Detection of Cyberattacks in an Software-Defined UAV Relay Network

Dennis Agnew Alvaro del Aguila Janise McNair
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL
dennisagnew@ufl.edu, delaguila.alvaro@hotmail.com, mcnair@ece.ufl.edu

***Abstract*—Unmanned aerial vehicles (UAVs), e.g., drones, have become crucial assets in the military's fleet of vehicles. UAVs can provide limited bandwidth for tactical communications and can act as relays over battlefields. Modern drones provide much higher bandwidth with dynamic antenna capabilities that would be useful in communicating around obstacles, such as urban corridors formed by rows of tall buildings that limit terrestrial lines of sight and attenuate high frequencies. While it is still more likely that one UAV is used for this purpose, a well-managed cluster of UAVs could increase the functionality of the entire terrestrial-drone network. Software-defined wireless networking (SDWN) is recognized as an effective way to manage distributed wireless networks. This paper proposes to use software-defined UAV networks (SD-UAV) to provide well-coordinated, secure communication resources and relaying capabilities to on-the-ground soldiers, military vehicles, and assets in an urban, signal-challenged environment. A mobility and packet delivery analysis is performed to determine the flow of packets through the simulated network, and, to maintain secure communication, a multi-cyberattack detection model is proposed to defend against jamming, black hole, and gray hole attacks using the Light Gradient Boosting (LightGBM) machine learning (ML) algorithm. Results show our model can provide an average of greater than 98% detection accuracy, precision, recall, and F1-scores.**

***Keywords*—software-defined networking (SDN), cybersecurity, UAVs, machine learning**

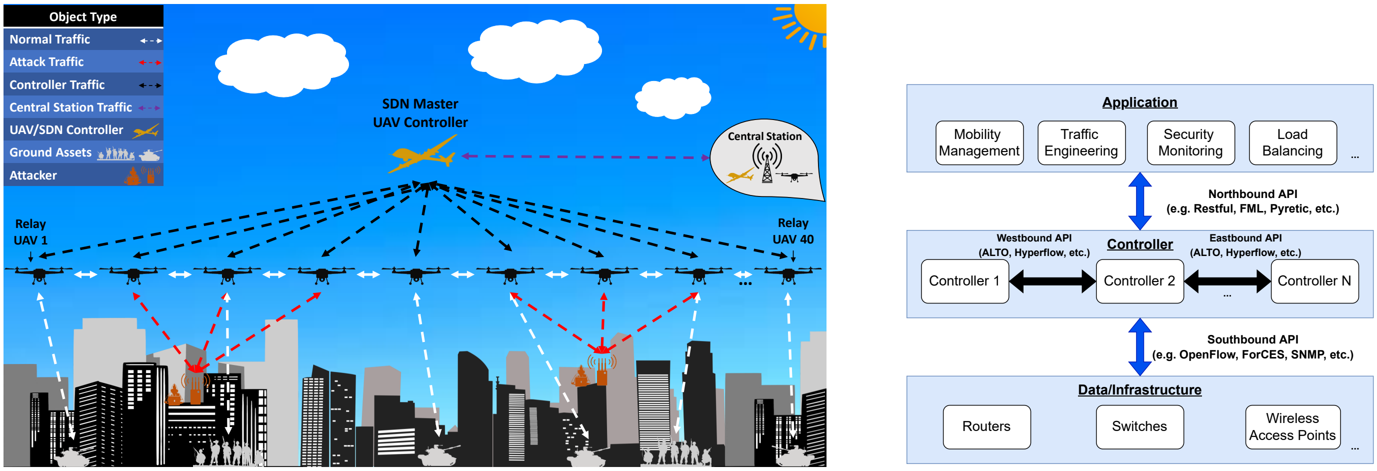
I. Introduction

Unmanned Aerial Vehicles (UAVs), or drones, have provided critical utility to both commercial and military applications such as communication relay, surveillance, network mobilization, and disaster relief, and the use of UAVs for a variety of applications, including defense, is expected to grow 25 to 75% by 2035 [1]. The military, including the Navy, Army, and Air Force, have used UAVs as relay nodes in tactical deployments to improve the quality of communication in varying geography [2]. UAVs are also being researched for ways to increase communications performance in areas where radio signals suffer from attenuation when passing through terrain obstacles such as mountains, or buildings [3], an increasing concern, as shown recently in Ukraine. Starlink's low earth orbit (LEO) constellation, has provided the region with stable communications, but a commercial, proprietary service can sometimes be a limitation for military operations [4].

Recent studies have explored the capabilities of various types of UAVs, including drones [5]. Although the capabilities of UAVs continue to improve, they still have a limited range and operational energy, and thus stay airborne for a limited amount of time. More often, a single UAV is operated due to effective control and mobility. To provide a cluster of UAVs as a communications network, mechanisms are needed to manage and control the network in a dynamic and adaptive approach. For tactical deployments of UAV clusters, software-defined networking (SDN) can be used to manage the dynamic behavior and capabilities. SDN is a networking concept that decouples the data plane and control plane of networking forwarding devices (i.e. switches and routers) and consolidates network control within one or more devices called SDN controllers. It allows for greater management/oversight, visibility, and security compared to traditional, or legacy, networking approaches [6]. Researchers have investigated using software-defined UAVs (SD-UAVs) and satellite communication (SATCOM) to supplement battlefield communications with relay networks, multi-path tcp solutions, or autonomous configurations of UAV swarms over disaster relief areas, e.g., [7]–[9]. Ongoing efforts of the military [10], [11] use UAVs for relay networks. However, these efforts do not consider SD-UAV networks in an urban environment nor provide mention of security against jamming, black hole, and gray hole attacks by malicious forces.

This paper proposes to use software-defined UAV networks (SD-UAV) to provide well-coordinated, secure communication resources and relaying capabilities to on-the-ground soldiers, military vehicles and assets in an urban, signal-challenged environment. A mobility and packet delivery analysis is performed to determine the flow of packets through the simulated network, and, to maintain secure communication, a multi-cyberattack detection model is proposed to defend against jamming, black hole and gray hole attack using the Light Gradient Boosting (LightBGM) machine learning (ML) algorithm. Few papers have proposed effective SD-UAV network architecture with cyber attack protection, much less a machine learning-based multi-cyberattack detection and identification mechanism. This paper is organized as follows. Section II presents a brief overview of SDN, followed by a discussion of the types of cyberattacks, envisioned attack scenarios, and the SD-UAV envisioned architecture. Section III contains our analysis of machine learning and network performance

'DISTRIBUTION STATEMENT A. Approved for public release: distribution is unlimited.' This work was supported by the National Science Foundation under Grant Number 1738420 and by the University of Florida/Harris Corporation Excellence in Research Fellowship.



(a)

(b)

Fig. 1: (a) Envisioned SD-UAV Network Deployment in an Urban Environment (b) General SDN Architecture [12]

statistics. Section IV provides a discussion of SD-UAV traffic and the creation of the datasets necessary to train our ML model for detection of the cyberattacks. Section V discusses the results of our ML model classification of cyberattacks with average accuracy, precision, recall, and F1-scores above 98%. Lastly, Section VI concludes the paper.

II. System Architecture

A. Software-Defined Networks (SDN)

As mentioned previously, SDN is a networking concept that decouples the data plane and control plane of networking forwarding devices (i.e. switches and routers) and consolidates network control within one or more devices called SDN controllers. The programmability of network devices makes the concept and practice of SDN appealing to various applications. As shown in figure 1(b), SDN can be described along three planes: (1) **Application Plane:**, which addresses SDN applications for network administration, policy implementation, and security services; (2) **Control Plane:**, which operates the network operating system and provides hardware abstractions to SDN applications. The controller can set up flows, a set of instructions followed by a series of packets in the data plane; (3) **Data Plane:** A collection of components used to forward traffic in response to instructions from the control planes.

As shown in Figure 1(b), the data/infrastructure layer includes routers, switches, and access points. In the case of SD-UAV, UAVs, ground stations, and possibly LEO satellites, information is transmitted between SDN architecture planes utilizing application programming interfaces (APIs). The controller employs southbound APIs such as OpenFlow [13], to interact with the data plane. Multiple controllers can communicate via Westbound and Eastbound APIs, such as ALTO [14]. The topmost layer is the application plane. The network operator may implement functional applications for mobility management, access control, energy efficiency, and/or security management at this layer. The application layer employs northbound APIs such as FML [15] to communicate with the control layer. These APIs can be used to communicate required modifications to the control layer, allowing the controller to make the required data/infrastructure layer adjustments.

B. Attack Scenario and SD-UAV System Architecture

A variety of denial-of-service (DoS) attacks such as jamming, black hole, and gray hole attacks may disrupt the communication of the SD-UAV network. We define the categories of these attacks as follows:

- **Jamming Attack:** Disruption or inhibition of wireless communication between nodes by transmitting a high-level radio frequency at the same frequency as the nodes.
- **Black Hole Attack:** A hijacked drone is instructed to drop packets which can loss of communication between endpoints in the network.
- **Gray Hole Attack** A hijacked drone is instructed to drop packets at variable rates, resulting in long-term QoS degradation.

Figure 1(a) illustrates our anticipated attack scenario. We assume the attackers use jammers and small antennas aimed at the relay UAVs. Disruptions due to attacks are modeled by changes in the inter-arrival time, transmission delays, packets received, and packets sent outlined in Section III-A. Observed behavior from these attacks are used to train a machine-learning model to detect and identify such attacks, as discussed in Section V.

C. SD-UAV System Architecture

Figure 1(a) depicts our envisioned architecture that consists of 40 relay drones, 1 master controller/drone, and 1 central station (CS) in an urban battlefield. At the central station, the network operator can monitor the network and launch additional drones. Each drone's buffer acts as a non-preemptive dual-priority class-based queue as explained in Section III-A. Packets arriving at a node from the master controller have a priority tag of 1 and a separate queue with greater priority than the regular queue of packets that have a label of 2. Non-preemptive means the priority packet only skips the line of the regular queue after the UAV has completed the service of its current packet. Prioritizing command traffic enables faster network reconfiguration and faster mitigation response to attacks.

As shown in Figure 1(a), the forwarding drones hover above the buildings to provide an overhead relay forwarding node for ground assets. The master controller oversees a global view of the network for the forwarding drones and installs flows in the forwarding UAVs to successfully route traffic to the cluster's required destination. In the event that the master controller fails, the relay drones will revert to legacy ad-hoc routing until a new master controller can be deployed from the CS. In future work, we will explore distributed and hierarchical arrangements of duplicate master controllers.

The CS is responsible for monitoring network traffic and implementing machine learning techniques to detect possible cyberattacks within the network. When an attack is detected, the CS communicates with the master controller and instructs it to command the relay drones to isolate the compromised UAV drones and reroute traffic around these drones until these devices can be further investigated. To accomplish this task, first the master controller pulls network statistics from the relay drones and sends this information back to the CS for processing and labeling to train the machine learning model so that it will be able to detect future cyberattacks.

III. Analysis

The network statistics collected by the master controller will include inter-arrival time (IAT), transmission delay (TD), packets sent, packets received, priority labels, etc. In this Section, we discuss analysis of the network model and performance statistics.

A. Network Performance Statistics

In the non-preemptive dual-class priority queue, each UAV represents the M/M/c queue [16], i.e. c ≥ 1, in which packet arrival is governed by a Poisson process and packet service time is governed by an exponential distribution, e.g., as in [17]. We denote the priority class as subscript 1 and the lesser priority class as subscript 2. The following equations describe the volume or intensity of traffic, p , for both classes:

$$p_1 \equiv \frac{\lambda_1}{\mu}, \quad p_2 \equiv \frac{\lambda_2}{\mu}, \quad p \equiv p_1 + p_2 = \frac{\lambda}{\mu} \tag{1}$$

The arrival rate of the packets is represented by λ and the service rate of the packets is represented by μ. The inter-arrival time (IAT) is the time difference (Δt) between packet arrivals and has an exponential distribution with the parameter λ. The probability density function is defined as follows for t $\geq$0:

$$f(t) = \lambda e^{\lambda t}. \tag{2}$$

The average IAT for both classes is defined as

$$IAT = \frac{1}{\lambda} \tag{3}$$

The service time follows an exponential distribution with parameter μ. The probability density function is as follows:

$$g(s) = \mu e^{-\mu s}, \forall \geq 0 \tag{4}$$

where $\frac{1}{\mu}$ is the average service time of the system. Utilizing Little's theorem, the total waiting time is defined as transmission delays (TD), and represented as the following for the two classes:

$$W_i = TD_i = \frac{1}{\mu_i - \lambda_i} \tag{5}$$

Let L_q^i denote the average number of packets for each class in the queues can be represented in equations 6 and 7 and the average total number is defined in equation 8:

$$L_q^1 = \frac{\lambda_1 p}{\mu - \lambda_1} \tag{6}$$

$$L_q^2 = \frac{\lambda_2 p}{(\mu - \lambda_1)(1-p)} \tag{7}$$

$$L_q = \frac{p^2}{1-p} \tag{8}$$

where $p = \frac{\lambda_1}{\mu} + \frac{\lambda_2}{\mu}$. Therefore, second-priority packets wait in a queue of UAVs longer than first-priority customers when $p < 1$ as shown in equation 9:

$$W_q^2 = \frac{p}{(\mu - \lambda_1)(1-p)} = \frac{p/(\mu - \lambda_1)}{1-p} = \frac{W_q^1}{1-p} > W_q^1 \tag{9}$$

The probability of observing a given number of packet arrivals in a period from [0,T] determined the normal distribution of network packet arrivals (i.e., non-attacked packets) within each system. This equation is utilized to model the bus's traffic volume:

$$P(n\ arrivals\ in\ interval\ T) = \frac{(\lambda T)^n e^{-\lambda T}}{n!} \tag{10}$$

where T is the IAT, and n represents the number of packets. The packet count (PC) is modeled as the following:

$$PC = \lambda T \tag{11}$$

B. Machine Learning Statistics

We employ accuracy, precision, recall, and F1-score to assess the performance of the LightGBM algorithm for attack classification. We compute these metrics utilizing the following: true negative (TN), true positive (TP), false negative (FN), and false positive (FP). TN refers to the predicted and actual negative sample for a given class. TP refers to a predicted positive and positive sample for a class. FN denotes a sample that was predicted to be negative but turned out to be positive for a given class. Lastly, FP refers to a sample that was predicted to be positive for a class but turned out to be negative. Keerthiraj et al. [18] define accuracy, precision, recall, and F1-score as follows:

Accuracy is the ratio of samples correctly predicted to the total number of samples. It is defined as follows:

$$Accuracy = 100 \times \frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

Precision is the ratio of correctly predicted instances to the total number of positively predicted instances. A high precision indicates that the model has a low false positive rate, indicating that the model's positive predictions tend to be correct. Precision is defined as follows:

$$Precision = 100 \times \frac{TP}{TP + FP} \tag{13}$$

Recall is the ratio of correctly predicted instances to all correct instances. A high recall indicates that the model has a low false positive rate, indicating that the majority of positive instances are correctly identified. It is defined as follows:

$$Recall = 100 \times \frac{TP}{TP + FN} \tag{14}$$

The F1-score quantifies the harmonic mean of precision and recall. A high F1-score indicates that both the precision and recall of the data are high. In addition, F1-score is valuable for imbalanced datasets, such as the one used to train our machine learning model, which is discussed in greater detail in Section V. F1-Score is defined as follows:

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{15}$$

IV. Simulation

A. Mobile Connection and Routing Simulation

Simulations have been implemented using Mininet-Wifi and the Open Network Operating System (ONOS) controller from our previous work [5], using the principles of reactive forwarding, the shortest path algorithm, and a code modification, to create the shortest possible paths between the hosts on demand. In addition, the routing algorithm favors lower energy routes to preserve drone batteries, as well as routes that avoid bottlenecks in the network.

The simulation scenario is an area of 1 square mile with 80 mobile devices that may connect to the SDN-UAV network through 40 drones or access points (APs). In Figure 2, the numbered dots from 1 to 80 represent the mobile devices, and the circles numbered from 81 to 120 represent the drones, each with a 120m radius range of network support.

Figure 2 (a), (b) and (c) shows the dynamic paths that can be created in the network over time as the drones move in response to the impact of the combat zones. In the simulations, performing a pingall between all the hosts, reveals the level of connectivity. When the mobile devices and drones are moving, there is a slight increase in the delay to create paths, as well as the packet loss, due to some drones and hosts not being connected at various moments. However, the drone configuration adjusts and regains connectivity dynamically.

When the number of drones is increased, Table I demonstrates that the level of disruption and routing delay increases. We tested increasing the number of drones to 50, which created 14% more packet loss and a 14-fold to 28-fold increase in delay. This may be due to there being congestion or saturation at the controller. It will be key to evaluate the optimal number of drones each type of area.

TABLE I: Comparison of Path Establishment and Packet Loss for Different Emulated Scenarios

Scenario	**Longest Time to Create a Path**	**% of Packages Lost**
40 Drones Connected to all the Hosts	1 ms	0%
40 Drones with Random Movement	2 ms	14%
50 Drones with Random Movement	28 ms	19%

TABLE II: Simulation Parameters for each Class

Class	**Port Rate (KBps)**	**Queue Size (KB)**
Normal	10	100
Jamming	0.5	100
Black Hole	10	0.005
Gray Hole	10	0.03

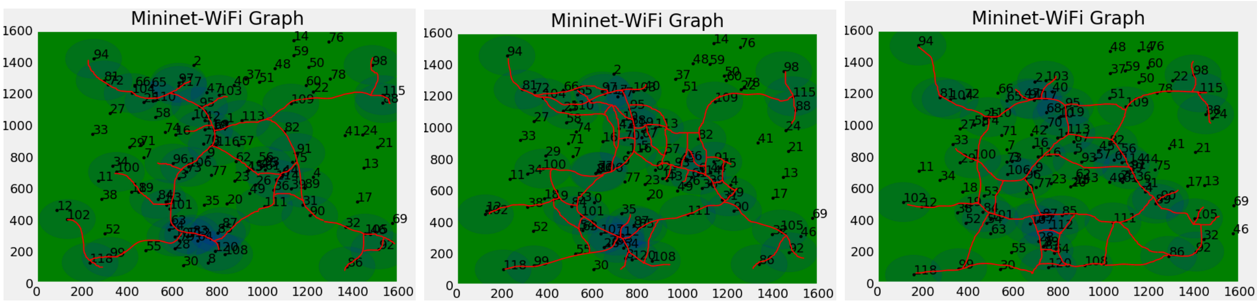
Figure 3 shows the network packet flow, with the 40 drones, as captured by WireShark. The peaks represent the number of packets being sent. The proposed system architecture, with an effective choice in the number of drones, results in a very small amount of packet loss.

B. Simulation of Network Traffic under Cyber Attack

In addition to testing the effectiveness of our architecture with our routing scheme, we tested the effectiveness of our cyberattack defense mechanisms. The SD-UAV network traffic was tested using SimComponents [19], a network traffic simulation program built using the SimPY process-based discrete event simulation framework. We modified the original open source code of SimComponents to incorporate a non-preemptive priority M/M/C queuing model as explained in Section III-A. We modelled our simulation after the UHF/VHF radio traffic often used during military operations. The packet inter-arrival times and packet transmission delays, based on the military UAV wireless communication standards STANAG 4586, MIL-STD-6016, and related research, were achieved by altering the port rate and queue size parameters of the servers as shown in Table II. In addition, we added 5% Gaussian noise to the dataset after generation to increase data variability for training and testing.

We completed a 1-hour network traffic simulation for the 40 UAVs described in Section II-B on a Linux server running Redhat Enterprise 8.6 with an Intel® 5th Gen Core$^{\text{TM}}$ i5-6500 CPU @ 3.20GhZ. The network traffic consists of the following classes: normal, jamming, black hole, and gray hole. We consider our normal class as our non-attack traffic while the others are considered to be attack traffic. Our attack traffic makes up $\sim 15\%$ of our dataset with each attack class accounting for $\sim 5\%$ of the total dataset. The goal is for the CS that monitors the overall SD-UAV network traffic to be able to detect abnormalities and identify the beginning of the aforementioned attacks.

To simulate these varied attacks, we created three different attack classes based on the behavior of the attack. To mimic



(a)

(b)

(c)

Fig. 2: Example UAV topology in Mininet-WiFi for 80 hosts and 40 drones (APs), with X and Y axes scaled in meters. Dynamic Network Paths (a) Step 1 (b) Step 2 (c) Step 3

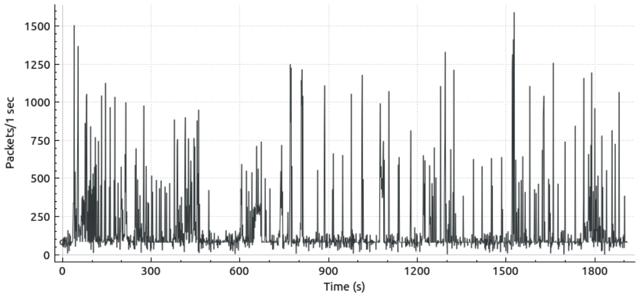


Fig. 3: Packet Flow within the UAV Network with 80 hosts and 40 drones (APs)

the behavior of a jamming attack on a node, we limited the throughput of an affected node by limiting the port-based rate to create a throughput lost 95%. For the black hole attack class, we assume the attacker has successfully hijacked a node and begun to instruct the node to drop packets. Therefore, we limit the queuing buffer of a targeted node by $> 95\%$ which causes it to drop packets when its limited queue is full. To simulate the gray hole attack, we assume the node is hijacked as well but the attacker is aiming to disrupt QoS. Hence, we instruct the node to limit the buffer size randomly by $70\% - 75\%$ to create a variable packet drop.

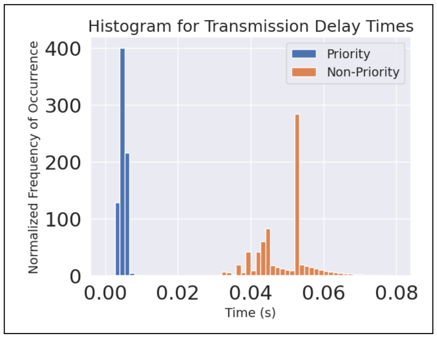


Fig. 4: Histogram for Normal Traffic Transmission Delays

From the data and simulation, we create five 3600 rows x 40 column CSV files for each of the following features: priority factor, inter-arrival time, transmission delay/latency, number of packets sent, and number of packets received. In our dataset, each row represents a time stamp for each second in an hour and each column represents a drone's communication for the entire duration. Figure 4 shows a normalized histogram of our transmission delay times for both priority and non-priority normal traffic. The priority traffic receives faster service times (μ) than non-priority traffic. (Priority traffic accounts for 8.5% of the total dataset.) Our dataset was then processed and prepared for our LightGBM algorithm which is explained in the next section.

V. Numerical Results

To detect, identify, and classify jamming, black hole, and gray hole attacks, we utilize the LightGBM machine learning algorithm. LightGBM was chosen over other comparable models, e.g., extreme gradient boost (XGBoost) and categorical boost (CatBoost) due to faster training and prediction times [20]. We apply k-fold cross-validation (k=5) to our LightGBM output to validate our results. The average and standard deviation of the accuracy, precision, recall, and F1-score of the five folds are shown in Table III. The results show that our model was able to correctly classify the network traffic on average at greater than $> 98\%$. Thus, using our model, the CS would be able to detect the cyber attacks present in the SD-UAV network.

TABLE III: Average Performance Results for k-fold Cross Validation (k=5) for Normal and Attack Traffic Classes

Algorithm	**Avg. Accuracy** $\mu \pm \sigma$	**Avg. Precision** $\mu \pm \sigma$	**Avg. Recall** $\mu \pm \sigma$	**Avg. F1-score** $\mu \pm \sigma$
LightGBM	99.78 ± 0.02	99.03 ± 0.13	98.95 ± 0.13	**98.98 ± 0.13**

To further demonstrate the efficacy of our detection algorithm, we have included the output of k-fold cross-validation for each individual class in Table IV and the confusion matrix in Figure 5. As shown in Table IV, each individual class of network traffic achieved performance greater than $> 96\%$. The normal class achieved the best results overall due to the natural skew in the dataset since attack traffic, as aforementioned, only composed of only $\sim 15\%$. The gray hole attack class has the lowest overall performance because the algorithm had difficulty distinguishing from the other classes. This is due to the variable nature of gray hole attacks which makes them more difficult to detect than black hole attacks due to their similar network statistics. However, the LightGBM algorithm was still able to make the correct classification and classify gray hole attacks with an accuracy greater than $> 96\%$.

TABLE IV: Individual Performance Results for k-fold Cross Validation (k=5) for each Class

Class	Avg. Accuracy $\mu \pm \sigma$	Avg. Precision $\mu \pm \sigma$	Avg. Recall $\mu \pm \sigma$	Avg. F1-score $\mu \pm \sigma$
Normal	99.99 ± 0.01	99.91 ± 0.04	99.99 ± 0.01	**99.95 ± 0.02**
Jamming	98.29 ± 0.79	97.86 ± 1.07	98.29 ± 0.79	**98.07± 0.75**
Black Hole	99.31 ± 0.45	99.06 ± 0.73	99.31 ± 0.45	**99.18 ± 0.42**
Gray Hole	96.67 ± 0.82	98.63 ± 0.66	96.67 ± 0.82	**97.64 ± 0.41**

The results of our confusion matrix for the k-fold cross-validation shown in Figure 5 demonstrate the efficiency of our model in detecting and classifying these attacks in our dataset among the four classes. We contribute our model's success to the five features we extract from our simulation: priority, inter-arrival times, transmission delay, packets sent, and packets received.

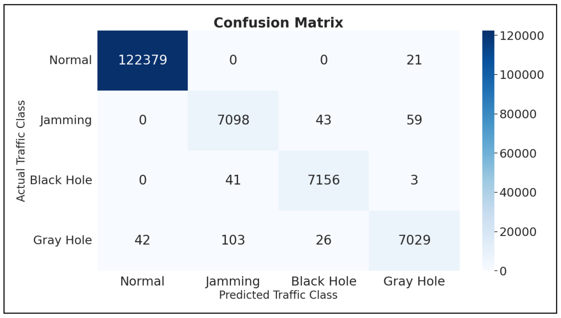


Fig. 5: Confusion Matrix of LightGBM ML Algorithm k-fold Cross Validation ($k = 5$) for the Multiple Traffic Classes

VI. Conclusion and Future Work

In this paper, we present an SD-UAV system for overcoming attenuation for ground assets in urban areas. In our Mininet-Wifi-simulated testbed, we find that one controller with 40 drones can be effective for a 1 mile squared zone and that higher drone counts have a negative impact on the longest time to establish paths and packet loss. In our simulations of network traffic under cyberattack, SimComponent was used to generate traffic using non-preemptive priority-based queues. This increases overall network security response since priority traffic, such as traffic from the controller, has shorter service times (μ) than non-priority traffic. Using data sets from our network traffic simulations, we trained the LightGBM machine learning algorithm. The results demonstrated average accuracy, precision, recall, and F1-scores $> 98\%$.

In future work, we plan to develop this framework further to incorporate a broader range of cyberattack detection such as botnets, false data injection, low-rate denial of service, etc. We also will develop mitigation techniques after attack detection to maintain network QoS while under attack.

References

[1] A. Hanscom and M. Bedford, "Unmanned aircraft system (uas) service demand 2015-2035," *Literature review & projections of future usage. Research and Innovative Technology Administration US Department of Transportation, Washington, DC, USA*, 2013.

[2] G. Djukanovic, D. N. Kanellopoulos, and G. Popovic, "Evaluation of a uav-aided wsn for military operations: Considering two use cases of uav," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 14, no. 1, pp. 1–16, 2022.

[3] M. Kryk, K. Malon, and J. M. Kelner, "Propagation attenuation maps based on parabolic equation method," *Sensors*, vol. 22, no. 11, p. 4063, 2022.

[4] M. Gongadze, "Spacex's starlink becomes crucial tool in ukrainian war effort," Jan 2023. [Online]. Available: https://www.voanews.com/a/spacex-s-starlink-becomes-crucial-tool-in-ukrainian-war-effort/6922510.html

[5] A. del Aguila, J. V. Mendoza, S. B. Mandavilli, and J. McNair, "Remote and rural connectivity via multi-tier systems through sdn-managed drone networks," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 956–961.

[6] S. H. Haji, S. R. Zeebaree, R. H. Saeed, S. Y. Ameen, H. M. Shukur, N. Omar, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, and H. M. Yasin, "Comparison of software defined networking with traditional networking," *Asian Journal of Research in Computer Science*, vol. 9, no. 2, pp. 1–18, 2021.

[7] Q. Zhao, P. Du, M. Gerla, A. J. Brown, and J. H. Kim, "Software defined multi-path tcp solution for mobile wireless tactical networks," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 1–9.

[8] Q. Zhao, A. J. Brown, J. H. Kim, and M. Gerla, "An integrated software-defined battlefield network testbed for tactical scenario emulation," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 373–378.

[9] B.-W. Chen and S. Rho, "Autonomous tactical deployment of the uav array using self-organizing swarm intelligence," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 52–56, 2020.

[10] K. D. Atherton, "Why the us army wants an "aerial tier network" for better communications," Jun 2022. [Online]. Available: https://www.popsci.com/technology/army-develops-aerial-tier-network/

[11] O. Media, "Military embedded systems," Dec 2022, https://militaryembedded.com/comms/satellites/laser-comms-for-network-relay-demonstrated-at-us-navy-facility-by-general-atomics.

[12] D. Agnew, N. Aljohani, R. Mathieu, S. Boamah, K. Nagaraj, J. McNair, and A. Bretas, "Implementation aspects of smart grids cyber-security cross-layered framework for critical infrastructure operation," *Applied Sciences*, vol. 12, no. 14, p. 6868, 2022.

[13] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[14] R. Alimi, R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, and R. Woundy, "Application-layer traffic optimization (alto) protocol," Tech. Rep., 2014.

[15] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker, "Practical declarative network management," in *Proceedings of the 1st ACM workshop on Research on enterprise networking*, 2009, pp. 1–10.

[16] D. Gross, *Fundamentals of queueing theory*. John Wiley & Sons, 2008.

[17] A. Baltaci, M. Klügel, F. Geyer, S. Duhovnikov, V. Bajpai, J. Ott, and D. Schupke, "Experimental uav data traffic modeling and network performance analysis," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[18] K. Nagaraj, S. Zou, C. Ruben, S. Dhulipala, A. Starke, A. Bretas, A. Zare, and J. McNair, "Ensemble corrdet with adaptive statistics for bad data detection," *IET Smart Grid*, vol. 3, no. 5, pp. 572–580, 2020.

[19] G. Bernstein, "Basic network simulations and beyond in python introduction." [Online]. Available: https://www.grotto-networking.com/DiscreteEventPython.html

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.