Buletin Ilmiah Sarjana Teknik Elektro
Vol. 5, No. 4, December 2023, pp. 505-514
ISSN: 2685-9572, DOI: 10.12928/biste.v5i4.9415

Adaptive Cyber-Defense for Unmanned Aerial Vehicles: A Modular Simulation Model with Dynamic Performance Management

Gregorius Airlangga
Information Systems Study Program, Universitas Katolik Indonesia Atma Jaya, Jakarta, Indonesia

ARTICLE INFORMATION

Article History:

Submitted 23 October 2023
Revised 27 November 2023
Accepted 07 December 2023

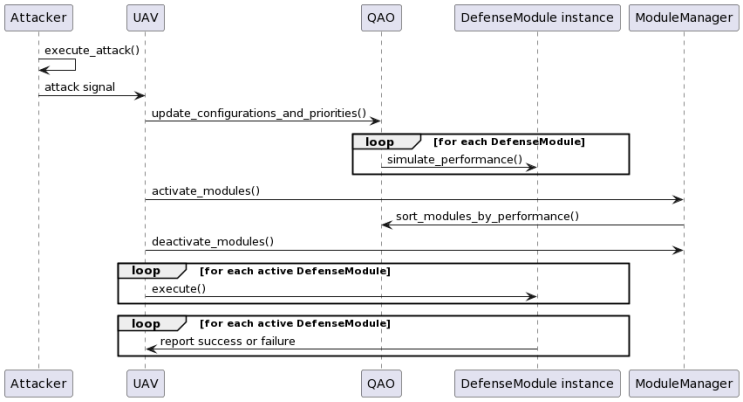
Keywords:

UAV;
Cyber Security;
Modular System;
Monolithic;
Multi UAV

Corresponding Author:

Gregorius Airlangga,
Universitas Katolik Indonesia Atma Jaya, Jakarta, Indonesia.
Email: gregorius.airlangga@atmajaya.ac.id

ABSTRACT



In light of escalating cyber threats, this study tackles the cybersecurity challenges in UAV systems, underscoring the limitations of static defense mechanisms. Traditional security approaches fall short against the sophisticated and evolving nature of cyber-attacks, particularly for UAVs that depend on real-time autonomy. Addressing this deficiency, we introduce an adaptive modular security system tailored for UAVs, enhancing resilience through real-time defensive adaptability. This system integrates scalable, modular components and employs machine learning techniques—specifically, neural networks and anomaly detection algorithm to improve threat prediction and response. Our approach marks a significant leap in UAV cybersecurity, departing from static defenses to a dynamic, context-aware strategy. By employing this system, UAV stakeholders gain the flexibility needed to counteract multifaceted cyber risks in diverse operational scenarios. The paper delves into the system's design and operational efficacy, juxtaposing it with conventional strategies. Experimental evaluations, using varied UAV scenarios, measure defense success rates, computational efficiency, and resource utilization. Findings reveal that our system surpasses traditional models in defense success and computational speed, albeit with a slight increase in resource usage a consideration for deployment in resource-constrained contexts. In closing, this research underscores the imperative for dynamic, adaptable cybersecurity solutions in UAV operations, presenting an innovative and proactive defense framework. It not only illustrates the immediate benefits of such adaptive systems but also paves the way for ongoing enhancements in UAV cyber defense mechanisms.

Document Citation:

G. Airlangga, "Adaptive Cyber-Defense for Unmanned Aerial Vehicles: A Modular Simulation Model with Dynamic Performance Management," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 505-514, 2023, DOI: 10.12928/biste.v5i4.9415.

1. INTRODUCTION

The rapid proliferation of unmanned aerial vehicles (UAVs) in a myriad of sectors, from agriculture to defense, signals a technological paradigm shift [1]-[3]. However, this expansion brings with it an escalating array of cybersecurity challenges. UAVs, increasingly embedded in critical operational frameworks, face a myriad of cyber threats, from data interception to system hijacking. Current research in UAV cybersecurity, though extensive, primarily gravitates towards static defense mechanisms, for example, UAVs face cyber threats such as GPS spoofing, which can misdirect navigation, and signal jamming, which can disrupt communications [4]-[6]. These methods, proficient in combating known threats, they are rapidly becoming obsolete in an environment where cyber threats are not just evolving but also becoming more nuanced and sophisticated [7]-[10]. This scenario highlights a critical gap in UAV cybersecurity research: the urgent need for dynamic, adaptive security models that can evolve in tandem with the shifting landscape of cyber threats [11][12]. The inherent limitation of the prevailing UAV cyber-defense strategies lies in their static, predefined nature. These traditional security systems, designed to counter known and predictable threats, are ill-equipped to adapt to the dynamic and unpredictable nature of modern cyber-attacks [13]-[15]. This static approach is particularly problematic in the context of UAV operations, which often require real-time, autonomous decision-making and data processing [16]-[19]. Furthermore, the increasing integration of UAVs into complex and sensitive operational domains amplifies the impact of any cybersecurity breach. The stark contrast between the evolving complexity of cyber threats and the static nature of existing defense mechanisms underscores a significant shortcoming in current UAV cyber-defense methodologies [20]-[21].

Recent advancements in UAV technology and cybersecurity have led to a growing recognition among researchers of the inadequacy of static defense systems. The state-of-the-art research is gradually shifting towards the development of more dynamic, intelligent cybersecurity solutions [22]-[24]. These emerging paradigms propose the use of advanced machine learning algorithms, real-time threat assessment, and adaptive response mechanisms [25]-[27]. However, there is a noticeable gap in the practical implementation of these theoretical frameworks, particularly in the unique context of UAV operations, which present distinct challenges such as limited computational resources, the need for rapid response to threats, and operational versatility. In addition, static defense mechanisms, while historically effective, now fall short in anticipating and mitigating the intricacies of modern cyber threats that are characterized by their dynamic and adaptive nature. This study seeks to bridge this gap by proposing a novel, adaptive modular security system specifically crafted for UAVs. Our approach is distinguished by its dynamic nature, enabling the system to adjust its defense mechanisms in real-time in response to changing cyber threats. The proposed system leverages a modular design, allowing for scalable and flexible integration of various defense components. Furthermore, it incorporates an element of machine learning, enabling the system to evolve by learning from past cyber incidents and enhancing its predictive capabilities.

Our research is positioned to make a pivotal contribution to the field of UAV cybersecurity. It addresses a critical need for dynamic, adaptable defense systems in the face of increasingly sophisticated cyber threats. For UAV manufacturers, operators, and cybersecurity experts, this study offers an advanced framework for enhancing UAV security. The adaptive nature of the proposed system marks a significant departure from traditional cybersecurity approaches, offering a more robust and responsive solution to protect UAVs in various operational environments. The paper is structured to provide a thorough understanding of the current state of UAV cyber-defense. It begins with an in-depth analysis of existing cybersecurity strategies, highlighting their limitations in the face of modern cyber threats. This is followed by a detailed presentation of our proposed adaptive modular security system, including its architectural framework, operational capabilities, and potential advantages over existing models. The methodology section outlines the simulation setup and the criteria used to evaluate the system's effectiveness. The results section presents a comprehensive analysis of the system's performance in various simulated cyber threat scenarios. The discussion section extrapolates these findings to real-world UAV operations, underscoring the practical implications and potential impact of the research. Finally, the paper concludes with a summary of key insights, a discussion on the limitations of the current study, and recommendations for future research directions in the rapidly evolving field of UAV cybersecurity.

2. LITERATURE SURVEY

The integration of Unmanned Aerial Vehicles (UAVs) in various domains has underscored the need for robust cybersecurity measures [28]. As UAVs become increasingly central to critical operations, the focus has shifted to developing adaptive cyber-defense systems. This literature survey provides an in-depth examination of the trajectory of UAV cybersecurity, highlighting the evolution from static models to adaptive, dynamic systems. It also critically assesses existing work in the field, identifying gaps and areas of innovation that our research aims to address. Early UAV cybersecurity research laid the groundwork for what would become a rapidly evolving field. Initial studies, such as those by [29][30] were pivotal in establishing basic security protocols. These studies focused on static defense mechanisms, providing a strong foundation but limited in

scope and adaptability. They were effective against known threats but lacked the flexibility to evolve with emerging cyber-attack strategies, a limitation that became increasingly apparent as the field progressed.

The realization that static models were insufficient against dynamic cyber threats marked a significant turning point in UAV cybersecurity research. Scholars like [31] began advocating for systems capable of responding to an ever-changing threat landscape. This shift was critical in acknowledging that cyber threats are not static entities but evolve continuously, necessitating a similar evolution in defense strategies. The development of modular simulation models in UAV cybersecurity was a response to the need for more flexible and adaptable defense systems. These models, explored in studies like those by [32], allowed for individual components of a cybersecurity system to be updated or modified independently. This modularity offered a significant advantage over monolithic systems, allowing for quicker adaptation to specific threats and more efficient overall system management. However, these models often faced challenges in integration and coordination among the modules, an aspect our research aims to refine.

Dynamic performance management emerged as a key feature of adaptive cyber-defense. Research by [33] focused on developing algorithms for continuously assessing and optimizing the performance of cyber-defense modules. This approach was a step forward in enabling real-time adjustments and proactive defense strategies. While these systems marked a significant advancement, there remained a gap in their ability to predict and manage performance under highly dynamic and unpredictable operational conditions. The integration of AI and machine learning into UAV cybersecurity has been a transformative development. The work from [34] highlighted the potential of these technologies in enhancing predictive capabilities and identifying novel threats. However, there is still a need for further development in the application of these technologies, particularly in the context of real-time decision-making and performance optimization under diverse operational scenarios.

The implementation of adaptive cyber-defense systems in UAVs is not without challenges, [35][36] pointed out the constraints of UAVs' computational resources, which limit the complexity of deployable cybersecurity solutions. Additionally, as [37][38] noted, these systems must be resilient to varying operational environments, a factor often overlooked in current models. This survey underlines the evolutionary path of UAV cybersecurity, from foundational static models to sophisticated, adaptive systems. It highlights that while significant progress has been made, there are still gaps and challenges to be addressed. Our research contributes to this evolving field by developing an advanced modular simulation model with dynamic performance management. This model not only addresses the integration and coordination challenges of modular systems but also enhances the predictive and real-time decision-making capabilities through the integration of AI and machine learning, all while being mindful of the resource constraints inherent to UAVs. Through this, we aim to push the boundaries of what's achievable in UAV cybersecurity, ensuring safer and more reliable UAV operations in an increasingly digital world.

3. PROPOSED SECURITY SYSTEM MODEL

3.1. Security System Model

In this simulation, the cyber-defense system is designed to protect an UAV from various types of cyber-attacks. To provide a comprehensive analysis, let us further dissect the system into its core components and their interactions, elucidating the mathematical concepts underlying the model. The attack types are defined as a set $A = \{a_1, a_2, \ldots, a_n\}$, where n denotes the total number of attack types and a is the sequence of attack types. Correspondingly, there exists a set $D = \{d_1, d_2, \ldots, d_n\}$ comprising defense modules designed to counter each type of attack. The system is designed such that each attack type $a_i \in A$ has a corresponding defense module $d_i \in D$. The performance of the defense modules is modeled by a function $p: D \rightarrow [0.5, 1]$, which assigns a performance score to each defense module d_i. The performance score of a module is not constant and varies throughout the simulation, mimicking real-world fluctuations in performance. To effectively manage the defense modules and prioritize their activation, a function $QAO: D \times p(D) \rightarrow D_{sorted}$ is defined. This function accepts the set of defense modules and their performance scores as input parameters, and it outputs the sorted set of defense modules, D_{sorted}, arranged in descending order of their performance scores. In addition to the QAO function, a module manager, denoted as MM, is employed. The module manager's role is to activate the top k defense modules based on their performance scores and deactivate the remaining n-k modules. By activating only, the most effective defense modules, the system aims to minimize the time and resources spent on handling cyber-attacks. At simulation start, the UAV initializes its defense orchestrator, setting the stage for real-time cyber-threat management. It maintains a success flag, denoted as $S \in Z$, which is initialized at $S = 0$. This flag represents the overall efficacy of the UAV's cyber-defense system, with a higher value signifying better performance in defending against cyber-attacks. Throughout the simulation, the

system probabilistically determines whether an attack will occur at each time step t. This is achieved by computing the probability $P(attack) = 0.1$. If an attack is indeed initiated, the attacker selects an attack type $a \in A$ and a duration d ∈ R for the attack, with the constraint that $d \in [1, 5]$. The UAV then updates the performance scores $p(D)$ of the defense modules using the QAO function. Consequently, the module manager MM activates the top k defense modules in D_{sorted} and deactivates the remaining n-k modules. The UAV iteratively attempts to execute the defense modules in MM against the ongoing attack. This process continues until one of the active modules successfully defends against the attack or all active modules fail. The success flag S is updated accordingly: if a defense module succeeds, $S \leftarrow S\ 1$; if all active defense modules fail, $S \leftarrow S - 1$. The final success flag, S, serves as an indicator of the overall performance of the UAV's cyber-defense system throughout the simulation.

A higher value of S implies that the system has been more successful in defending against cyber-attacks. This simulation models a cyber-defense system for an Unmanned Aerial Vehicle, taking into account various attack types, defense modules, and their performance scores. The model employs a module manager and a QAO function to prioritize and activate defense modules based on their performance, aiming to optimize the system's efficiency in defending against cyber-attacks. The success flag, S, provides a metric to assess the overall effectiveness of the cyber-defense system. In addition, the underlying mathematical model of the simulation can be further refined by examining the interactions between the attacker and the cyber-defense system.

The attacker can be represented as function $A: T \rightarrow A$, where T denotes the set of time steps in the simulation, and A represents the attack types. The function A maps each time step $t \in T$ to an attack type $a \in A$, with a probability $P(attack) = 0.1$. Moreover, the attacker assigns a duration d ∈ R to the selected attack, subject to the constraint that $d \in [1, 5]$. To enhance the system's adaptability, one may consider incorporating a learning mechanism that updates the performance scores $p(D)$ based on past experiences. In this regard, an online learning model L can be introduced, defined as $L: D \times p(D) \times H \rightarrow p'(D)$, where H represents the set of historical data comprising past attack types and defense module performance. The learning model L takes the current defense modules D, their performance scores $p(D)$, and the historical data H as input, and outputs the updated performance scores $p'(D)$. Upon executing the defense modules against an ongoing attack, the UAV can utilize the historical data H to better predict the attack type and select the most appropriate defense module. This approach enables the system to continually improve its defense capabilities based on past experiences. Furthermore, the success flag S can be normalized to obtain a metric that is invariant to the total number of attacks, facilitating a fair comparison between different simulation scenarios. The normalized success flag, S', can be computed as $S' = S / N$, where N denotes the total number of attacks that occurred during the simulation. The normalized success flag $S' \in [-1, 1]$, with $S' = 1$ indicating that the cyber-defense system successfully defended against all attacks, and $S' = -1$ signifying that it failed to defend against any attack.

3.2. Structure and Dynamics

The Attack class signifies an attack on the UAV system, possessing two attributes: the name, a string indicating the type of attack, and the duration, a float that signifies the length of the attack in seconds. The Attacker class embodies an entity that initiates attacks on the UAV system. It consists of a single method called execute_attack, which instantiates an object of the Attack class and returns it. The DefenseModule class rep-

resents a module capable of defending against a specific type of attack. It contains three attributes: defense_name, a string denoting the module's defense mechanism; problem_flag, a boolean that indicates any issues with the module; and performance, a float that describes the module's performance. Additionally, this class has two methods: simulate_performance for simulating the module's performance and execute for carrying out the defense against a given attack.

The QAO module is represented by the class, which manages the defense modules. It comprises two attributes: alpha, a float representing the learning rate, and modules, a list that holds instances of the DefenseModule class. The QAO class has three methods: add_module for adding a defense module to the list, update_configurations_and_priorities for modifying the configurations and priorities of the defense modules, and the module called sort_modules_by_performance for organizing the modules according to their performance. The ModuleManager class is responsible for managing and activating defense modules. It has one attribute, modules, which is a list containing instances of the DefenseModule class. The class has four methods: add_module for adding a defense module to the list, activate_modules for enabling the highest-performing defense modules, deactivate_modules for disabling the remaining modules, and utilizes for incorporating the QAO instance. The UAV system is represented by the UAV class. It has three attributes: qao, an instance of the QAO class; module_manager, an instance of the ModuleManager class; and success_flag, an integer indicating the number of successful defenses. The class contains one method,

execute_defense_mechanisms, which accepts an instance of the Attack class and implements the appropriate defense mechanisms against the attack. The relationships between these classes can be summarized as follows: The Attacker class generates instances of the Attack class, while the DefenseModule class defends against instances of the Attack class. The QAO class manages instances of the DefenseModule class, and the ModuleManager class oversees instances of the DefenseModule class and makes use of the QAO instance. Lastly, the UAV class collaborates with the QAO and ModuleManager classes and executes instances of the DefenseModule class.

The sequence diagram provides an in-depth portrayal of the intricate interplay between the classes involved in the UAV defense system, presenting an elaborate representation of the various components' collaboration as they work together to safeguard the UAV system. The subsequent description offers a thorough account of these interactions, detailing each step in the process and the communications between the classes as they strive to maintain the security and integrity of the UAV system. As the Attacker class initiates an attack on the UAV system by executing the execute_attack() method, it triggers a complex chain of events within the UAV system, specifically designed to counteract and mitigate the harmful effects of the attack. Upon receiving the attack signal, the UAV class promptly responds by initiating a series of coordinated defense mechanisms. The first course of action involves the UAV class interacting with the QAO class, calling upon it to update the configurations and priorities of the defense modules by invoking the update_configurations_and_priorities() method shown in Figure 1.

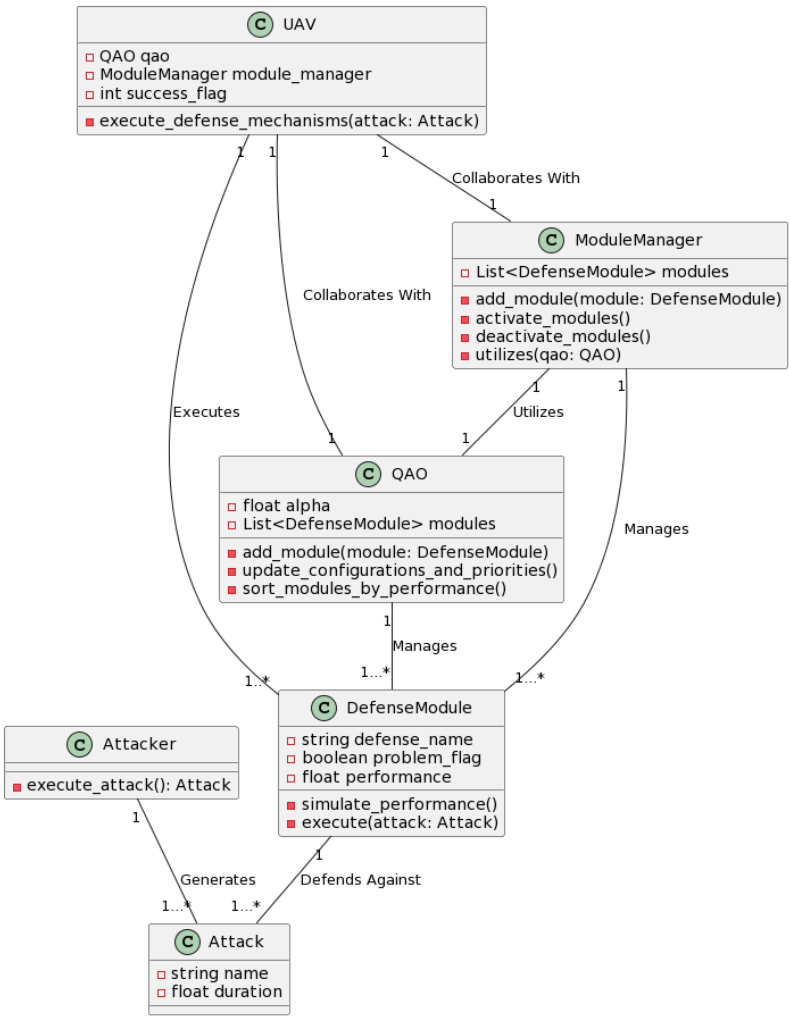


Figure 1. QAO class diagram

In order to response from the UAV class, the QAO class engages with each Defense Module instance contained within its modules list. By calling the simulate_performance() method for every module, the QAO

class updates the performance values and problem_flag attributes for each DefenseModule instance, thereby obtaining a dynamic and current evaluation of their potential efficacy. Armed with these performance updates, the UAV class proceeds to interact with the ModuleManager class, calling the activate_modules() method in order to activate the top-performing defense modules. In doing so, the ModuleManager enlists the assistance of the QAO class to sort the Defense Module instances based on their performance by invoking the sort_modules() method.

With the modules sorted, the ModuleManager activates the highest-ranked modules, adding them to its active modules list and preparing them for deployment as part of the defense effort. After the activation of the top-performing modules, the UAV class directs the ModuleManager to deactivate the lower-ranking modules by calling the deactivate_modules() method. Complying with the UAV's instructions, the ModuleManager removes these less effective modules from its active modules list, thereby streamlining the defense resources and optimizing the system's response to the attack. With the activation and deactivation processes for the defense modules now complete, the UAV class shifts its focus to the execution of the defense strategies themselves. To do this, the UAV class calls the execute() method on each of the active DefenseModule instances. These modules then attempt to thwart the attack utilizing their distinct defense mechanisms. Upon completion, the DefenseModule instances report the results of their endeavors, communicating the outcomes as either success or failure back to the UAV class. This sequence diagram emphasizes the sequence of events and interactions between the various classes that constitute the UAV defense system. By thoroughly examining the dynamic response to attacks and the cooperation among the system components, this explanation demonstrates the system's capacity to adapt and deliver successful defense outcomes in a constantly evolving threat environment shown in Figure 2.

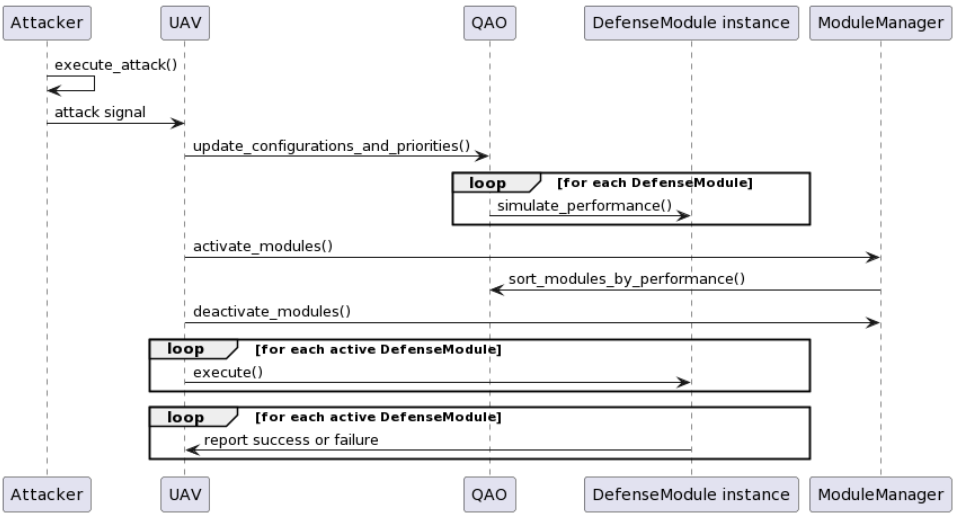


Figure 2. UAV defense system sequence diagram

4. EXPERIMENT

4.1. Experiment Setup

The experiment setup aims to compare the performance of two distinct systems: the monolithic, which refers to a non-adaptive modular security system, and the proposed pattern based system which is an adaptive modular security system. In our study, 'monolithic' denotes a traditional software architecture where functions are interdependent and not separated into modules, while 'adaptive modular' describes a system with distinct, interchangeable modules that can be modified in response to changing security threats. The performance comparison takes into account three main aspects: the success flag, computation time, and computation resource consumption. At the heart of this comparison is the ComparisonModule class, which is designed to conduct the performance assessment. When initialized, it takes two simulation builder objects as input parameters: one for the monolithic system (monolithic_simulation_builder) and another for the adaptive modular system (adaptive_modular_simulation_builder). The method named compare() within the class allows the user to specify two optional parameters. The first one, num_runs, represents the number of simulation runs to be executed for each system. The second parameter, num_uavs, indicates the number of UAVs involved in each simulation.

The experiment proceeds by executing both the monolithic and adaptive modular simulations, considering the number of runs and UAVs defined previously. This process is composed of several steps: First, six lists are initialized to store the performance metrics for each system: monolithic_results, adaptive_modular_results,

monolithic_times, adaptive_modular_times, monolithic_resources, and adaptive_modular_resources. Next, the experiment iterates through each simulation run, following a sequence of actions. The monolithic_simulations and adaptive_modular_simulations lists are created by invoking the respective simulation builders a number of times equal to num_uavs. The initial resource usage is recorded using the resource.getusage() function, and then the monolithic simulations are executed, with the success flag for each simulation being stored in the monolithic_results list. The time spent executing the monolithic simulations is calculated and stored in the monolithic_times list. After that, the resource usage is measured again, and the difference between the initial and final resource usage is stored in the monolithic_resources list. This same sequence of actions is repeated for the adaptive modular pattern simulations, and the results are saved in the corresponding adaptive_modular_results, adaptive_modular_times, and adaptive_modular_resources lists. After the simulation runs have been completed, the average success flag, computation time, and computation resource for the monolithic and adaptive modular systems are calculated by dividing the sum of each respective list by the list's length. Finally, the calculated averages are printed to provide a comparison of the two systems' performance.

4.2. Experiment Result

In the comparison of adaptive modular security systems as presented in the Table 1, we are looking at two methods, monolithic and adaptive modular system, to analyze their performance in terms of success rates, average computation time, and average computation resource usage. The comparison is conducted across four different scenarios with varying numbers of UAVs: 5, 10, 15, and 30. As the number of UAVs increases, the complexity and the number of potential attacks in the system also increase, which can impact the performance of both methods. The success rates are an important metric to consider, as they represent the percentage of successful defenses against attacks for the monolithic and adaptive modular systems. In all the scenarios, the adaptive modular pattern demonstrates higher success rates compared to the monolithic pattern, indicating that the adaptive modular pattern is more effective in defending against attacks. The effectiveness of a security system is crucial, as it determines the system's ability to protect the UAVs from various types of attacks.

Another aspect to consider is the average computation time, which refers to the time taken to run the simulations for each method. In all the scenarios, the adaptive modular pattern exhibits slightly shorter computation times than the monolithic pattern. This suggests that the adaptive modular pattern is more efficient in terms of computation time, which can be an important factor when implementing real-time security systems where quick response times are essential. The average computation resource is the memory usage during the simulations, representing the resource requirements of the security system. In all the scenarios, the adaptive modular pattern has slightly higher computation resource usage than the monolithic pattern, indicating that the adaptive modular pattern might require more memory resources to operate effectively.

The resource usage of a security system is an important consideration, as it can impact the overall performance and feasibility of implementing the system in real-world scenarios, especially when memory resources are limited. Taking all these factors into account, the adaptive modular pattern provides better performance in terms of both effectiveness and efficiency, making it a more attractive choice for adaptive modular security systems. The higher success rates and shorter computation times demonstrate the superiority of the adaptive modular pattern in defending against attacks and responding quickly. However, the increased resource usage of the monolithic pattern may need to be considered for systems with memory constraints. In such cases, the balance between the effectiveness, efficiency, and resource usage should be carefully evaluated to determine the most suitable method for the specific security requirements and constraints of the system.

Table 1. Monolithic vs Adaptive Modular based simulation (5, 10, 15, 20)

Num. UAVs	Attributes	Monolithic	Adaptive Modular
5	Success Rate(%)	73.91	100
5	Avg. Computation Time(s)	304.59	300.32
5	Avg. Computation Resource (MB)	108.55	108.64
10	Success Rate(%)	71.42	96.3
10	Avg. Computation Time(s)	611.37	600.65
10	Avg. Computation Resource (MB)	109.27	109.55
15	Success Rate(%)	70.76	95.06
15	Avg. Computation Time(s)	916.03	901.00
15	Avg. Computation Resource (MB)	109.76	109.96
30	Success Rate(%)	70.37	96.72

30	Avg. Computation Time(s)	1821.70	1801.99
30	Avg. Computation Resource (MB)	110.22	110.30

5. CONCLUSIONS

Our research involved a comparative analysis of two cybersecurity methods for UAVs: monolithic and adaptive modular systems. The evaluation focused on three critical aspects: success rates, average computation time, and average computation resource usage. The study was conducted across varying operational scales with 5, 10, 15, and 30 UAVs, representing different levels of system complexity and potential cyber-attack scenarios. Below are our key results, Success Rates: In all tested scenarios, the adaptive modular system consistently outperformed the monolithic system in terms of success rates. The 100% success rate of the adaptive modular system with 5 UAVs, as opposed to the 73.91% of the monolithic system, starkly highlights its superior effectiveness in defending against cyber-attacks. Even with the increase in the number of UAVs, the adaptive modular system maintained a notably higher success rate, underscoring its robustness and reliability in complex environments. Average Computation Time: The adaptive modular system exhibited slightly shorter computation times across all scenarios, suggesting its higher efficiency. In time-sensitive applications and real-time security systems, where quick response is crucial, this efficiency is of paramount importance. For instance, with 30 UAVs, the adaptive modular system completed simulations in 1801.99 seconds, marginally faster than the monolithic system's 1821.70 seconds. Average Computation Resource Usage: Although the adaptive modular system required marginally more computation resources than the monolithic system, this increase was minimal. For example, with 30 UAVs, the memory usage was 110.30 MB for the adaptive modular system, compared to 110.22 MB for the monolithic system. This indicates that while the adaptive system is slightly more resource-intensive, it does not significantly burden the computational resources. The results of our study clearly demonstrate the superiority of the adaptive modular pattern in providing effective and efficient cyber-defense for UAVs. The enhanced success rates and shorter computation times of the adaptive modular system make it a compelling choice for UAV cybersecurity, particularly in scenarios demanding rapid response and high reliability. While the adaptive system exhibits a slightly higher resource usage, the marginal increase is a worthwhile trade-off for the significant gains in performance and security effectiveness. Given these findings, the adaptive modular system stands out as a more attractive and viable option for UAV cyber-defense, especially in complex and dynamic operational environments. However, for systems where memory resources are a critical constraint, the balance between effectiveness, efficiency, and resource usage should be carefully considered. In conclusion, our research underscores the potential of adaptive modular systems in enhancing the cybersecurity posture of UAVs, paving the way for more secure and resilient UAV operations in various applications.

REFERENCES

[1] K. Chávez and O. Swed, "The proliferation of drones to violent nonstate actors," *Defence Studies*, vol. 21, no. 1, pp. 1-24, 2021, https://doi.org/10.1080/14702436.2020.1848426.

[2] G. Markarian and A. Staniforth. *Countermeasures for aerial drones*. Artech House. 2020. https://books.google.co.id/books?hl=id&lr=&id=In0qEAAAQBAJ.

[3] A. Al-Dhaqm, R. A. Ikuesan, V. R. Kebande, S. Razak, and F. M. Ghabban, "Research challenges and opportunities in drone forensics models," *Electronics*, vol. 10, no. 13, no. 1519, 2021, https://doi.org/10.3390/electronics10131519.

[4] V. Kharchenko and V. Torianyk, "Cybersecurity of the Internet of Drones: Vulnerabilities analysis and IMECA based assessment," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pp. 364-369, 2018, https://doi.org/10.1109/DESSERT.2018.8409160.

[5] K. M. Giannoutakis *et al*., "A Blockchain Solution for Enhancing Cybersecurity Defence of IoT," *2020 IEEE International Conference on Blockchain (Blockchain)*, Rhodes, Greece, 2020, pp. 490-495, 2020, https://doi.org/10.1109/Blockchain50366.2020.00071.

[6] J.-H. Cho et al., "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709-745, 2020, https://doi.org/10.1109/COMST.2019.2963791.

[7] A. Kenyon, L. Deka, and D. Elizondo, "Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets," *Computers & Security*, vol. 99, no. 102022, 2020, https://doi.org/10.1016/j.cose.2020.102022.

[8] T. Sobb, B. Turnbull, and N. Moustafa, "Supply chain 4.0: A survey of cyber security challenges, solutions and future directions," *Electronics*, vol. 9, no. 11, no. 1864, 2020, https://doi.org/10.3390/electronics9111864.

[9] A. Sabella, R. Irons-Mclean, and M. Yannuzzi. *Orchestrating and automating security for the internet of things: Delivering advanced security capabilities from edge to cloud for IoT*. Cisco Press. 2018. https://books.google.co.id/books?hl=id&lr=&id=SjFbDwAAQBAJ.

[10] T. Rains. *Cybersecurity Threats, Malware Trends, and Strategies: Learn to mitigate exploits, malware, phishing, and other social engineering attacks*. Packt Publishing Ltd. 2020. https://books.google.co.id/books?hl=id&lr=&id=8YLoDwAAQBAJ.

[11] N. Kaloudi and J. Li, "The AI-based cyber threat landscape: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1-34, 2020, https://doi.org/10.1145/3372823.
[12] E. C. K. Cheng and T. Wang, "Institutional strategies for cybersecurity in higher education institutions," Information, vol. 13, no. 4, no. 192, 2022, https://doi.org/10.3390/info13040192.
[13] W. Hoffman, "Is Cyber Strategy Possible?," *The Washington Quarterly*, vol. 42, no. 1, pp. 131-152, 2019, https://doi.org/10.1080/0163660X.2019.1593665.
[14] R. Walters and M. Novak, "Artificial Intelligence and Law," In *Cyber Security, Artificial Intelligence, Data Protection & the Law*, pp. 39-69, 2021, https://doi.org/10.1007/978-981-16-1665-5_3.
[15] E. Schrom et al., "Challenges in cybersecurity: Lessons from biological defense systems," *Mathematical Biosciences*, no. 109024, 2023, https://doi.org/10.1016/j.mbs.2023.109024.
[16] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42-51, 2018, https://doi.org/10.1109/MNET.2018.1700286.
[17] H. Hildmann and E. Kovacs, "Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security and public safety," *Drones*, vol. 3, no. 3, no. 59, 2019, https://doi.org/10.3390/drones3030059.
[18] M. Yahuza *et al*., "Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges," in *IEEE Access*, vol. 9, pp. 57243-57270, 2021, https://doi.org/10.1109/ACCESS.2021.3072030.
[19] R. Bonatti et al., "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606-641, 2020, https://doi.org/10.1002/rob.21931.
[20] C. Stevens, "Assembling cybersecurity: The politics and materiality of technical malware reports and the case of Stuxnet," *Contemporary Security Policy*, vol. 41, no. 1, pp. 129-152, 2020, https://doi.org/10.1080/13523260.2019.1675258.
[21] S. Jamshidi, A. Nikanjam, M. A. Hamdaqa, and F. Khomh, "Attack Detection by Using Deep Learning for Cyber-Physical System," in *Artificial Intelligence for Cyber-Physical Systems Hardening, Cham: Springer International Publishing*, pp. 155-179, 2022, https://doi.org/10.1007/978-3-031-16237-4_7.
[22] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467-3501, 2019, https://doi.org/10.1109/COMST.2019.2938259.
[23] L. Da Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941-2962, 2018, https://doi.org/10.1080/00207543.2018.1444806.
[24] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities," *Computer Networks*, vol. 183, Art. no. 107556, 2020, https://doi.org/10.1016/j.comnet.2020.107556.
[25] Y.-x. Xie, L.-x. Ji, L.-s. Li, Z. Guo, and T. Baker, "An adaptive defense mechanism to prevent advanced persistent threats," *Connection Science*, vol. 33, no. 2, pp. 359-379, 2021, https://doi.org/10.1080/09540091.2020.1832960.
[26] A. Srivastava, V. Parmar, S. Patel, and A. Chaturvedi, "Adaptive Cyber Defense: Leveraging Neuromorphic Computing for Advanced Threat Detection and Response," in *Proc. 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pp. 1557-1562, 2023, https://doi.org/10.1109/ICSCSS57650.2023.10169393.
[27] J. L. Cremer and G. Strbac, "A machine-learning based probabilistic perspective on dynamic security assessment," *International Journal of Electrical Power & Energy Systems*, vol. 128, no. 106571, 2021, https://doi.org/10.1016/j.ijepes.2020.106571.
[28] K. Al-Dosari and N. Fetais, "A new shift in implementing unmanned aerial vehicles (UAVs) in the safety and security of smart cities: a systematic literature review," *Safety*, vol. 9, no. 3, no. 64, 2023, https://doi.org/10.3390/safety9030064.
[29] J.-H. Cho et al., "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709-745, 2020, https://doi.org/10.1109/COMST.2019.2963791.
[30] S. K. Shandilya, S. Upadhyay, A. Kumar, and A. K. Nagar, "AI-assisted Computer Network Operations testbed for Nature-Inspired Cyber Security based adaptive defense simulation and analysis," *Future Generation Computer Systems*, vol. 127, pp. 297-308, 2022, https://doi.org/10.1016/j.future.2021.09.018.
[31] M. R. Rahman, R. Mahdavi Hezaveh, and L. Williams, "What are the attackers doing now? Automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1-36, 2023, https://doi.org/10.1145/3571726.
[32] N. Chowdhury, S. Katsikas, and V. Gkioulos, "Modeling effective cybersecurity training frameworks: A delphi method-based study," *Computers & Security*, vol. 113, no. 102551, 2022, https://doi.org/10.1016/j.cose.2021.102551.
[33] O. Yurekten and M. Demirci, "SDN-based cyber defense: A survey," *Future Generation Computer Systems*, vol. 115, pp. 126-149, 2021, https://doi.org/10.1016/j.future.2020.09.006.
[34] K.-Y. Tsao, T. Girdler, and V. G. Vassilakis, "A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks," *Ad Hoc Networks*, vol. 133, Art. no. 102894, 2022, https://doi.org/10.1016/j.adhoc.2022.102894.
[35] H. Sedjelmaci, A. Boudguiga, I. Ben Jemaa, and S. M. Senouci, "An efficient cyber defense framework for UAV-Edge computing network," *Ad Hoc Networks*, vol. 94, no. 101970, 2019, https://doi.org/10.1016/j.adhoc.2019.101970.

[36] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies," *Sensors*, vol. 20, no. 12, no. 3537, 2020, https://doi.org/10.3390/s20123537.
[37] Y. Tan, J. Wang, J. Liu and Y. Zhang, "Unmanned Systems Security: Models, Challenges, and Future Directions," in *IEEE Network*, vol. 34, no. 4, pp. 291-297, 2020, https://doi.org/10.1109/MNET.001.1900546.
[38] Z. Amiri, A. Heidari, N. J. Navimipour, and M. Unal, "Resilient and dependability management in distributed environments: A systematic and comprehensive literature review," *Cluster Computing*, vol. 26, no. 2, pp. 1565-1600, 2023, https://doi.org/10.1007/s10586-022-03738-5.

AUTHOR BIOGRAPHY

Gregorius Airlangga Received the B.S. degree in information system from the Yos Sudarso Higher School of Computer Science, Purwokerto, Indonesia, in 2014, and the M.Eng. degree in informatics from Atma Jaya Yogyakarta University, Yogyakarta, Indonesia, in 2016. He got Ph.D. degree with the Department of Electrical Engineering, National Chung Cheng University, Taiwan. He is also an Assistant Professor with the Department of Information System, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia. His research interests include artificial intelligence and software engineering include path planning, machine learning, natural language processing, deep learning, software requirements, software design pattern and software architecture.