STUART: ReSilient archiTecture to dynamically manage Unmanned aeriAl vehicle networks undeR atTack

Isadora G. Ferrão, Daniel Fernando Pigatto,
João Vitor Carvalho Fontes, Natássya B. F. Silva, David Espes,
Catherine Dezan and Kalinka Regina Branco

July 7, 2020

STUART: ReSilient archiTecture to dynamically manage Unmanned aeriAl vehicle networks undeR atTack

1st Isadora G. Ferrão
Univ. de São Paulo (USP)
São Carlos, SP - Brazil
isadoraferrao@usp.br

2rd Daniel F. Pigatto
Univ. do Tecnológica Federal do Paraná (UTFPR)
Curitiba, PR - Brazil
pigatto@utfpr.edu.br

3rd João V. C. Fontes
Univ. Federal de São Carlos (UFSCar)
São Carlos, SP - Brazil
joaofontes@ufscar.br

4th Natassya B. F. Silva
Univ. Tecnológica Federal do Paraná (UTFPR)
Cornélio Procópio, PR - Brazil
natassyasilva@utfpr.edu.br

5th David Espes
Université de Bretagne Occidentale (UBO)
Brest, France
david.espes@univ-brest.fr

6th Catherine Dezan
Université de Bretagne Occidentale (UBO)
Brest, France
Catherine.Dezan@univ-brest.fr

1st Kalinka R. L. J. C. Branco
Univ. de São Paulo (USP)
São Carlos, SP - Brazil
kalinka@icmc.usp.br

***Abstract*—The growing demand for Unmanned Aerial Vehicles (UAV) has the potential to increase productivity and economy in the industry, due to its use in various fields, such as health, security, aerial photography, surveillance, military missions, agriculture, etc. The production and use of the UAV have increased lately, and there is a demand for the improvement of decision-making, security, safety, and knowledge about relevant technologies. Thus, these vehicles must continually adapt to complex missions where they face unpredictable issues. In this context, the aim of this paper is to advance the state of the art through the definition and development of a resilient architecture for UAV that dynamically manages the network, even when subjected to an attack during a mission, integrating security methods and safety. The architecture will be composed by three modules: (1) decision-making module, (2) diagnosis module, and (3) resilient module. This work also investigates the incorporation of safety and security as a unified concept in the development of UAV.**

***Index Terms*—Unmanned Aerial Vehicle, Security, Safety, Resilient Architecture, Decision Making.**

I. Introduction

As a consequence of advances in the development and miniaturization of communication technology, Unmanned Aerial Vehicles are being used on a large scale in various fields, such as health [1], security [2], military missions [3] etc. In the industry, the growing demand for UAV has the potential to increase productivity and economy, as it can be seen in recent research and reports [4], [5]. However, increasing the production and use of these vehicles requires the improvement of solid decision–making principles, security, safety, and relevant technologies by the computing community, as they must continually adapt to face missions subject to unpredictable problems.

These improvements must consider the objective of the mission and the adaptation of the actions. In other words, the actions that the UAV can perform must be adaptable to the random events that arise during the mission, thus aiming to improve autonomy and safety for the aircraft. A failure can compromise the entire mission and the damage can be serious, including the UAV fall and even injuries to humans.

UAV face numerous challenges to navigate autonomously in a viable and safe way [6]. According to [7], only Remotely Piloted Autonomous Systems (RPAS) can be integrated with manned aircraft in unsegregated airspace and aerodromes. To fully integrate UAV into today's airspace, it is necessary to work on autonomous monitoring and management technologies that are resistant to attacks, so that they provide the necessary security for the aircraft and the environment. In this context, autonomy means dealing with limited resources for processing, storage, and high performance computing. However, this attribute is indispensable to guarantee the mission's success.

Therefore, this work advances the state of the art by defining an architecture that dynamically manages the network and can be resilient under attacks during a mission execution. It also investigates the incorporation of security and safety as a unified concept for the UAV development. The architecture proposed in this work will be located within the sphere, a safety and security platform for UAVs.

A. Motivation

Technological progress in electronic and avionic systems, especially about miniaturization and cost reduction, has boosted academic and economic interest in marketing and

*This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also acknowledge the support granted by UTFPR.

studying these aircraft. One of the main motivations of this work is focused on the future growth of the UAV market. Reports and studies foresee an important development in the commercial UAV market in the coming years, with an exponential growth [4], [5]. According to [8], by 2022 UAV will represent the largest share in the global market. Therefore, this rapid and growing evolution must be accompanied by cutting edge solutions and with particular attention to the UAV network to guarantee its safe integration into the airspace. Unmanned aerial vehicles can compromise the safety of people and environment, so social and environmental concerns are also motivating factors in this work.

II. Related Work

The focus of this work relies on resilient architectures taking into account that the *resilient* attribute is relevant to maintain the operation level of a stabilized UAV, even in the case of successful exploration [9]–[11]. Resilience in the field of engineering derives from the resilience of materials science and is characterized by the material's ability to return to its original state after deformation. In the figurative sense, it means the "ability to adapt or recover".

The authors in [10] address the improvement of system resilience to emerging attacks in the controlled environment, such as sensor attacks. Consequently, the authors developed a procedure called checkpoint and recovery, using historical data to recover the failed system states. First, they proposed a new concept of physical state recovery, presented as advancing the system to the current time, starting from a consistent historical physical state to match the values of the internal elements to the states. After that, a checkpoint protocol was designed to record system states for recovery. The protocol employs a sliding window that accommodates the detection delay to improve the correction of stored states. If no failure occurs, a checkpoint is considered correct and, therefore, a trusted state. The inspection is then stored and used for possible future retrievals. The protocol stipulates that detection is performed before saving system states. This characteristic improved the correction of checkpoints and therefore brought good results for the recovery likelihood.

Likewise, [9] presented the ContainerDrone framework that proposes resilient control of Denial of Service (DoS) attacks for real-time UAV systems using containers. Container technology is an open source and offers software isolation, and abstraction of many features in the Linux kernel. System isolation using containers leads to less execution overhead, less memory usage, and less footprint. ContainerDrone provides support mechanisms for three system resources, namely the Central Processing Unit (CPU), communication channel, memory, and also offers resilient control to DoS attacks for real-time UAV. To validate a prototype, a quadcopter with commercially available hardware was used and open-source software was implemented. The ContainerDrone framework has proven to be reliable in protecting against DoS attacks launched within the container by limiting attacker access to three critical system resources: CPU, memory, and communication channel. Experiments have shown that the proposed structure can be effective against various types DoS attacks launched effectively. In this paper, the authors had not considered physical component failures, software failures caused by bugs and logic failures, or any other attack than DoS.

As noted in the paper mentioned above, an effective way to improve attack resilience is to develop methods that can estimate system states with sufficient precision for control, regardless of compromised components. An advantage is that it informs the state of the system, even when some elements have been compromised, allowing the use of the same controllers as in case without attacks.

1) Comparative analysis: As mentioned above, previous works propose solutions to improve the security of UAV. However these works do not cover all the aspects of security. A comparison of these papers with our solution (STUART) is given in TABLE I.

TABLE I
Comparative analysis of architectures

	Architecture				
Feature	[9]	[10]	[11]	[12]	**[STUART]**
Resilience	Total	Total	Total	Incomplete	Total
Safety and Security (aggregates)	Incomplete	Incomplete	Incomplete	Total	Total
Automatic Detection	Partial	Incomplete	Partial	Incomplete	Partial
Network Monitoring	Total	Total	Total	Incomplete	Total
Decision Making	Partial	Incomplete	Partial	Incomplete	Total

As shown in TABLE I, although there are several architectures in the open literature, they do not satisfy all the requirements for making a generic and safe UAV. In the comparative analysis, only the authors in [12], and the one here proposed, STUART, are concerned with aspects of safety and security in an integrated way. However, the research in [9] is the only architecture that does not include the resilience characteristic.

Only STUART, [9], and [11] propose to carry out attacks automatically in a UAV network. However, all these papers detect attacks partially, because it is difficult at this time to embed the hardware necessary to identify all types of attack.

III. SPHERE

Security and safety Platform for HEteRogeneous systEms (SPHERE) is a safety and security platform. Unmanned vehicles have different peripherals and modules, so they require different levels of safety, thus leading to the need to classify modules according to their importance and criticality. According to [13], categorization is done in modules regarding their criticality: primary and secondary. Primary modules correspond to the crucial components of the aircraft to fly, to be aware of its location and to be able to make an emergency landing safely. Examples of these modules are GPS receiver, Autopilot, Radar, etc.

SPHERE is implemented by three modules: (1) CSU: it is responsible for the authentication and confidentiality mechanisms of the transmitted data, (2) SMU: is responsible for managing the registration, discovery, and forecasting of services, considering the security policies of sensitivity, and data trust between devices, and (3) SPMU : is responsible for monitoring the integrity of the modules.

IV. STUART: ReSilient archiTecture to dynamically manage Unmanned aeriAl vehicle networks undeR atTack

The developed architecture will be located within the security and safety platform in SPHERE framework. It will consist on three modules and two metrics, one being security and the other safety. The modules will be: (1) decision making module, (2) diagnostics module, and (3) resilience module.

- **Decision making module:** The increase in the production and use of UAV requires a significant improvement in built-in decision-making capabilities, as UAV must continually adapt to missions to solve unexpected internal problems or external dangers. Therefore, the decision-making module of this architecture will be responsible for deciding which actions are most suitable for the UAV to complete a given mission. It will make the decision autonomously, through random events that arise during a mission. This module will be responsible for ensuring the autonomy of the UAV through autonomous decision making. For the implementation of the decision-making module, different techniques are being analyzed to verify which is the most appropriate one. For the decision module, the possibility of using the Markov decision process is being analyzed. However, the final technique that will be used is still in the definition phase.
- **Diagnostic module:** The diagnostic module will be responsible for monitoring the network through the diagnosis of possible anomalies. To ensure that a UAV is secure on the network and does not compromise the mission, it is necessary to ensure that different attacks and failures are detected autonomously by the architecture. Therefore, this module will communicate directly with the decision–making module also responsible for notifying about the diagnosed anomaly. For example, in the case of a fake GPS signal during a mission, the diagnostic module will be responsible for detecting and informing the decision–making module about the failure. Such failures fall into the category of integrity. Integrity is an important feature to ensure that internal and external communication of the different modules that make up a UAV architecture is not compromised. For the implementation of the diagnostic module, different techniques are being analyzed to verify which is the most appropriate.The possibility of using Bayesian networks, Petri networks, or Complex networks is being investigated.
- **Resilience module:** Bearing in mind that safety and security flaws can be exploited by malicious entities, the architecture proposed will include a module for resilience. The resilience module will be responsible for the recovery and restoration of the UAV, in case it is subjected a under attack. Resilience is a relevant safety attribute to maintain the level of system operation of a stabilized UAV, even in the face of successful exploration. For the implementation of the resilience module, different techniques are being analyzed to verify which is the most appropriate. The resilience module will estimate the states of the system for the control and restoration of the UAV. Therefore, it will recover the states through historical data, or a state machine, techniques that are still in the definition phase. The advantages of these ways of recoveries are that they allow knowing the states of the system, even when some components are compromised. The resilience module will be located within the security and safety platform of SPHERE, as shown in Figure 1.

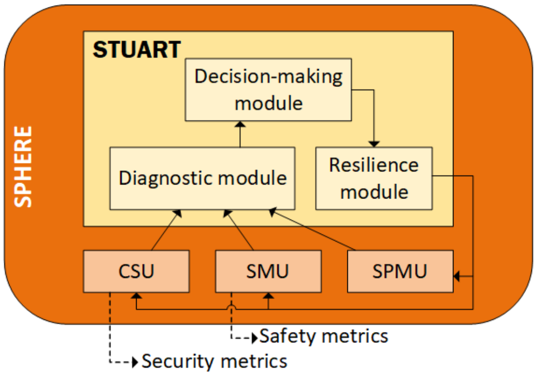

Fig. 1. Resilience Module

A. *Safety and security metrics*

One of the concerns in the design and development of UAV systems has been to ensure safety requirements. However, since these vehicles communicate with external entities, some architectures that were designed to provide safety requirements can present security flaws. Similarly, security requirements can have safety flaws. Therefore, it is necessary to investigate and treat safety and security in UAV together. Based on this, this architecture investigates the incorporation of security and safety metrics as a unified concept in the development of UAVs.

1) Node Criticality Index (NCI): The metric used to ensure safety and security in this work will be the Node Criticality Index (NCI) [12]. The NCI is the specification of a formal criticality classification responsible for determining the priority of the nodes in the network through an index. The reason for choosing this metric is twofold: 1) NCI guarantees the quality of service, safety, and security for the nodes that make up the network and 2) NCI provides valuable information to the system that can influence the decision-making of tasks.

Therefore, each module must be assigned with independent safety and security scores.

The determination of each score takes into account the need for different approaches. Although the scope of this work does not include the automatic attribution of this score, it is an open research topic for future integration. The fact that it requires human intervention is not necessarily a problem, as it is performed only once before the operation of the unmanned vehicle begins. Scoring is carried out first during a configuration phase and then automatically updated as a result of changes and events during system operation. The score assignment must be a number within a range from 0 to 1, meaning common and critical data, respectively.

The $NCIm_i^{Saf}$ will be the safety metric for a module i of this architecture and will be located in the secondary module of SPHERE, at SMU. The $NCIm_i^{Saf}$ will be used as a safety metric to measure the health of the module. The $NCIm_i^{Saf}$ of a module can be found by calculating the average between the $health$ index and the $modulePriority$ index, as shown in Equation 1. $Health$ represents a score between [0,1] that indicates the health status of a module. The $modulePriority$ is a score [0,1] that identifies the importance of a module for the overall security of the system.

$$NCIm_i^{Saf} = average(health_i, modulePriority_i) \qquad (1)$$

As security metrics for this architecture, $NCIm_i^{Sec}$ will be used and will be located in the secondary module of SPHERE, at CSU. The $NCIm_i^{Sec}$ will be used as a security metric to measure the security of the module. The $NCIm_i^{Sec}$ of a module can be achieved through the maximum score obtained by measuring how critical are two types of data: stored data (data stored by a module) and temporary data (data handled by a module but not stored), as shown in Equation 2. The $storedData$ is a score between [0,1] that represents the sensitivity of the stored data. $TemporaryData$ is a score between [0,1] that indicates the sensitivity of the temporary data. Both $storedData$ and $temporaryData$ must have independent approaches, as eventual security-related problems will affect the system in different ways, for example, stored data becomes a potential security concern if an unmanned vehicle is eventually stolen or captured. Temporary data is a relevant security concern for an unmanned vehicle under attack, as it will likely contain control messages that should replace the autopilot, assuming it is the attacked module.

$$NCIm_i^{Sec} = max(storedData_i, temporyData_i) \qquad (2)$$

2) Safety and security as unified concepts: The general NCI of a SPHERE module ($NCIm_i$), in this case, calculated by SPHERE, can be found by averaging security and safety sub–indices, as represented in Equation 3. In this case, both safety and security metrics will be working together to generate a final value that will define SPHERE overall security status, as shown in Figure 2. So the general NCI will directly affect the decision making of tasks. For example, if the general security value is 0.2 of [min: 0, max: 1], it means that it is safe and it has low critical problems regarding security and safety. However, if the final general security value indicates 1 out of [min: 0, max: 1], it means that it is unsafe, the module presents critical problems regarding security and safety, and action must be taken. This value can be used to provide relevant data for the development of communication protocols, task delegation management units, and also contribute to greater security and safety in communication architectures.

$$NCIm_i = average(NCIm_i^{sec}, NCIm_i^{saf}) \qquad (3)$$

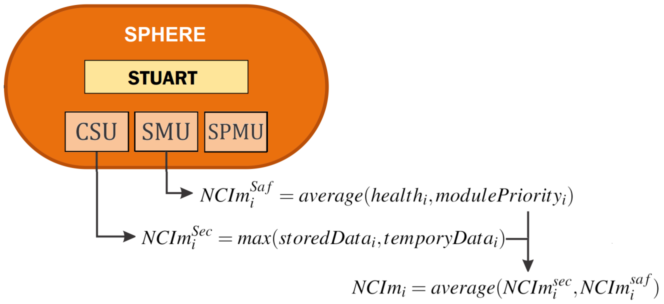
SPHERE
STUART
CSU
SMU
SPMU
$NCIm_i^{Saf} = average(health_i, modulePriority_i)$
$NCIm_i^{Sec} = max(storedData_i, temporyData_i)$
$NCIm_i = average(NCIm_i^{sec}, NCIm_i^{saf})$

Fig. 2. NCI Metrics

V. Results

The first step for the development of an architecture comprises the use of NCI as a safety and security metric. Therefore, a case study was carried out to assess how NCI could improve security in a traffic inspection mission. The NCI index is applied experimentally in different scenarios to allow discussions about likely implications.

A. Development Stages

The development of this study comprised the realization of tasks divided into six stages, which are: (1) Choice of mission, (2) Choice of test cases, (3) Choice of aircraft models, (4) NCI calculation methodology, (5) Generation of results and (6) Analysis of results.

1) Choice of mission: Unmanned aerial vehicles can provide aerial monitoring to traffic, road conditions, and respond to emergencies [14]. The aerial view offers a better perspective on the ability to cover a large area. It has the advantage of being mobile and being present in time and space. Another advantage of them is that the UAV can monitor a whole set of network of roads at a time and inform the base station of emergency or accidental sites.

Therefore, with increasing interest and investment in the use of UAV in traffic inspections, techniques to support the delegation of security and safety should also be carefully investigated. In this direction, NCI can provide relevant information for a variety of traffic enforcement applications. Therefore, a case study will be presented to show how NCI can improve safety in a traffic inspection mission by obtaining aerial images.

2) Choice of test cases: This study consists on a test where an aircraft is arranged in two case scenarios. In the first case, an aircraft acquires the data normally, without any problems. In the second scenario, the aircraft has a failure and requires a decision by a human operator.

3) Choice of aircraft model: The aircraft model determined was the 3DR Solo Quadcopter. The reason for choosing the aircraft is because it meets the needs for acquiring high-resolution aerial images.

4) NCI calculation methodology: The fact that NCI requires human intervention is not necessarily a problem, as it is performed only once before the operation of the unmanned vehicle begins. Scoring is carried out first during a configuration phase and then automatically updated as a result of changes and events during system operation. The score assignment in this study was a number within a range of 0 to 1, meaning common and critical data, respectively.

5) Generation and analisys of results : In this step, the values for the NCI were assigned to their respective formulas for security, safety and general aspects. This is the last step and comprises the analysis and discussion of the results obtained from the allocation of the NCI values for the different test cases.

B. Analysis of NCI in a traffic inspection and operation scenario

1) Scenario 1: Normal operation: In this scenario, only one aircraft, more specifically a 3DR Solo, is used to capture images of an accident involving two cars. It communicates with a base station via a Wi-Fi transmitter. The Solo has an Inertial Measurement Unit (IMU) and a GPS to identify its location, a camera to capture images, an autopilot that also activates the camera, motors, and Wi-Fi transmitter/receiver.

Security and safety aspects are analyzed by defining the NCI of each UAV module and then the NCI of the UAV. The comparison between the NCI of the modules of a UAV is an important resource that can help to move towards an immediate solution during the execution of the task. The NCI_m for each Solo module in regular operation is assumed, as shown in TABLE II.

TABLE II
SCENARIO 1: NORMAL OPERATION

Module	**NCIm^{se}**			**NCIm^{saf}**			**NCIm**
	stored-Data	temporary-Data	total	health	priority	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motors	0	0	0	0	0.5	0.25	0.125
Wifi	0	0.3	0.3	0	0.3	0.15	0.225

Since the GPS and IMU sensors, the engine, and the Wi-Fi transmitter do not store any data, the $storedData$ is set to 0. The camera stores images of the area affected by accident, so the $storedData$ index for this model is 0.5. The autopilot is responsible for saving the position data of the UAV, so it is set to 0.3. Although the GPS record is essential for the mission, it is not as crucial as the acquired images, which justifies the difference in scores between these modules. The GPS, IMU, autopilot, and Wi-fi transmitter/receiver manipulate data related to Solo's positioning, so $temporaryData$ is set to 0.3. The remaining modules do not deal with any data that could be considered risky for the UAV, so $temporaryData$ is set to 0.

Regarding safety, in regular operation, all modules are working correctly, so $health$ is set to 0. The most critical modules for proper functioning are IMU and autopilot, so they have been set to the highest $priority$, equal to 1. The GPS and Motors $priority$ scores are set to 0.5 because it is still possible to land the UAV, even if any of these modules fails. If the Solo is forced to fall, it is necessary to establish Wi-Fi communication to locate and retrieve the UAV, which justifies its value of 0.3 as $priority$ index. Finally, in relation to the camera's $priority$ index, it is set to 0 because, if it fails, the UAV can safely return to the base.

2) Scenario 2: Failed operation: In this case, Solo is used to capture images of an accident involving two cars. However, when carrying out the mission, it presents an engine failure and requires a decision by a human operator who is monitoring the entire operation.

This results in a change in the $health$ value of the damaged UAV engine to 1, which reflects in its end, which increases to 0.375. As a consequence, it affects NCI and creates an alert to prioritize communication. Therefore, a new mission is defined for the damaged Solo to maintain communication as long as possible. Consequently, the $priority$ of the Wi-Fi transmitter/receiver is changed to 1. This scenario is shown in TABLE III.

TABLE III
SCENARIO 2: FAILED OPERATION

Module	**NCIm^{se}**			**NCIm^{saf}**			**NCIm**
	stored-Data	temporary-Data	total	health	priority	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motors	**0**	**0**	**0**	**1**	**0.5**	**0.75**	**0.375**
Wifi	0	0.3	0.3	0	**1**	**0.5**	**0.4**

As seen above, the NCI is an index that can be applied not only for prioritizing communication but also for security and safety purposes due to its sub-indices. For example, prioritizing communications due to a failure in an entity can be handled quickly, as in scenario 2, where there was a failure in the engine, and an alert was issued to prioritize its communication. On the other hand, when it comes to ensuring the safety of an entity, SPHERE SMU can take appropriate measures based on the increase in the health index.

C. GPS spoofing attack with Fake GPS Location

GPS spoofing is considered to be one of the most recurring threats to UAVs [15]. The principle behind the GPS spoofing attack is that, by sending the drone's false geographic coordinates to the control system, it is possible to trick the onboard system that hijacks the vehicle in a different location for which it is commanded. In practice, there are GPS "spoofers" that are devices that create false GPS signals to trick receivers into thinking that they are in a different location or at different times. An example of a spoofer is [16], an application for android phones that falsifies the position by rewriting the location. In addition to the case study presented in the previous section, Fake GPS Location was used in this work to analyze how NCI behaves with this type of attack.

First, the Fake GPS application was installed on a smartphone with the Android operating system. It was then configured to falsify the location. The correct location would be from the Institute of Mathematical and Computer Sciences to the hospital in the city of São Carlos. However, after activating the counterfeiting with Fake GPS, the location shown indicates an opposite path.

From this, if NCI were applied to this scenario, the integrity of the module containing the GPS would reflect this variation, increasing the $health$ score to the value of 1 [min: 0, max: 1], as represented in the TABLE IV. Value 1 represents the most critical value. Therefore, an alert would be issued in the architecture, and appropriate action should take place. The Wi-Fi communication is prioritized and the $priority$ index is increased to 1. Although very specific, this case study can be seen in many applications of unmanned vehicles, and this validation can be extended to other scenarios.

TABLE IV
SCENARIO: FAILURE IN GPS

Module	**NCIm^{se}**			**NCIm^{saf}**			**NCIm**
	stored-Data	temporary-Data	total	health	priority	total	
GPS	**0**	**0.3**	**0.3**	**1**	**0.5**	**0.75**	**0.52**
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motors	0	0	0	0	0.5	0.25	0.125
Wifi	0	0.3	0.3	0	**1**	**0.5**	**0.4**

VI. CONCLUSION

This research presents a new resilient architecture named STUART to allow the recovery of a UAV network under attack. Aiming to provide the dynamical recovery, the architecture uses a resilient model, which integrates security and safety in a unique metric named NCI. Two case studies were presented: (1) applying the methodology to traffic inspection and operation, which allows to identify the behaviour of NCI while there is a motor failure; and (2) applying the methodology in a GPS spoofing attack situation, which highlights how robust NCI can be for the identification of attacks.

As it can be seen in the analysis, the NCI is an index that can be applied not only for prioritizing communication but also for security and safety purposes due to its sub-indices. As future work, we intend to continue the development of the other modules that will compose the STUART architecture.

REFERENCES

[1] Q. Chen, X. Wen, F. Wu, and Y. Yang, "Defect detection and health monitoring of steel structure based on uav integrated with image processing system," in *Journal of Physics: Conference Series*, vol. 1176, no. 5. IOP Publishing, 2019, p. 052074.

[2] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing uav communications via joint trajectory and power control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1376–1389, 2019.

[3] S. K. Chaturvedi, R. Sekhar, S. Banerjee, and H. Kamal, "Comparative review study of military and civilian unmanned aerial vehicles (uavs)," *INCAS Bulletin*, vol. 11, no. 3, pp. 183–198, 2019.

[4] DroneIndustry, "The european drone industry," https://www.droneii.com/wp-content/uploads/2018/06/The-European-Drone-Industry-v1.1.pdf, 2018, accessed: 2019-11-23.

[5] Goldman, "Goldman sachs research report," https://www.goldmansachs.com/insights/technology-driving-innovation/drones/, 2019, accessed: 2019-11-25.

[6] M. Alwateer, S. W. Loke, and A. Zuchowicz, "Drone services: issues in drones for location-based services from human-drone interaction to information processing," *Journal of Location Based Services*, vol. 13, no. 2, pp. 94–127, 2019.

[7] H. Usach, J. A. Vila, C. Torens, and F. Adolf, "Architectural design of a safe mission manager for unmanned aircraft systems," *Journal of Systems Architecture*, vol. 90, pp. 94–108, 2018.

[8] Markets and Markets, "Le cabinet marketsandmarkets annonce de belles perspectives de croissance annuelle pour ce marché mondial dans les 5 ans à venir, prévoyant notamment la multiplication rapide des applications commerciales," http://www.mesures.com/vision-industrielle/item/13721-une-croissance-proche-de-20-sur-le-marche-des-drones, 2017, accessed: 2019-11-25.

[9] J. Chen, Z. Feng, J.-Y. Wen, B. Liu, and L. Sha, "A container-based dos attack-resilient control framework for real-time uav systems," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1222–1227.

[10] F. Kong, M. Xu, J. Weimer, O. Sokolsky, and I. Lee, "Cyber-physical system checkpointing and recovery," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 22–31.

[11] M.-K. Yoon, B. Liu, N. Hovakimyan, and L. Sha, "Virtualdrone: virtual sensing, actuation, and communication for attack-resilient unmanned aerial systems," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*. ACM, 2017, pp. 143–154.

[12] D. F. Pigatto, L. Gonçalves, G. F. Roberto, J. F. Rodrigues Filho, N. B. F. Da Silva, A. R. Pinto, and K. R. L. J. C. Branco, "The hamster data communication architecture for unmanned aerial, ground and aquatic systems," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 705–723, 2016.

[13] D. F. Pigatto, J. Smith, K. R. Lucas, and J. C. Branco, "Sphere: A novel platform for increasing safety & security on unmanned systems," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 1059–1066.

[14] R. I. H. Abushahma, M. A. Ali, N. A. A. Rahman, and O. I. Al-Sanjary, "Comparative features of unmanned aerial vehicle (uav) for border protection of libya: a review," in *2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 2019, pp. 114–119.

[15] J. Gaspar, R. Ferreira, P. Sebastião, and N. Souto, "Capture of uavs through gps spoofing," in *2018 Global Wireless Summit (GWS)*. IEEE, 2018, pp. 21–26.

[16] FakeGPS, "Fake gps location," 2020. [Online]. Available: https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=pt_BR