

Article

Not peer-reviewed version

---

# Autonomous Strike UAVs for Counterterrorism Missions: Challenges and Preliminary Solutions

---

[Meshari Aljohani](#)\*, [Ravi Mukkamala](#), [Stephan Olariu](#)

Posted Date: 3 January 2024

doi: 10.20944/preprints202401.0072.v1

Keywords: Drone; UAV; Smart contract; black box; random forest



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

# Autonomous Strike UAVs for Counterterrorism Missions: Challenges and Preliminary Solutions

Meshari Aljohani \*<sup>1</sup>, Ravi Mukkamala<sup>1</sup> and Stephan Olariu<sup>1</sup>

Department of Computer Science, Old Dominion University, Norfolk, Virginia, 23529; maljo001@odu.edu, rmukkama@odu.edu, solariu@odu.edu

\* Correspondence: maljo001@odu.edu

**Abstract:** UAVs are becoming a crucial tool in modern warfare, primarily due to their cost-effectiveness, risk reduction, and ability to perform a wider range of activities. The use of autonomous UAVs to conduct strike missions against highly valuable targets is the focus of this research. Due to developments in ledger technology, smart contracts, and machine learning, such activities formerly carried out by professionals or remotely flown UAVs are now feasible. Our study provides the first in-depth analysis of challenges and potential solutions for successful implementation of an autonomous UAV mission.

**Keywords:** Drone; UAV; Smart contract; black box; random forest

## 1. Introduction and motivation

For several decades, the United States has employed remotely piloted Unmanned Aircraft Vehicles (UAV) across its military services [1]. As several military analysts have pointed out, UAVs are attractive from both strategic and tactical standpoints because they are cheaper to deploy than crewed (i.e., manned) aircraft and they can carry out dangerous missions without risking human lives [2].

In addition, with the gradual introduction of more and more sophisticated UAVs, supported by advances in machine learning (ML), several new types of missions are within reach. These include cargo and resupply, air-to-air combat, close air support, communication relays, aerial refueling, search-and-rescue, and counterterrorism missions [3]. It is becoming evident that, due to increased technological sophistication and reduced size, UAVs are well suited to carry out many types of missions that, until very recently, could only be performed successfully by crewed aircraft. Such considerations could enable the military to station UAVs closer to the front lines than crewed aircraft, potentially reducing the time to carry out time-sensitive missions.

As the US is withdrawing from conflicts around the world, the military will have to increasingly rely on UAVs for various missions including intelligence, surveillance, and acquisition of ground targets in counterterrorism missions [4,5].

It is widely known that the U.S. Department of Defense (DOD) is developing several experimental concepts such as aircraft system-of-systems, swarming, and lethal autonomous weapons that explore new ways of employing future generation UAVs [3,6–8]. Aligned with this effort, the main vision of this paper is to take UAVs to the next level of sophistication by enabling autonomous UAVs to carry out strike missions against entrenched high-value terrorists.

While, in the past, such missions were carried out by Special Operations personnel and/or by remotely-piloted UAVs, recent advances in blockchain technologies, smart contracts, and ML have made it possible for these missions to be carried out successfully by autonomous UAVs. Our first main contribution is to identify the main challenges that have to be overcome to implement our vision; our second main contribution is to propose preliminary solutions to those challenges. To the best of our knowledge, ours is the first paper in the literature available to us, that discusses the challenges inherent in making such missions feasible and the ways in which these challenges can be successfully overcome.

The remainder of the paper is structured as follows: Section 2 introduces our working scenario and basic assumptions. Section 3 identifies the main challenges involved in enabling autonomous strike UAVs. Next, Section 4 provides preliminary solutions to the challenges identified in Section 3. Section 5, given a collection of tasks  $T_1, T_2, \dots, T_n$  that have to be executed as part of the mission, evaluates the conditional probability of a successful completion of a future task, say  $T_{i+1}$ , given the completion status of the currently executed task,  $T_i$ . Further, Section 6 identifies some of the sensor types that are provided in the UAV in support of its autonomous mission. Section 7 offers the details of our machine learning framework in support of autonomous strike UAVs as well as a host of empirical evaluations. Finally, Section 8 offers concluding remarks and maps out directions for future work.

## 2. Working scenario – A high-level description

It has been recognized that UAVs have tremendous potential for air-to-ground strike missions. A strike UAV has the capability to launch weapons such as *precision-guided missiles* against a ground target. While the state of the art in air-to-ground strike missions is that there is always a man in the loop, in the sense that the UAV is piloted remotely, the vision of this work is to leverage the latest technology to enable fully autonomous strike UAVs.

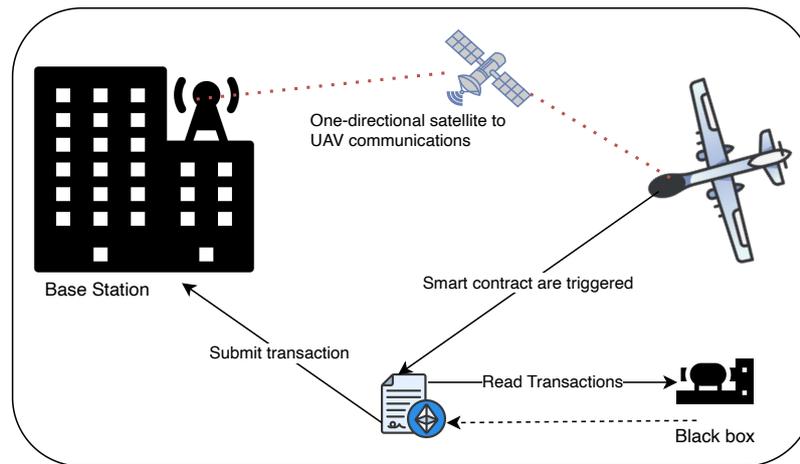
With this in mind, throughout this paper, we assume that a UAV, henceforth referred to as a *drone*, is deployed in support of a strike mission involving a high-value terrorist target in a foreign country. Such missions may well operate in “contested territory” in which terrorist forces have an active presence. By their nature, these missions are top secret and do not rely on intelligence collected from foreign state actors. In fact, the mission may be deployed without the specific approval of foreign state actors.

Given the context of the mission we are contemplating, we assume that the targeted terrorist organization does not have the wherewithal to take out or jam US communication satellites and, consequently, we rely on satellite-to-drone communications for the duration of the mission.

We further assume that the drone carries, as part of its payload, standard on-board sensory equipment, including a gyroscope (or inertial navigation system), electro-optical cameras, and infrared (IR) cameras for use at night, as well as synthetic aperture radar (SAR). SAR is a form of radar used to create two- or three-dimensional reconstructions of objects, such as landscapes. SAR uses the motion of the radar antenna over a target region to provide finer spatial resolution than conventional radars. Such missions must avoid civilian casualties. In fact, we assume that the mission will be aborted if civilians are close to the intended target. In this regard, night missions are safer to execute, because civilians (especially children) are very unlikely to be present, but they require far more sophistication in terms of localization and imagery processing.

Figure 1 provides a comprehensive overview of our working scenario. Here, we see a network of systems working together to achieve a targeted mission. The systems consist of a base station (also referred to as a Mission Control Center (MC2)), a satellite communication system, a UAV (complete with an on-board Blackbox (BBX), a Smart Contract (SC)), and a blockchain (on the ground). While this paper assumes a single UAV, the proposed system is equally applicable to multiple coordinated UAVs.

Preparing a mission of the type we have in mind, requires the UAV to execute a number of *training runs* both day-time and night-time, each intended to evaluate the sequence of tasks (see Section 5) that, collectively, make up the mission. The data collected in each such training run is carefully analyzed by human experts back at the MC2 to establish the conditional probability of a future task given the status of the current one. Data from successive runs are aggregated by *human experts* at the MC2 and used to train an ML model. Specifically, human experts analyze the flight information stored in the on-board *Black Box* (BBX, for short), e.g., collected drone imagery along with maneuvers performed by the UAV in response to sensory information. As a result, human experts are able to assign conditional probabilities to individual tasks based on the successful or partly failed status of the previous task in the sequence. As already mentioned, strict conditions for avoiding civilian casualties are stipulated and encoded as part of the SC that is to oversee the mission.



**Figure 1.** A comprehensive overview of the working scenario.

In our vision, once the drone is trained, it is ready to carry out the mission autonomously, without the need for continuous communication with the MC2. This absence of communication is crucial for security purposes, as it prevents unauthorized access and tampering with the data or drone operations.

### 3. Challenges

In order to make the vision of autonomous strike UAV a reality, several technical challenges have to be overcome:

- Tamper-free collection and storage of accurate in-flight sensory data. Indeed, reliable, untampered data collected during training missions are absolutely necessary to train the ML model. Such data is also essential for auditability purposes, especially if the mission is aborted. Finally, the provenance of each piece of sensory data will have to be recorded and ascertained;
- Determining exact drone location. This challenge contains, as a sub-challenge, the time synchronization of the drone and the MC2. While initially the drone and the MC2 are assumed to be synchronized, due to clock drift, synchronization may be lost and needs may need re-synchronization. In addition to time synchronization, the drone has to know its exact location (e.g., its 3-D geographic or polar coordinates) all the time;
- Enabling secure communication between the MC2 and the drone. Such communication may involve time-dependent frequency hopping and, as such, requires tight time synchronization between the two;
- Identifying the target and confirming that the target is clear of civilians. A fundamental requirement of a successful mission is to avoid civilian casualties;
- Allowing dynamic mission changes. This presupposes that some form of reliable communication has been established between the MC2 and the drone.
- Tamper-resistance: if downed or captured, the drone should blank/destroy its BBX.

### 4. Technical details: addressing the challenges

The main goal of this section is to outline ways in which some of the challenges identified in Section 3 can be addressed. Our solutions are, necessarily, sketchy but should give the reader a sense of what the technical solutions involve.

#### 4.1. Tamper-free collection and storage of sensory data

In our vision, the workhorse of the mission is the on-board BBX implementing, for the duration of the current flight, the functionality of an *append-only ledger*. As such, the BBX records and stores every piece of sensory data collected by the drone's sensors, along with a time stamp and provenance. In addition, the BBX serves as an on-board command and control center. Pre-flight, the BBX is loaded with

the individual tasks that make up the mission, with the aggregated conditional probabilities discussed in Section 5, confirmed by human experts at MC2. The integration of advanced BBX and append-only ledger (i.e., blockchain) technology creates a powerful and secure system for data collection and mission management. In our system, the BBX functions as an essential safeguard for the information collected by the drone, while at the same time, ensuring integrity, traceability, transparency, security, and auditability. Each data entry or transaction can be traced back to its origin, making it easier to verify its authenticity and identify any potential issues.

#### 4.2. Determining exact drone location

Localization is one of the fundamental challenges in autonomous navigation and has received extensive attention in the scholarly literature [9–14]. Since, as already mentioned, GPS communications are available to the mission, we contemplate using GPS for both synchronization (to within 100 nano-seconds) with the MC2 and drone localization [2,14].

#### 4.3. Enabling secure communication between the MC2 and the drone

One of the fundamental tasks that have to be performed as part of a successful mission is communication with the MC2. Depending on the specifics of the mission, two types of communication may be required. For missions deployed within about 50 miles of the MC2, line-of-sight communications may be used. For missions beyond 50 miles from the home base, non-line-of-sight (NLoS) communications are required. In this paper, we assume NLoS control communications between the MC2 and the drone, forwarded through one of the several satellite constellations established, for similar purposes, by the US military [3]. If, however, the communication link to the MC2 is lost, the drone is programmed to return to its base, and the mission is aborted.

#### 4.4. Identifying and confirming target

As already mentioned, the missions we contemplate involve a number of training runs whose stated goal, among others, is to locate and identify the target (which, to fix ideas, we assume the target to be an isolated building). The location of the building is acquired through day-time training missions and confirmed by a human expert at the MC2. The same is repeated during subsequent night-time training runs, where the target is confirmed using IR and SAR imagery. As already mentioned, the mission is aborted if civilians are identified close to the target. However, since we envision a night-time execution of the strike, the presence of civilians in close proximity to the target is a very unlikely event.

#### 4.5. Allowing dynamic mission changes

It is essential for the MC2 to be able to order aborting a mission in progress. This is accomplished by sending a specific encoded message to the drone using the communication channel. Such an order will have to be confirmed by the drone using a different communication channel, such as a different satellite in the constellation.

### 5. Evaluating the probability of mission success

We take the view that the *mission* involves performing a *sequence* of tasks  $T_1, T_2, \dots, T_n, \dots$ . Let the sequence of tasks performed by a given drone be observed and evaluated by a SC and their outcome recorded in an on-board, tamper-free, append-only ledger, implemented as a standard on-board BBX. While, in this work, we assume that the tasks are atomic and indivisible, in reality, each task could be an amalgam of other, simpler tasks. An example would be the localization task, one of the fundamental tasks to be performed by the drone. This task typically involves a Markovian sequence of other tasks, each processing sensor readings and inertial system input [9]. In this context, a task is considered either “successfully completed” (“successful”, for short), or “partially completed” (“incomplete”, for short), depending on whether or not certain task-specific performance parameters are met.

We assume that just prior to being deployed, the drone's on-board BBX was loaded with a SC cognizant of the tasks to be performed and of the conditions under which the mission cannot be completed successfully and should be aborted. We assume that such conditions are expressed in terms of the *unconditional* probability of a task being completed successfully. Since each task in the sequence of tasks contributes to the success of the mission if the probability of success of any task is below a mission-specific threshold, the SC will inform the MC2 and seek permission to abort the mission and return to base.

In the light of the above, in the remainder of this work, we concern ourselves with reasoning about the outcome of a future task, given the observed sequence of completed tasks. As already mentioned, the goal is to advise the MC2 on the likelihood of mission failure as a result of improper completion of individual tasks.

For a positive integer  $k$ , ( $k \geq 1$ ), let  $A_k$  denote the event that task  $T_k$  has been completed successfully. Our basic assumption is that the following relation holds true

$$\Pr[A_{k+1}|A_k \cap A_{k-1} \cap \dots \cap A_2 \cap A_1] = \Pr[A_{k+1}|A_k]. \quad (1)$$

Equation (1) states that the outcome of the  $(k+1)$ -th task, in terms of being successful or incomplete, only depends on the completion status of the previous task, namely task  $T_k$ , and not on the status of earlier tasks.

Recall that the SC associated with the mission stipulates the conditions that have to be met for a mission to be aborted. In our derivation,  $p_1 = \Pr[A_1]$ , the probability that the first task is successful, plays a distinguished role. A good example of the first task to be performed as part of the overall mission is drone localization. If the drone is deployed from an aircraft, say, a C-130 transport aircraft, then with a high probability, the initial localization task will be successful. Otherwise, it might not. With this in mind, in Subsection 5.1  $p_1$  will be assumed to be known, while in Subsection 5.2  $p_1$  will be assumed to be unknown. We begin by introducing notation:

- Let  $\Pr[A_{k+1}|A_k]$  be the conditional probability that the  $(k+1)$ -th task is successful given that the  $k$ -th task was successful;
- Let  $\Pr[A_{k+1}|\bar{A}_k]$  be the conditional probability that the  $(k+1)$ -th task is successful given that the  $k$ -th task was incomplete;
- Let  $\Pr[\bar{A}_{k+1}|A_k]$  be the conditional probability that the  $(k+1)$ -th task is incomplete given that the  $k$ -th task was successful;
- Let  $\Pr[\bar{A}_{k+1}|\bar{A}_k]$  be the conditional probability that the  $(k+1)$ -th task is incomplete given that the  $k$ -th task was incomplete.

We use  $\tau_k(s,s)$ ,  $\tau_k(u,s)$ ,  $\tau_k(s,u)$ ,  $\tau_k(u,u)$ , as shortcuts for  $\Pr[A_{k+1}|A_k]$ ,  $\Pr[A_{k+1}|\bar{A}_k]$ ,  $\Pr[\bar{A}_{k+1}|A_k]$ ,  $\Pr[\bar{A}_{k+1}|\bar{A}_k]$ .

While these conditional probabilities are, in general, functions of  $k$ , we assume that they are all *time-independent*, and will drop any reference to  $k$ , writing  $\tau(s,s)$ ,  $\tau(u,s)$ ,  $\tau(s,u)$ ,  $\tau(u,u)$ . We assume that these conditional probabilities are known for the specific mission in support of which the drone is being deployed. Indeed, the various probabilities can be learned during the training runs and are loaded into the BBX at deployment time. Certainly, the SC knows these conditional probabilities as well.

### 5.1. When the probability $p_1$ is known

In this subsection, assuming a known value for  $p_1$  as well as the conditional probabilities defined above, we are turning our attention to the task of finding the *unconditional* probability of the event that the  $n$ -th task is successful, for  $n \geq 2$ . Using the Law of Total Probability we write

$$\begin{aligned}
 \Pr[A_n] &= \Pr[A_n|A_{n-1}] \Pr[A_{n-1}] + \Pr[A_n|\bar{A}_{n-1}] \Pr[\bar{A}_{n-1}] \\
 &= \tau(s, s) \Pr[A_{n-1}] + \tau(u, s) \Pr[\bar{A}_{n-1}] \\
 &= \tau(s, s) \Pr[A_{n-1}] + [1 - \tau(u, u)] [1 - \Pr[A_{n-1}]] \\
 &= \Pr[A_{n-1}] [\tau(s, s) + \tau(u, u) - 1] + [1 - \tau(u, u)] \\
 &= \lambda \Pr[A_{n-1}] + \mu,
 \end{aligned} \tag{2}$$

where  $\lambda = \tau(s, s) + \tau(u, u) - 1$  and  $\mu = 1 - \tau(u, u)$ . For further reference, we note that

$$\tau(s, s) = \lambda + \mu. \tag{3}$$

In order to avoid trivialities, we assume that  $\lambda$  is different from  $-1$ ,  $0$ , and  $1$ . Indeed,  $\lambda = -1$  implies  $\tau(s, s) + \tau(u, u) = 0$  which, in turn, implies that  $\tau(s, s) = \tau(u, u) = 0$ . Similarly,  $\lambda = 1$  implies  $\tau(s, s) + \tau(u, u) = 2$  which, in turn, implies  $\tau(s, s) = \tau(u, u) = 1$ . Finally if  $\lambda = 0$  then  $\Pr[A_n] = \mu$  for all  $n \geq 2$ . In such a case, however, the sequence of task outcomes is rather trivial. Thus, from now on, we assume  $0 < |\lambda| < 1$ . Now, (2) and  $\Pr[A_1] = p_1$ , combined, yield the following recurrence:

$$\Pr[A_n] = \begin{cases} p_1 & \text{for } n = 1; \\ \lambda \Pr[A_{n-1}] + \mu & \text{for } n \geq 2. \end{cases} \tag{4}$$

A simple telescoping argument affords us the following closed form for  $\Pr[A_n]$ , ( $n \geq 1$ ):

$$\Pr[A_n] = \left[ p_1 - \frac{\mu}{1 - \lambda} \right] \lambda^{n-1} + \frac{\mu}{1 - \lambda}. \tag{5}$$

### 5.2. When $p_1$ is unknown

We now assume that  $p_1$  is merely a parameter and that we have no knowledge about whether or not  $A_1$  will be successful. In this case, we will be evaluating the following conditional probabilities:

- $\Pr[A_{k+1}|A_1]$  – the conditional probability that the  $(k+1)$ -th task is successful given that the first task is successful. We use  $\rho_k(s, s)$  as a shortcut for  $\Pr[A_{k+1}|A_1]$ ;
- $\Pr[A_{k+1}|\bar{A}_1]$  – the conditional probability that the  $(k+1)$ -th task is successful given that the first task is incomplete. We use  $\rho_k(u, s)$  as a shortcut for  $\Pr[A_{k+1}|\bar{A}_1]$ ;
- $\Pr[\bar{A}_{k+1}|A_1]$  – the conditional probability that the  $(k+1)$ -th task is incomplete given that the first task was successful. We use  $\rho_k(s, u)$  as a shortcut for  $\Pr[\bar{A}_{k+1}|A_1]$ .
- $\Pr[\bar{A}_{k+1}|\bar{A}_1]$  – the conditional probability that the  $(k+1)$ -th task is incomplete given that the first task was also incomplete. We use  $\rho_k(u, u)$  as a shortcut for  $\Pr[\bar{A}_{k+1}|\bar{A}_1]$ ;

We note that the semantics of the conditional probabilities just defined do not allow us to drop the subscript  $k$  from  $\rho_k(s, s)$ ,  $\rho_k(u, s)$ ,  $\rho_k(s, u)$  and  $\rho_k(u, u)$ . We begin by evaluating  $\rho_k(s, s)$ .

$$\begin{aligned}
 \rho_k(s, s) &= \Pr[A_{k+1}|A_1] = \Pr[A_{k+1} \cap (A_k \cup \bar{A}_k) | A_1] \\
 &= \Pr[A_{k+1} \cap A_k | A_1] + \Pr[A_{k+1} \cap \bar{A}_k | A_1] \\
 &= \Pr[A_{k+1}|A_k \cap A_1] \Pr[A_k|A_1] \\
 &+ \Pr[A_{k+1}|\bar{A}_k \cap A_1] \Pr[\bar{A}_k|A_1].
 \end{aligned} \tag{6}$$

Let us note that

- By (1),  $\Pr[A_{k+1}|A_k \cap A_1] = \Pr[A_{k+1}|A_k] = \tau(s, s)$ ;
- $\Pr[A_k|A_1] = \rho_{k-1}(s, s)$ ;
- Similarly,  $\Pr[A_{k+1}|\bar{A}_k \cap A_1] = \Pr[A_{k+1}|\bar{A}_k] = \tau(u, s) = 1 - \tau(u, u)$ ;
- $\Pr[\bar{A}_k|A_1] = 1 - \Pr[A_k|A_1] = 1 - \rho_{k-1}(s, s)$ .

On replacing the expressions above into (6), we write

$$\begin{aligned}\rho_k(s, s) &= \tau(s, s)\rho_{k-1}(s, s) + [1 - \tau(u, u)] [1 - \rho_{k-1}(s, s)] \\ &= \rho_{k-1}(s, s) [\tau(s, s) + \tau(u, u) - 1] + 1 - \tau(u, u) \\ &= \lambda\rho_{k-1}(s, s) + \mu,\end{aligned}\tag{7}$$

where  $\lambda$  and  $\mu$  were defined in Subsection 5.1. As before, we assume  $0 < |\lambda| < 1$ . Now, noticing that  $\rho_1(s, s) = \tau(s, s)$ , we obtain the following recurrence for  $\Pr[A_{k+1}|A_1]$ , ( $k \geq 1$ ):

$$\rho_k(s, s) = \begin{cases} \lambda + \mu & \text{for } k = 1; \\ \lambda\rho_{k-1}(s, s) + \mu & \text{for } k \geq 2. \end{cases}\tag{8}$$

Simple manipulations yield the following closed form for  $\rho_k(s, s)$  for  $k \geq 1$ :

$$\begin{aligned}\rho_k(s, s) &= \left[ \tau(s, s) - \frac{\mu}{1 - \lambda} \right] \lambda^{k-1} + \frac{\mu}{1 - \lambda} \\ &= \left[ \lambda + \mu - \frac{\mu}{1 - \lambda} \right] \lambda^{k-1} + \frac{\mu}{1 - \lambda} \\ &= \frac{1 - \lambda - \mu}{1 - \lambda} \lambda^k + \frac{\mu}{1 - \lambda} \\ &= \left( 1 - \frac{\mu}{1 - \lambda} \right) \lambda^k + \frac{\mu}{1 - \lambda},\end{aligned}\tag{9}$$

which is the desired closed form for  $\rho_k(s, s)$ . Next, we turn our attention to the task of evaluating  $\rho_k(u, s)$ . We notice that

$$\rho_1(u, s) = \Pr[A_2|\bar{A}_1] = \tau(u, s) = 1 - \tau(u, u) = \mu.\tag{10}$$

Further, for  $k \geq 2$ , we write

$$\begin{aligned}\rho_k(u, s) &= \Pr[A_{k+1}|\bar{A}_1] = \Pr[A_{k+1} \cap (A_k \cup \bar{A}_k) | \bar{A}_1] \\ &= \Pr[A_{k+1} \cap A_k | \bar{A}_1] + \Pr[A_{k+1} \cap \bar{A}_k | \bar{A}_1] \\ &= \tau(s, s)\rho_{k-1}(u, s) + [1 - \tau(u, u)] [1 - \rho_{k-1}(u, s)] \\ &= [\tau(s, s) + \tau(u, u) - 1] \rho_{k-1}(u, s) + 1 - \tau(u, u) \\ &= \lambda\rho_{k-1}(u, s) + \mu.\end{aligned}\tag{11}$$

Notice that (10) and (11) lead quite naturally to the following recurrence describing the behavior of  $\rho_k(u, s)$ .

$$\rho_k(u, s) = \begin{cases} \mu & \text{for } k = 1; \\ \lambda\rho_{k-1}(u, s) + \mu & \text{for } k \geq 2, \end{cases}\tag{12}$$

from where we obtain the following closed form for  $\rho_k(u, s)$  for  $k \geq 1$ :

$$\rho_k(u, s) = \frac{\mu}{1 - \lambda} [1 - \lambda^k].\tag{13}$$

Finally, we turn our attention to computing the *unconditional* probability of  $A_n$ . Conditioning on  $A_1$ , we write

$$\begin{aligned}
 \Pr[A_n] &= \Pr[A_n|A_1] \Pr[A_1] + \Pr[A_n|\bar{A}_1] \Pr[\bar{A}_1] \\
 &= p_1 \rho_{n-1}(s, s) + (1 - p_1) \rho_{n-1}(u, s) \\
 &= \rho_{n-1}(u, s) + p_1 [\rho_{n-1}(s, s) - \rho_{n-1}(u, s)] \\
 &= \left( p_1 - \frac{\mu}{1 - \lambda} \right) \lambda^{n-1} + \frac{\mu}{1 - \lambda},
 \end{aligned} \tag{14}$$

which, amazingly, matches the expression of  $\Pr[A_n]$  derived in (5). Equations (5) and (14) provide us with the probability of a mission of  $n$  tasks being successful, given the relationship between adjacent tasks. This is very helpful in designing a UAV mission. Obviously, as the number of tasks in a mission increases, the probability of its success decreases, depending on the interdependence between the tasks.

## 6. Sensors utilized in autonomous UAV missions

This section discusses various sensors built into the UAV that enable it to operate over any type of area and in all weather conditions. Proper choice of sensor configuration is utilized to improve mission success by ensuring accurate navigation, target identification, engagement accuracy, and damage assessment [15]. The authors in [16] present a comprehensive review of sensors.

- **GPS sensor:** Offers current location information necessary for positioning and navigation. high precision in height, longitude, and latitude determination. During the entire mission, it assists the UAV with navigation and spatial orientation.
- **Accelerometer:** This device calculates the UAV's acceleration, which helps with an investigation of flight dynamics and stability. important for keeping an eye on the UAV's required motion profile during takeoff, in-flight adjustments, strike execution, and landing.
- **Gyroscope:** Maintains the UAV's angular velocity and orientation. Provides stability in flight and accurate targeting when carrying out the attack.
- **Battery sensor:** Keeps track of the health and charge level of the battery. It ensures the UAV has enough power to complete the job, which is essential for the accomplishment of extending operations.
- **Electro-Optical sensor:** During the day, it records visual imagery with excellent resolution. important for damage assessment, target confirmation, and observation.
- **Infrared sensor:** Provides thermal imaging, which is especially useful in low-light circumstances. Captures heat signatures to enable target detection and surveillance during night missions.
- **Synthetic Aperture Radar (SAR):** Provides terrain analysis and change detection. Provides vital data on geographical features and environmental changes, regardless of weather or light availability.
- **High-Resolution camera:** Provides detailed visual data for surveillance and target identification. It helps identify the target and assess post-strike damage.
- **Anemometer:** Determines direction and speed of wind. Provides information to the UAV's navigation and stability systems, which can then adapt to the wind to improve flying accuracy.

The incorporation of several sensors into the UAV platform provides a robust system capable of operating in difficult situations and carrying out crucial missions with great precision and minimal collateral damage. The sensor suite not only helps with the primary goal of target neutralization but also ensures the UAV's safety and efficiency during the mission duration [17,18].

## 7. Machine Learning Methodology

Autonomous drones for military tasks show potential when using ML algorithms. Although Artificial Intelligence (AI) has not been widely used in battlefield settings, the consensus among

military analysts is that AI technologies (including ML) could have a significant impact on future wars [3,19]. Indeed, advanced ML algorithms can learn from past experience, adapt to novel conditions, and make *correct* decisions autonomously. All of this enhances the drones' ability to execute complex tasks and navigate challenging environments. Furthermore, the capacity of ML to process and interpret vast amounts of data in real-time can enhance the situational awareness of autonomous drones, thereby improving their precision and efficiency [20].

We used analytical expressions to figure out how likely it is that a UAV mission will succeed in the previous section. In this section, we use a ML model [3,19] that can be trained on data from previous training missions, put on the UAV, and used to make decisions in real-time about its final mission. The drones' situational awareness, accuracy, and efficiency are improved because of ML's ability to manage and analyze enormous data volumes in real-time [20]. In this work, we have used a Random Forest Model to identify successful UAV missions based on the provided features of the mission tasks.

### 7.1. Random Forest Model

For the analysis of our UAV mission dataset, we chose Random Forest, a powerful machine learning classifier introduced by [21]. This model uses a majority vote approach to assign a class based on the predictions of many decision trees. We selected the Random Forest (RF) technique because of its better capability to handle high-dimensional data and resistance to over-fitting, which is critical given the multidimensional nature of our synthetic dataset. As indicated, the dataset includes a wide range of features from multiple sensors, each of which contributes complex data about the UAV's performance across various mission tasks.

Our Random Forest model implementation relies on the Scikit-learn package [22], which is well-known for its efficiency and applicability in machine learning tasks. The number of decision trees in our model was set to  $n = 500$  when it was built. This value provides an appropriate balance between computing efficiency and model accuracy. To prevent over-fitting, the maximum depth for each tree was limited to five levels.

The dataset was randomly divided into two parts for training and validation: 80% for training and 20% for testing. The strength of a random forest lies in its ability to utilize the data discrimination capabilities of individual trees, creating an effective classification model. This feature is very useful for our dataset, which consists of  $P$  data points and  $Q$  characteristics and covers a wide range of mission-specific metrics and sensor readings [23]. By combining decisions from several trees, the Random Forest model is better able to handle the complicated and possibly non-linear relationships in our dataset. This dataset includes mission-specific metrics (like success rates for each task) and multiple sensor readings (like GPS coordinates, battery levels, and environmental sensors).

### 7.2. Data Description

Unmanned aerial vehicle (UAV) training exercises are critical for preparing UAVs for real-world missions. During these training sessions, UAVs are deployed to carry out simulated missions that closely match actual military conditions. A critical component of that process is the thorough collection and analysis of data by human professionals. From takeoff to return to base, these specialists carefully monitor and assess a wide range of parameters linked to each task of their mission. They meticulously calculate a success ratio for each mission, capturing the effectiveness and precision of the UAV's performance in various environments. The cumulative assessment of these task-specific success rates, together with a thorough review of the mission's overall execution, helps the human expert assess whether the operation may be classified as successful or unsuccessful. This detailed, expert-driven analysis is critical to improving UAV capabilities and ensuring their ability to prepare for real deployment.

Due to the absence of military UAV sensor data, a synthetic dataset has been created to precisely simulate these sensor readings. Figure 2 shows a detailed overview of a simulated dataset of the UAV

training missions. The work addresses the lack of real-world operational data in unmanned aerial vehicle missions, particularly in sensitive operations such as high-value target eradication.



**Figure 2.** Comprehensive overview of UAV training mission data

The synthetic dataset aims to capture realistic mission situations with feature patterns and relationships based on actual drone mission characteristics. It includes a variety of characteristics of a UAV's functioning, including task-related variables, environmental conditions, and performance metrics. For example, "GPS\_Latitude" and "GPS\_Longitude" offer geographical positioning, while "Battery\_Level" and "AI\_Decision" indicate the drone's operating state and autonomous decision-making ability, respectively. It also shows the task success ratio, offering insight into the UAV's performance and operational efficacy throughout the simulated flight.

Each mission phase, from "takeoff" to "return\_to\_base," is characterized by relevant features. These include environmental sensors like "Electro\_Optical\_Visibility" and "Infrared\_Visibility," essential for understanding the conditions under which the drone operates. The dataset also includes decision points, such as target identification and engagement, based on fused sensor data and predefined rules.

The synthetic data is used as a training dataset for machine learning algorithms, which are designed to predict mission success and uncover important variables influencing results. The UAV system can learn to forecast mission outcomes and optimize its decision-making process in various settings by training the machine learning model on this synthetic dataset.

This dataset<sup>1</sup>, which was created to reflect the complexity of real-world UAV operations, serves as an excellent base for specifically seeking mission outcomes and measuring success indicators. It provides information about the dataset and model availability. This approach allows for risk-free, complete training and evaluation of UAV systems, ensuring capability for a wide range of operational scenarios and improving mission efficacy overall.

### 7.2.1. UAV Operational Data Analysis

In addition to the previously defined parameters and success ratios, the operations of the UAV are closely linked with its responses to sensory input. While the unmanned aerial vehicle (UAV) performs its mission, each task it executes in response to sensor data is fully reported. This methodology can be likened to a "black box" approach, in which each maneuver is recorded, including course corrections, altitude changes, and responses to environmental factors such as wind.

Following the completion of the mission, this detailed record allows the human expert to assess the UAV's behavior in the context of the input parameters at each instant. For example, if the UAV effectively adjusts its altitude under difficult wind conditions, the expert would consider the task a success based on the UAV's adept flexibility with environmental input. A failure to adapt or an inaccurate reaction, on the other hand, would be recorded as unsuccessful, providing significant insights into potential areas for development in UAV programming and decision-making algorithms.

Furthermore, our approach takes into account the future employment of smart weaponry to give rapid and clear evidence of mission success, particularly in activities such as strike execution. Smart missiles with on-board computers can transmit real-time data and photographs as they approach and strike their target. This advanced equipment provides a more direct and reliable technique for confirming target impact than traditional methods such as post-mission reports or external sources such as spy satellites or ground operations.

However, it is important to point out that the use of smart weapons for mission success confirmation is only one component of a larger scheme. Alternative verification methods, such as ground reports or satellite imaging, are considered valid in scenarios where smart weapons are not employed or available.

This enhanced approach to data analysis and mission evaluation corresponds to the growing nature of UAV technology and military methods. We aim to provide an integrated view of UAV mission success and its drivers by combining both traditional data analysis methods and new smart weapons capabilities.

## 7.3. Model Evaluation and Results

The evaluation of classification models includes a wide range of metrics [25], each providing distinct perspectives on the model's performance. These metrics are of the utmost significance in determining the model's efficacy, especially in situations involving particular demands such as class imbalance or varying costs linked to various types of classification errors.

### 7.3.1. Evaluation Metrics

In the following, we explain the metrics for evaluating the classification models [26].

Accuracy is the most straightforward metric. The metric shows the ratio of accurate projections (including true positives and true negatives) to the overall count of cases analyzed.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (15)$$

---

<sup>1</sup> The dataset and the model are provided in the following: [24]

Precision is crucial when the cost of a false positive is large. In such cases, it is critical to reduce the rate of false positives to avoid the potentially negative implications of wrong positive classifications. As a result, in situations where the implications of mistaking a negative instance for a positive instance are severe, precision becomes a more essential measure than just improving total accuracy.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (16)$$

Recall (sensitivity) is especially important when missing a positive instance (false negative) has a high cost. In such cases, identifying as many true positive occurrences as possible is critical, even if it results in a higher number of false positives. A strong recall is critical in situations where the consequences of missing a positive case are severe, exceeding the disadvantages of false positive errors.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (17)$$

The F1-score is especially useful when precision and recall must be balanced. In a classification task, for example, where both false positives and false negatives are costly, the F1-score gives a single statistic that balances these two characteristics. It is the harmonic mean of precision and recall, which ensures that both metrics contribute equally to the overall score.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

The AUC is a common machine learning statistic for binary classification issues. A higher AUC value generally implies a better model because it demonstrates that the model can distinguish between positive and negative classes across all feasible thresholds. This is especially useful for analyzing models in cases where the ideal classification threshold is unknown and must be altered based on the specific costs or benefits associated with true positives, false positives, true negatives, and false negatives.

### 7.3.2. Evaluation and results

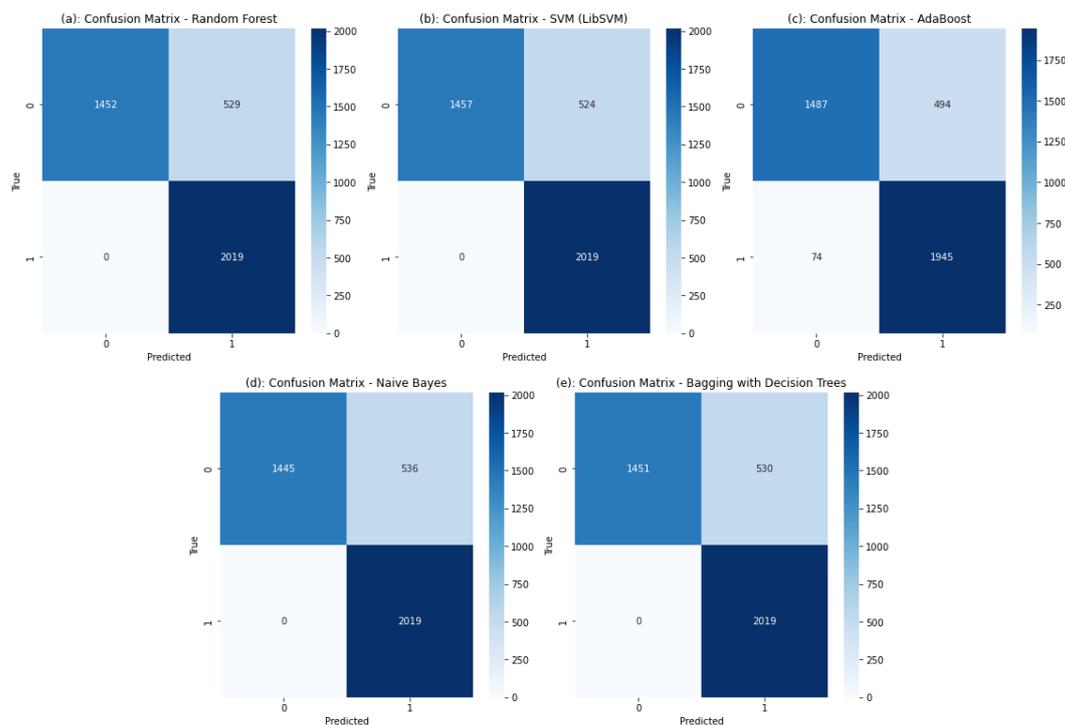
We assessed the performance of our model random forest using the previously described measures. Its overall efficacy can be seen by the large number of outcomes it correctly predicted, with an accuracy of 0.87. Its precision of 0.79 and perfect recall of 1.00, in particular, show that it can effectively detect positive outcomes while reducing false positives. Its F1 score of 0.88 further illustrates this performance balance.

We also compared our model Random Forest to other classifiers; SVM (LibSVM), AdaBoost, Naive Bayes, and Bagging all show a similar pattern of accuracy, with each model obtaining an accuracy of 0.87. Table 1 shows the outcome of these models. Random Forest, SVM (LibSVM), Naive Bayes, and Bagging with Decision Trees all have similar precision values of 0.79; however, AdaBoost has a slightly higher precision of 0.80. Random Forest, SVM (LibSVM), Naive Bayes, and Bagging with Decision Trees all maintain a perfect recall score of 1.0, but AdaBoost has a slightly lower recall of 0.96. All models had similar F1 scores, indicating a fair trade-off between recall and precision. AdaBoost trails slightly behind with an F1 score of 0.87, while the Random Forest model, SVM (LibSVM), Naive Bayes, and Bagging with Decision Trees also have an F1 score of 0.88. These results show that all classifiers function well, with just a little difference in metrics; however, the Random Forest model shows a slight edge, especially in terms of precision and recall.

**Table 1.** Performance Metrics of Classification Models.

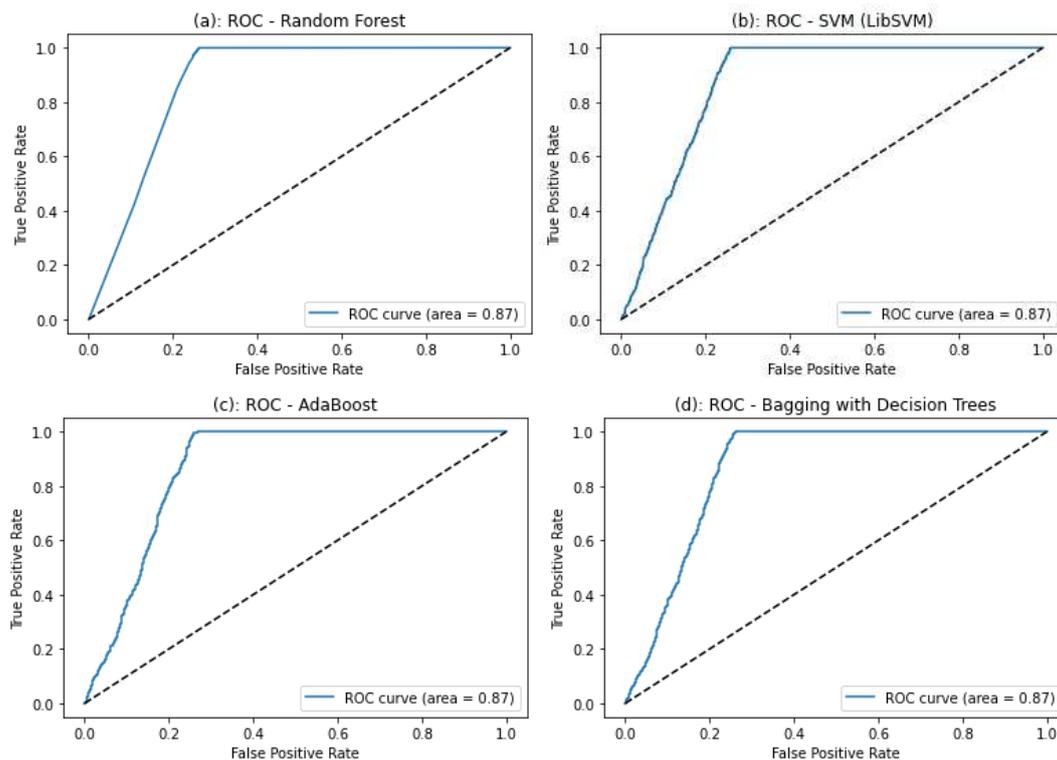
Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.87	0.79	1.00	0.88
SVM (LibSVM)	0.87	0.79	1.00	0.89
AdaBoost	0.86	0.80	0.96	0.87
Naive Bayes	0.87	0.79	1.00	0.88
Bagging	0.87	0.79	1.00	0.88

We evaluated the effectiveness of the models for predicting the outcome of UAV missions, we noticed significant differences in their confusion matrices. The Random Forest, SVM (LibSVM), Naive Bayes, and Bagging with Decision Trees models show exceptional accuracy in correctly identifying successful missions, as indicated by the absence of any false negatives in their confusion matrices. The Random Forest algorithm produces a confusion matrix of  $[[1452, 529], [0, 2019]]$ , the SVM (LibSVM) algorithm shows a confusion matrix of  $[[1457, 524], [0, 2019]]$ , the Naive Bayes algorithm provides a confusion matrix of  $[[1445, 536], [0, 2019]]$ , and the Bagging with Decision Trees algorithm produces in a confusion matrix of  $[[1451, 530], [0, 2019]]$ . This shows an increased ability to identify successes in missions. Nevertheless, these models demonstrated a significant amount of false positives, suggesting an ability to frequently expect success. On the other hand, the AdaBoost model shows an equal performance in detecting both successful and unsuccessful missions, as evidenced by its confusion matrix of  $[[1487, 494], [74, 1945]]$ . Although many cases of false negatives, this model had a reduced rate of false positives in comparison to the other models. This balance indicates a more precise ability of the mission parameters. Figure 3 shows the confusion matrices for the five different classification models.



**Figure 3.** Confusion matrices for five different classification models. (a) Random Forest model. (b) SVM (LibSVM) model. (c) AdaBoost model. (d) Naive Bayes model. (e) Bagging with Decision Trees model.

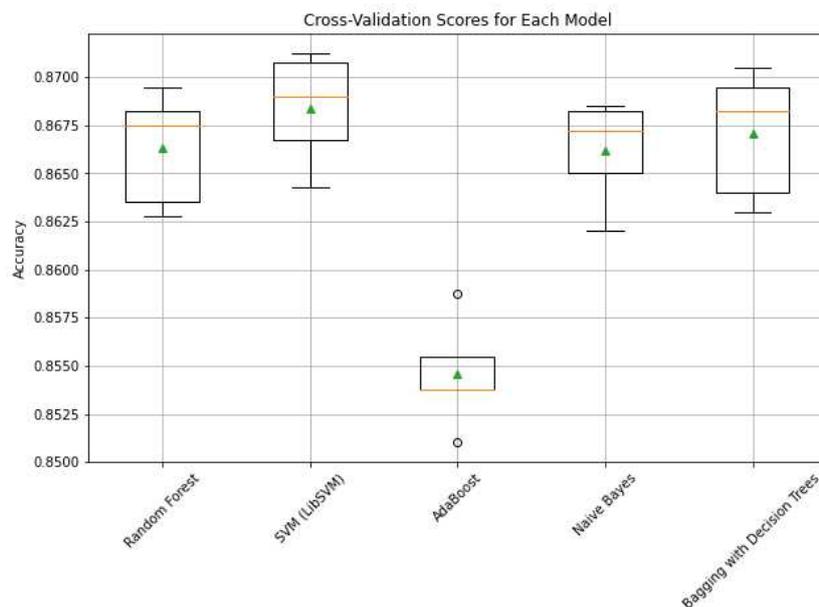
We also evaluated our models using Receiver Operating Characteristic (ROC) curve, which is an essential element in the evaluation of classification models since it shows how well the model can differentiate between classes. Figure 4 shows the Receiver Operating Characteristic (ROC) curves for the Random Forest, SVM (LibSVM), AdaBoost, and Bagging with Decision Trees models. The models Random Forest, SVM (LibSVM), AdaBoost, and Bagging with Decision Trees all showed ROC AUC (Area Under the Curve) values of 0.87, which is a clear indication of their high degree of classification effectiveness and ability to distinguish between positive and negative classifications. The Naive Bayes model was excluded from the ROC AUC analysis as our implementation didn't work to provide the probability estimates required for ROC curve creation.



**Figure 4.** Receiver Operating Characteristic (ROC) curves for four different classification models. (a) ROC curve for the Random Forest model (b) ROC curve for the SVM (LibSVM) model (c) ROC curve for the AdaBoost model (d) ROC curve for the Bagging with Decision Trees model

We performed a thorough cross-validation evaluation to assess the efficacy of five distinct classification models: Random Forest, SVM (LibSVM), AdaBoost, Naive Bayes, and Bagging with Decision Trees. To provide accurate and trustworthy assessments, the cross-validation procedure was performed for five different runs. We created a box plot Figure 5 by combining the data from each of the five runs to contrast the performance of these models. This box plot clearly illustrates the distribution of accuracy ratings for each model while also illuminating the robustness and variability of the various models.

The box plot illustrates the cross-validation results, which are as follows: The Random Forest model demonstrated consistently high performance across the runs, as shown by its mean accuracy scores, which varied from 0.8555 to 0.8695. The accuracy ratings of the SVM (LibSVM) model ranged from 0.8538 to 0.8635, suggesting a consistent and similar level of performance. The scores for the AdaBoost model ranged from 0.8538 to 0.8708, demonstrating its efficacy across multiple iterations. The Naive Bayes model demonstrated competitive prediction skills with scores between 0.8588 and 0.8712. Finally, the Bagging with Decision Trees model produced scores ranging from 0.8658 to 0.8510, indicating that it is a reliable classifier.



**Figure 5.** Box Plot of Classification Model Cross-Validation Results. The distribution of accuracy scores from five-fold cross-validation for five different classification models is shown in this figure: AdaBoost, Random Forest, SVM (LibSVM), Naive Bayes, and Decision Tree-Based Bagging.

## 8. Concluding remarks

In this paper, we have identified the challenges related to enabling autonomous drones to carry out strike missions against high-value terrorists. We have shown that by enlisting the help of the latest technology, including blockchain technology, smart contracts, and machine learning, all these challenges can be overcome. We have suggested that recent developments in ledger technology, smart contracts, and ML enable autonomous UAVs to successfully complete these missions. We have derived analytical expressions for the success of a mission depending on the interdependence of tasks within the mission. Last but not least, we have demonstrated an ML framework for autonomous drones.

A number of issues are still open and are getting attention. One of them is security. While we have developed our mission with minimal communication requirements, in the future communications may play an important role [27,28], and ensuring a high-level of security will turn out to be essential. Yet another open problem is the type of communications and local processing that the drone must perform [29] while in flight.

**Author Contributions:** Conceptualization, R.M. and S.O.; methodology, M.A., R.M. and S.O.; software, M.A.; validation, M.A., R.M. and Z.Z.; formal analysis, R.M. and S.O.; writing—original draft preparation, M.A., R.M. and S.O.; writing—review and editing, M.A., R.M. and S.O.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** The authors declare no conflict of interest.

## References

1. DoD Dictionary of Military and Associated Terms as of March 2017. 2017.
2. Ahmed, F.; Mohanta, J.C.; Keshari, A. Recent Advances in Unmanned Aerial Vehicles: A Review. *Arab Journal of Science and Engineering* **2022**, *7*, 7963–7984.
3. Hoehn, J.R.; K., K.P. Unmanned Aircraft Systems: Current and Potential Programs. In Proceedings of the Congressional Research Service Report CRS R47067, February 2022.

4. Hoehn, J.R. Precision-Guided Munitions: Background and Issues for Congress. In Proceedings of the Congressional Research Service Report CRS R45996, October 2020.
5. Hoehn, J.R.; DeVine, M.F.; Sayler, K.M. Unmanned Aircraft Systems: Roles, Missions, and Future Concepts. In Proceedings of the Congressional Research Service Report CRS R47188, July 18 2022.
6. Schneider, J.; MacDonald, J. Why troops don't trust drones: The 'Warm Fuzzy' Problem. *Foreign Affairs* **2017**.
7. Andersen, C.; Balir, D.; Byrnes, M. Trust, Troops and Reapers: Getting 'Drone' Research Right. *War on the Rocks* **2018**.
8. Harrison, D. Rethinking the Role of Remotely Crewed Systems in the Future Force. *Center For Strategic and International Studies* **2021**.
9. Fox, D.; Burgard, W.; Thrun, S. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research* **1999**, *11*, 391–427.
10. Pandey, S.K.; Zaveri, M.A.; Choksi, M.; Kumar, J.S. UAV-based Localization for Layered Framework of the Internet of Things. *Procedia Computer Science* **2018**, *143*, 728–735. Proc. 8-th International Conference on Advances in Computing and Communications (ICACC-2018), <https://doi.org/https://doi.org/10.1016/j.procs.2018.10.442>.
11. Zhao, B.; Chen, X.; Zhao, X.; Jiang, J.; Wei, J. Real-Time UAV Autonomous Localization Based on Smartphone Sensors. *Sensors* **2018**, *18*.
12. Li, Y.; Yu, R.; Zhu, B. 2D-Key-Points-Localization-Driven 3D Aircraft Pose Estimation. *IEEE Access* **2020**, *8*, 181293–181301. <https://doi.org/10.1109/ACCESS.2020.3026209>.
13. Espinosa, P.; Luna, M.A.; de la Puente, P. Performance Analysis of Localization Algorithms for Inspections in 2D and 3D Unstructured Environments Using 3D Laser Sensors and UAVs. *Sensors* **2022**.
14. Yousaf, J.; Ziai, H.; Alhalabi, M.; Yaghi, M.; Basmaji, T.; Shehhi, E.A.; Gad, A.; Alkhedher, A.; Ghazal, M. Drone and Controller Detection and Localization: Trends and Challenges. *Applied Sciences* **2022**.
15. Chen, J.; Johnsson, K.H.; Olariu, S.; Paschialidis, I.; Stojmenovic, I. Guest editorial, special issue on wireless sensor and actuator networks. *IEEE Transactions on Automatic Control* **2011**, *56*, 2244–2246.
16. Javaid, M.; Haleem, A.; Rab, S.; Singh, R.P.; Suman, R. Sensors for daily life: A review. *Sensors International* **2021**, *2*, 100121.
17. Olariu, S.; Xu, Q.; Zomaya, A. An energy-efficient self-organization protocol for wireless sensor networks. In Proceedings of the Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004., 2004, pp. 55–60. <https://doi.org/10.1109/ISSNIP.2004.1417437>.
18. Rizvi, S.R.; Zehra, S.; Olariu, S. ASPIRE: An Agent-Oriented Smart Parking Recommendation System for Smart Cities. *IEEE Intelligent Transportation Systems Magazine* **2019**, *11*, 48–61. <https://doi.org/10.1109/MITS.2018.2876569>.
19. Konert, A.; Balcerzak, T. Military autonomous drones (UAVs)-from fantasy to reality. Legal and Ethical implications. *Transportation research procedia* **2021**, *59*, 292–299.
20. Suresh, A. Machine learning – IEEE PES Dayananda Sagar College OF Engineering, Bangalore. <https://edu.ieee.org/in-dscepces/2019/12/11/machine-learning/>. (Accessed on 05/08/2023).
21. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
22. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **2011**, *12*, 2825–2830.
23. Jain, V.; Phophalia, A. M-ary Random Forest-A new multidimensional partitioning approach to Random Forest. *Multimedia Tools and Applications* **2021**, *80*, 35217–35238.
24. Aljohani, M. UAVs. <https://github.com/meshari-aljohani/UAVs>, 2023. (Accessed on 05/20/2023).
25. Sujatha, P.; Mahalakshmi, K. Performance evaluation of supervised machine learning algorithms in prediction of heart disease. In Proceedings of the 2020 IEEE international conference for innovation in technology (INOCON). IEEE, 2020, pp. 1–7.
26. Classification | Machine Learning | Google for Developers. <https://developers.google.com/machine-learning/crash-course/classification/video-lecture>.
27. Jones, K.; Wadaa, A.; Olariu, S.; Wilson, L.; Eltoweissy, M. Towards a new paradigm for securing wireless sensor networks. In Proceedings of the Proc. of the 2003 ACM Workshop on New Security Paradigms, Ascona, Switzerland, 2003; pp. 115–121.

28. Rawat, D.B.; Bista, B.B.; Yan, G.; Olariu, S. Vehicle-to-Vehicle Connectivity and Communication Framework for Vehicular Ad-Hoc Networks. In Proceedings of the 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems, 2014, pp. 44–49. <https://doi.org/10.1109/CISIS.2014.7>.
29. Nakano, K.; Olariu, S.; Schwing, J.L. Broadcast-efficient protocols for mobile radio networks. *IEEE Transactions on Parallel and Distributed Systems* **1099**, *10*, 1276–1289.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.