the overall process of providing security to the UAV network utilizing the proposed algorithm is displayed in Fig. 3.

To evaluate the performance of the proposed algorithm, several experiments were conducted based on varying parameters explicitly given in Table 4. In addition, this work investigated the performance of FedAVG and FedSGD aggregation algorithms alongside the proposed truncated averaging algorithm when subjected to different federated variables to validate the robustness of the proposed algorithm.

Ⅳ. Result Discussion and Performance Evaluation

Table 4. Parameters Utilized During Simulation

Parameter	Value
Local epoch (E)	1, 5
Communication rounds (R)	40
Number of clients (K)	20,30,40,50
Participating fraction of clients	0.5, 0.6, 0.7, 0.8
Model	MLP

In this section, the results obtained by DATA-FedAVG during several simulation experiments are discussed. Also, we evaluated the performance of the proposed algorithm alongside state-of-the-art FL algorithms based on essential metrics, and their results are presented herein.

4.1 Performance Metrics

- True Positive (TP): It is the amount of correctly predicted attacks from the total attack samples.
- True Negative (TN): It is the amount of normal data accurately detected as benign.
- False Positive (FP): It is the amount of incorrectly predicted benign data as attack data.
- False Negative (FN): It is the amount of incorrectly predicted attack data as benign data.
- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

- Precision:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

- Recall:

$$Recall = \frac{TP}{FP + FN} \tag{11}$$

- F1-Score:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

▪ Loss:

$$L(y,\hat{y}) = \frac{-1}{N}\sum_{i=1}^{N} y_i log(\hat{y_i}) \quad (13)$$

▪ Processing time

4.2 Performance Evaluation

This research focuses on reducing the high communication cost caused by straggling clients in the FL network, especially in attacks scenario, while improving the performance in securing the network. Thus, the WSN-DS cybersecurity is used for the simulation experiments and comparison is done with some state-of-the-art aggregating algorithms like FedAVG[27] and FedSGD[32]. The reliability of DATA-FedAVG was evaluated by iterating the simulation three times and then averaged using six different performance metrics.

As highlighted in Table 5, the performance of DATA-FedAVG is recorded given a local training iteration $E = 1$), clients fraction size of 5% $F = 0.5$ total client size in the network $K = 20$ for a communication round of $R = 40$. As detailed in Table 5, the global model optimally converged at an accuracy of 98.08%, loss of 0.042 precision of 90.46%, recall of 91.76%, and F1-Score of 91.09% during the communication round of 30. However, the least training time of 165secs is achieved during a communication round of 5, buttressing the trade-off between communication overhead and optimal performance experienced in the FL network.

4.2.1 Comparison of the Proposed Algorithm with State-of-the-Art Algorithms

For a more viable evaluation, the proposed model was compared with two popularly deployed federated algorithms; FedSGD[32] and FedAVG[27]. Given a local epoch of 5, 20 participating UAV edge servers as clients in the network, 5% fraction of clients, and a global communication round of 40, the prediction performance of the three algorithms based on accuracy and loss can be visualized in Fig. 4.

As displayed in Fig. 4, the global convergence of FedSGD and FedAVG was achieved at a prediction accuracy of less than 98% during the federated communication of 35. Conversely, at a reduced communication round of 20, an optimized global model is created with an accuracy even greater than 98% with the proposed algorithm as the aggregation function. Also, at the least communication round of 5, the proposed algorithm is capable of detecting attacks with an accuracy of 98%, indicating its capability of reducing the network communication cost by 75%. Thus, DATA-FedAVG is more efficient than FedSGD and FedAVG. Categorically, the proposed algorithm will immensely tackle the communication overhead issues caused by high communication rounds in a real-world federated setting, as within a minimal communication round high detection accuracy is guaranteed.

Similarly, the losses recorded by the algorithms are captured in Fig. 4. The loss graph in Fig. 4 depicts the discrepancies of each of the algorithms during predictions. High losses ranging from 0.06 to 0.05 can be observed to be obtained by both

Table 5. Performance of Proposed Algorithm @ E = 1, K = 20, F = 0.5

Round	Time (secs)	Accuracy (%)	Loss	Precision (%)	Recall (%)	F1-Score (%)
5	**165**	97.24	0.073	87.76	86.99	87.27
10	322	97.30	0.065	89.93	90.84	89.56
15	477	97.84	0.050	90.28	92.01	90.02
20	624	97.78	0.049	90.34	90.95	90.53
25	776	97.45	0.057	90.07	91.45	90.59
30	931	**98.08**	**0.042**	**90.46**	**91.76**	**91.09**
35	1060	97.85	0.047	90.37	91.53	90.52
40	1265	97.93	0.043	89.83	91.18	90.02

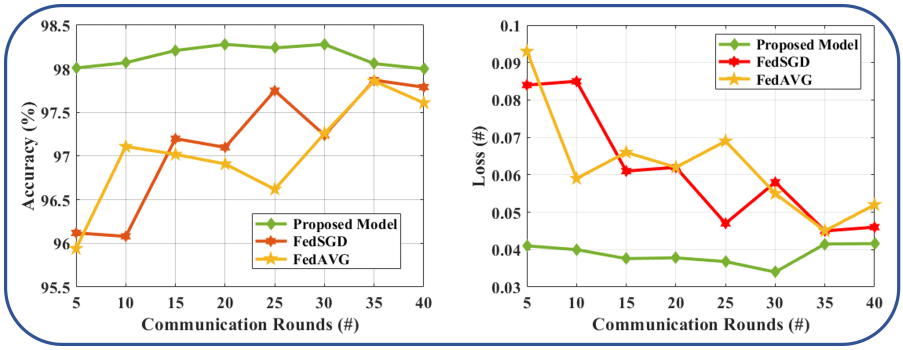

Fig. 4. Accuracy and Loss Recorded by the Proposed Algorithm and some-state-of-the-art Algorithms

FedAVG and FedSGD, with the highest loss recorded by FedAVG. Portraying the proneness of both algorithms to false negatives/ false positives. The lowest loss of average

0.037 during all communication rounds recorded by the proposed model shows its capability to correctly predict anomalous data from benign data, thus capable of securing the UAV network from cyberattacks.

4.2.2 Sensitivity Analysis and Evaluation: Impact of Varying Fraction of Participating Clients

This particular experiment is conducted to validate the robustness of DATA-FedAVG to straggling clients in situations of unsteady client participation, either due to network connectivity issues or resource limitations. In this experiment, varying fractions of clients, that is @ F = 0.6, 0.7, 0.8 were considered with a fixed total number of clients, K = 20, local training E = 5, and communication rounds of 40. This is to simulate a real-world scenario of straggling participating clients and observe the performance of the algorithms.

Fig. 5, captures the progressing uniform accuracy recorded by FedSGD and FedAVG during the communication rounds from 5 to 10, with consideration to all fractions of clients. A non-uniform increment can be observed amongst both algorithms when the communication round increased to 15 and 20 also in all fraction client sizes. However, a notable improvement of performance can be seen for FedSGD F = 0.8 (almost 1.3% accuracy increment). Typically in an FL setting, as the number of communication rounds increase, a corresponding improvement in the performance of the global model ought to be achieved, with a tradeoff on communication cost too. Ironically, FedAVG even with an increased fraction of clients (@ F =0.8), and a global iteration of 30, had its global model converged at a prediction accuracy of 97%. In contrast, the proposed DATA-FedAVG algorithm displays reliability to straggling clients scenario considering the highest global model prediction accuracy of 98.3% is achieved when the least fraction of clients F = 0.6 is participating in the updating of the global model for the detection of attacks. Also, visualizing Fig. 5 more closely, from rounds 15 through 20, DATA-FedAVG had similar accuracy performance amongst the varying fraction of clients, except in

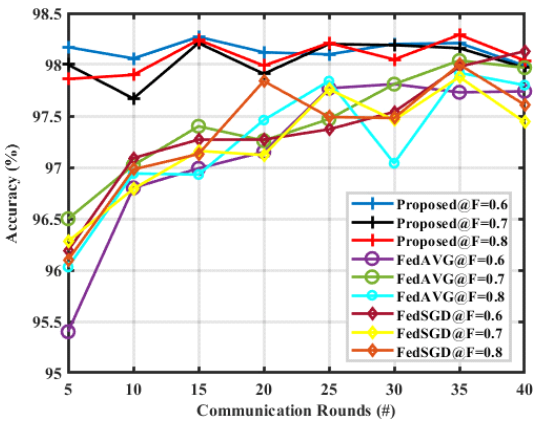

Fig. 5. Accuracy Performance of the Different Aggregation Functions for Varying Fraction of Clients @ E = 5, K = 20 and R = 40

round 35 for F = 0.8 where a significant increment of about 98.4% accuracy was achieved.

In addition, for a robust evaluation given the imbalanced dataset used, we investigated the performance of the algorithms based on other metrics, and their results are presented in Table 6. From Table 6, DATA-FedAVG achieved the best Precision, recall, and F1-score in all fractions of client variation. Moreover, a significant improvement is achieved by DATA-FedAVG at the highest fraction client F = 0.8, recording a precision value of 90.85%, recall of 92.33%, and F1-score of 91.58%. On the other hand, FedSGD performed better than FedAVG given all evaluation metrics, especially when F =0.6, with precision, recall, and F1-score of 89.87%, 91.07%, and 90.46% respectively. The overall enhanced performance of DATA-FedAVG validates the robustness of the proposed averaging algorithm in handling contingencies relating to inactive participation of some clients due to resource limitations or network connectivity issues and its capability to secure aerial wireless sensor network with minimal false alarm rate when deployed in the real world.

4.2.3 Time Complexity Analysis and Evaluation

Furthermore, the detection efficiency in terms of processing time of all three algorithms was evaluated with the same parameters, and the results are captured in Fig. 6. A steady processing time was achieved by all three algorithms @ F = 0.6, 0.7, and 0.8 from rounds 5 to 25. However, during round 30, FedSGD recorded the lowest and highest training times @ F = 0.6 and F = 0.8 respectively. Conversely, the proposed algorithm constantly maintained almost equal processing time for the different variations of clients but experienced spikes (latency issues) from rounds 30 to 40 @ F = 0.7, recording a processing delay of almost 1000secs. Although FedSGD had the least training time @ F = 0.6, the unwavering and considerable amount of time expended by the proposed model during all

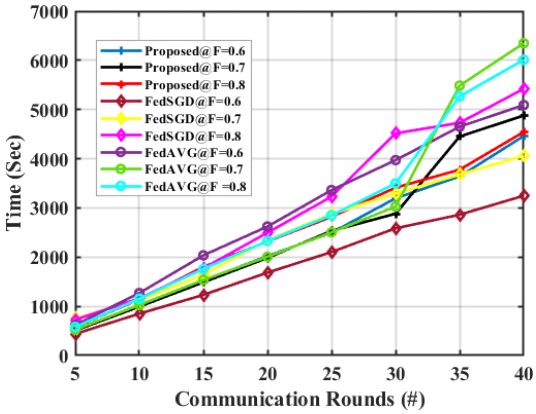

Fig. 6. Attack Detection Efficiency of the Proposed Model, FedSGD and FedAVG with Varying Fraction of Participating Clients, @ E = 5, and K = 20

Table 6. Sensitivity Evaluation of the Three Aggregating Algorithms with Varying Fraction of Clients F = 0.5, 0.6, 0.7 and 0,8, Given Total Client K = 20

Algorithm	Fraction Size (F)	Precision (%)	Recall (%)	F1-Score (%)
FedAVG [27]	F = 0.5	89.17	90.33	89.32
FedSGD [32]		89.78	91.15	90.44
Proposed		**90.46**	**91.76**	**91.09**
FedAVG [27]	F = 0.6	89.06	90.08	88.42
FedSGD [32]		89.87	91.07	90.46
Proposed		**90.56**	**91.59**	**91.06**
FedAVG [27]	F = 0.7	88.36	88.88	88.53
FedSGD [32]		87.45	88.20	87.81
Proposed		**90.18**	**90.66**	**89.42**
FedAVG [27]	F = 0.8	89.93	90.41	89.37
FedSGD [32]		89.50	91.04	90.26
Proposed		**90.85**	**92.33**	**91.58**

rounds of communication and fraction of client variations validates its attack efficiency and stability even with a fluctuating fraction of client size.

4.2.4 Scalability Analysis and Evaluation: Impact of Increasing Number of Clients Participation

To ascertain the performance of the proposed algorithm on a scalable network, simulation experiments were conducted with different client sizes of $K = 30$, $K = 40$, and $K = 50$, capturing the performances of all three algorithms based on their attack prediction accuracy and processing time in Figs 7 and 8 respectively.

In Fig. 7 and Table 7, an increase in the number of clients, for instance, $K = 50$, in Fig. 7 adversely affected the accuracy performance of FedSGD and FedAVG in building their optimized global model that can accurately predict malicious network traffic data. Considering both algorithms for $K = 50$, FedAVG accuracy decreased from almost 96% in round 10 to 95% in round 25. Likewise, FedSGD also recorded a reduced performance from almost 97% in round 10 to 96% in round 25 when $K = 50$. The performance degradation is a result of the aggregating principles of both algorithms. That is, the emergence of a convex global model is determined by the absolute contributions of the randomly selected clients amidst the delay encountered. Consequently, the network will experience a communication bottleneck, because as the number of clients increases, a corresponding

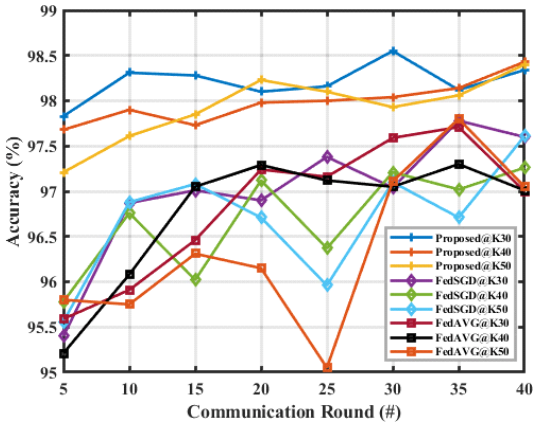

Fig. 7. Accuracy Comparison of the Different Aggregation Algorithms for Varying Client Size of K = 30, 40, 50, E = 5 and F = 0.6

increase in the communication between the clients and the server is inevitable. Thus, leading to laggard convergence and reduced accuracy of the global model.

In addition, while FedAVG had a better precision of 0.2% and 0.9% more than FedSGD when K=30 and K= 40 respectively in Table 7, FedSGD achieved a higher precision of 0.71% more than FedAVG when K=50. Based on the performance of both algorithms on other evaluation metrics highlighted also in Table 7, FedAVG outperformed FedSGD when the client sizes are K=30 and K=40, except for K=50 where FedSGD slightly outperformed FedAVG. For the performance of the proposed algorithm, also referencing Fig. 7 and Table 7, increasing the number of clients does not

Table 7. Scalability Evaluation of the Different Aggregating Algorithms @ Varying Clients Size K = 30, 40, 50 given E = 5, F = 0.6

Algorithm	Client Size (K)	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Time (secs)
FedAVG [27]	K = 30	89.59	90.87	89.22	97.60	3652
FedSGD [32]		89.39	90.00	89.69	97.00	4328
Proposed		**92.17**	**93.53**	**92.84**	**98.34**	**4069**
FedAVG [27]	K = 40	87.95	89.05	88.40	97.26	6119
FedSGD [32]		87.05	88.15	87.59	97.01	4270
Proposed		**90.53**	**91.89**	**91.20**	**98.43**	**5358**
FedAVG [27]	K = 50	88.37	87.97	88.81	97.61	6340
FedSGD [32]		89.08	90.19	89.63	97.05	7400
Proposed		**90.93**	**92.31**	**91.61**	**98.50**	**4281**

greatly affect the convergence of the global model.

In Fig. 8, no significant difference in the training time (1000 - 3500*secs*) among the algorithms during the communication rounds of 5 to 25 for all client sizes. While the proposed algorithm maintained an almost steady processing time for client sizes, an exploding processing time during the communication round at 35 was recorded for FedAVG when $K = 50$. Moreover, because DATA-FedAVG adaptively selects the fraction of clients based on models' accuracy contribution and their convergence speed during local training, stringently limiting the participating number of clients, thus, enhancing the efficiency of the global model and its reliability in providing robust security to a delay-tolerant network like the UAV.

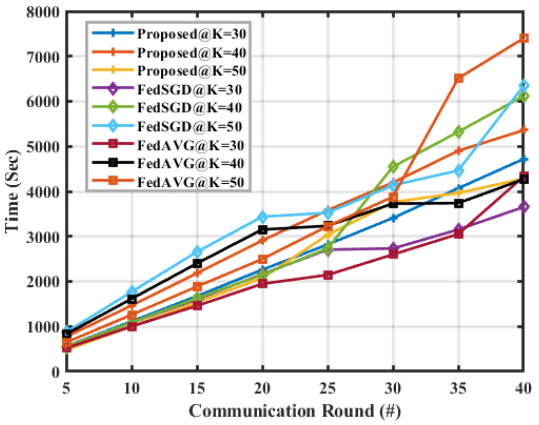

Fig. 8. Efficiency Comparison of the Different Aggregation Algorithms for Varying Client Size of K =30, 40, 50, E = 5 and F = 0.6

V. Conclusion

This paper proposes a reliable and optimized federated aggregation algorithm, DATA-FedAVG, for intelligently securing an edge-assisted UAV network from intrusions and cyberattacks. While ensuring privacy is achieved in each cluster in the network, DATA-FedAVG exploits a mechanism that allows only a fraction of clients with high attack accuracy contributions to participate in the federated learning, hence still accommodating straggling clients, to speed up global convergence within a minimal communication cost. Simulation experiments performed with the WSN-DS cybersecurity dataset and comparison with other federated aggregation algorithms ascertained the superior detection accuracy of DATA-FedAVG over FedAVG and FedSGD, even at a reduced communication round. DATA-FedAVG also demonstrated reliability for a scalable network, given its robust performance when the number of participating clients increases in the network. However, the additional computational complexity introduced by the proposed algorithm is the cost of adequately and efficiently detecting malicious network traffic data and can not be neglected. For our future work, we hope to contribute to the challenge of model and data poisoning from malicious clients in the federated network, by employing encryption techniques to mitigate such vulnerabilities.

References

[1] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for uavs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53841-53849, 2020. (https://doi.org/10.1109/ACCESS.2020.2981430)

[2] Q. Jiang, et al., "Intelligent intrusion detection based on federated learning for edge-assisted internet of things," *Secur. and Commun. Netw.*, 2021. [Online] Available: https://arxiv.org/abs/2106.09527.

[3] V. U. Ihekoronye, S. O. Ajakwe, D.-S. Kim, and J. M. Lee, "Hierarchical intrusion detection system for secured military drone network: A perspicacious approach," in *MILCOM 2022-2022 IEEE*, pp. 336-341, 2022. (https://doi.org/10.1109/MILCOM55135.2022.10017532)

[4] B. Fraser, S. Al-Rubaye, S. Aslam, and A. Tsourdos, "Enhancing the security of unmanned aerial systems using digital-twin technology and intrusion detection," in *2021 IEEE/AIAA 40th DASC*, pp. 1-10, 2021.

(https://doi.org/10.1109/DASC52595.2021.9594321)

[5] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Federated learning-based cognitive detection of jamming attack in flying adhoc network," *IEEE Access*, vol. 8, pp. 4338-4350, 2020. [Online] Available: https:// ieeexplore.ieee.org/document/8945183.

[6] S. O. Ajakwe, V. U. Ihekoronye, D.-S. Kim, and J.-M. Lee, "Alien: Assisted learning invasive encroachment neutralization for secured drone transportation system," *Sensors*, vol. 23, no. 3, 2023, ISSN: 1424-8220. [Online] Available: https://www.mdpi.com/1424-8220/23/3/1233.
(https://doi.org/10.3390/s23031233)

[7] E. M. Campos, et al., "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Netw.*, vol. 203, p. 108661, 2022, ISSN: 1389-1286. [Online] Available: https://www.sciencedirect.com/science/article/pii/S1389128621005405.
(https://doi.org/10.1016/j.comnet.2021.108661)

[8] J. Tursunboev, Y.-S. Kang, S.-B. Huh, D.-W. Lim, J.-M. Kang, and H. Jung, "Hierarchical federated learning for edge-aided unmanned aerial vehicle networks," *Applied Sci.*, vol. 12, no. 2, 2022, ISSN: 2076-3417. [Online] Available: https://www.mdpi.com/2076-3417/12/2/670.
(https://doi.org/10.3390/app12020670)

[9] S. Agrawal, et al., "Federated learning for intrusion detection system: Concepts, challenges and future directions," *CoRR*, vol. abs/2106.09527, 2021. [Online] Available: https://arxiv.org/abs/2106.09527.

[10] M. A. P. Putra, A. R. Putri, A. Zainudin, D.-S. Kim, and J.-M. Lee, "ACS: Accuracy-based client selection mechanism for federated industrial IoT," *Internet of Things*, vol. 21, p. 100657, 2023, ISSN: 2542-6605. [Online] Available: https://www.sciencedirect.com/science/article/pii/S254266052200138X.
(https://doi.org/10.1016/j.iot.2022.100657)

[11] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, vol. 8, pp. 217463-217472, 2020.
(https://doi.org/10.1109/ACCESS.2020.3041793)

[12] C. Xu, S. Liu, Z. Yang, Y. Huang, and K.-K. Wong, "Learning rate optimization for federated learning exploiting over-the-air computation," *IEEE J. Sel. Areas in Commun.*, vol. 39, no. 12, pp. 3742-3756, 2021.
(https://doi.org/10.1109/JSAC.2021.3118402)

[13] A. Belenguer, J. Navaridas, and J. A. Pascual, *A review of Federated Learning in Intrusion Detection Systems for IoT*, 2022. DOI: 10.48550/ARXIV.2204.12443. [Online] Available: https://arxiv.org/abs/2204.12443.

[14] O. Bouhamed, O. Bouachir, M. Aloqaily, and I. A. Ridhawi, "Lightweight IDS For UAV networks: A periodic deep reinforcement learning-based approach," in *2021 IFIP/ IEEE Int. Symp. Integrated Network Manag. (IM)*, 2021, pp. 1032-1037. [Online] Available: https://ieeexplore.ieee.org/document/9463947.

[15] L. Kou, S. Ding, T. Wu, W. Dong, and Y. Yin, "An intrusion detection model for drone communication network in SDN environment," *Drones*, vol. 6, no. 11, 2022, ISSN: 2504-446X. [Online] Available: https://www.mdpi.com/2504-446X/6/11/342.
(https://doi.org/10.3390/drones6110342)

[16] R. A. Ramadan, A.-H. Emara, M. Al-Sarem, and M. Elhamahmy, "Internet of drones intrusion detection using deep learning," *Electronics*, vol. 10, no. 21, 2021, ISSN: 2079-9292. [Online] Available: https://www.mdpi.com/2079-9292/10/21/2633.
(https://doi.org/10.3390/electronics10212633)

[17] V. U. Ihekoronye, S. O. Ajakwe, D.-S. Kim, and J. M. Lee, "Cyber edge intelligent intrusion detection framework for UAV network based on random forest algorithm," in *2022 13th Int. Conf. ICTC*, pp. 1242-1247, 2022.
(https://doi.org/10.1109/ICTC55196.2022.99524

00)

[18] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, "Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis," *IEEE Access*, vol. 9, pp. 138509-138542, 2021. (https://doi.org/10.1109/ACCESS.2021.3118642)

[19] F. Elena, N. Evgenia, and S. Anton, "Comparative review of the intrusion detection systems based on federated learning: advantages and open challenges," *Algorithms*, vol. 15, no. 7, p. 247, 2022. [Online] Available: https://doi.org/10.3390/a15070247. (https://doi.org/10.3390/a15070247).

[20] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "Federated learning for drone authentication," *Ad Hoc Netw.*, vol. 120, no. C, 2021, ISSN: 1570-8705. [Online] Available: https://doi.org/10.1016/j.ad hoc.2021.102574 (https://doi.org/10.1016/j.adhoc.2021.102574)

[21] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas in Commun.*, vol. 39, no. 10, pp. 3144-3159, 2021. [Online] Available: https://ieeexplore.ieee.org/document/ 9453811. (https://doi.org/10.1109/JSAC.2021.3088655)

[22] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "IoTDefender: A federated transfer learning intrusion detection framework for 5G IoT," in *2020 IEEE 14th Int. Conf. BigDataSE*, pp. 88-95, 2020. DOI: 10.1109/BigDataSE50710.2020.00020.

[23] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Phys. Commun.*, vol. 42, p. 101157, 2020, ISSN: 1874-4907. [Online] Available: https://www.sciencedirect.com/science/article/pii/S1874490720302342. (https://doi.org/10.1016/j.phycom.2020.101157)

[24] A. K. Chathoth, A. N. Jagannatha, and S. Lee, "Federated intrusion detection for IoT with heterogeneous cohort privacy," *arXiv preprint arXiv:2101.09878*, 2021.

[25] V. Rey, P. M. S. Sánchez, A. Huertas Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Comput Netw.*, vol. 204, p. 108693, 2022, ISSN: 1389-1286. [Online] Available: https://www.sciencedirect.com/science/article/pii/S1389128621005582. (https://doi.org/10.1016/j.comnet.2021.108693)

[26] C. Zheng, et al., "TiFL: A tier-based federated learning system," *CoRR*, vol. abs/2001.09249, 2020. [Online] Available: https://arxiv.org/abs/2001.09249.

[27] Y. Li, T.-H. Chang, and C.-Y. Chi, "Secure federated averaging algorithm with differential privacy," in *2020 IEEE 30th Int. Wkshp. MLSP*, pp. 1-6, 2020. (https://doi.org/10.1109/MLSP49062.2020.9231531)

[28] J. Xiao, C. Du, Z. Duan, and W. Guo, "A novel server-side aggregation strategy for federated learning in Non-IID situations," in *2021 20th ISPDC*, pp. 17-24, 2021. (https://doi.org/10.1109/ISPDC52870.2021.9521631)

[29] S. Ek, F. Portet, P. Lalanda, and G. Vega, "Evaluation of federated learning aggregation algorithms: Application to human activity recognition," in *UbiComp-ISWC'20,* pp. 638-643, 2020, ISBN: 9781450380768. [Online] Available: https://doi.org/10.1145/3410530.3414321.

[30] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020. [Online] Available: https://arxiv.org/abs/2007.14390.

[31] H. Vaisocherova, I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, 2016, ISSN: 1687-725X. [Online] Available: https://

doi.org/10.1155/2016/4731953.
(https://doi.org/10.1155/2016/4731953)

[32] G. G. Prabhugaonkar, X. Sun, X. Wang, and J. Dai, "Deep IoT monitoring: Filtering IoT traffic using deep learning," in *SVCC 2022, CCIS*, L. Bathen, et al., vol. 1683, Cham: Springer Nature Switzerland, pp. 120-136, 2023, ISBN: 978-3-031-24049-2.
(https://doi.org/10.1007/978-3-031-24049-2_8)

Vivian Ukamaka Ihekoronye

October 2014 : B.Tech. Information Management Technology, Federal University of Technology, Owerri, Nigeria.
2016~2021 : IT Analyst Donem Cargo Ltd, La-gos, Nigeria
2021~Current : Full-time researcher and Graduate Scholar Networked System Laboratory, IT-Convergence Engineering, Kumoh National Institute of Technology, South Korea.
<Research Interests> Cybersecurity, UAV Network Security, Optimization of AI models.
[ORCID:0000-0002-5088-0339]

Cosmas Ifeanyi Nwakanma

May 2005 : B.Eng. Electrical/ Electronics Engineering, Federal University of Technology, Owerri, Nigeria
Oct. 2012 : M.Sc. In-formation Technology, Federal University of Technology, Owerri, Nigeria
Feb. 2016 : MBA Project Management Technology, Federal University of Technology, Owerri, Nigeria
Feb. 2022 : Ph.D. IT-Convergence Engineering, Kumoh National Institute of Technology, Korea
Apr. 2009~Feb 2022 : Lecturer, Department of Information Technology, Federal University of Technology, Owerri, Nigeria
Mar. 2022~Current : Postdoctoral Research Fellow, Kumoh National Institute of Technology, Korea
<Research Interests> Explainable AI, Metaverse, Intrusion detection, Smart IoT Applications, Communication Engineering.
[ORCID:0000-0003-3614-2687]

Dong-Seong Kim

2003 : Ph.D. Electrical and Computer Engineering, Seoul National University, Korea.
2003~2004 : Postdoctoral researcher, Cornell University, NY, USA
2007~2009 : Visiting Professor, The University of California, Davis, CA, USA
2004~Current : Professor, Kumoh National Institute of Technology (KIT), Gyeongbuk, Korea
2014~Current : Director, ICT Convergence Research Center, KIT, Gyeongbuk, Korea
2017~2022 : Dean, Industry-Academic Cooperation Foundation and Office of Research (ICT), KIT, Gyeongbuk, Korea
2022~Current :CEO, NSLab co. Ltd., Korea
<Research Interests> Blockchain, Metaverse, Industrial IoT, real-time systems, industrial wireless control network, 5G+, and 6G.
[ORCID:0000-0002-2977-5964]

Jae Min Lee

2005 : Ph.D. Electrical and Computer Engineering, Seoul National University, Seoul, Korea

2005~2014 : Senior Engineer, Samsung Electronics Engineering, Su-won, Korea

2015~2016 : Principal Engineer, Samsung Electronics Engineering, Suwon, Korea

2017~Current : Associate Professor, School of Electronic Engineering, Kumoh National Institute of Technology, Gyeongbuk, Korea

<Research Interests> Blockchain, TRIZ, Smart IoT convergence Application, industrial wireless control network, UAV, Metaverse.

[ORCID:0000-0001-6885-5185]