algorithms

MDPI

Article

Blockchain PoS and PoW Consensus Algorithms for Airspace Management Application to the UAS-S4 Ehécatl †

Seyed Mohammad Hashemi *, Ruxandra Mihaela Botez ‡ and Georges Ghazi

Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE, École de Technologie Supérieure (ÉTS), Université de Québec, Montreal, QC H3C 1K3, Canada; ruxandra.botez@etsmtl.ca (R.M.B.); georges.ghazi@etsmtl.ca (G.G.)

* Correspondence: seyed-mohammad.hashemi.1@ens.etsmtl.ca

† This Paper is an Extended Version of the Conference Paper Published in the AIAA AVIATION 2023 Forum (San Diego, California & Online, USA, 12–16 June 2023).

‡ Canada Research Chair Tier 1 Holder in Aircraft Modeling and Simulation, Head of the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE, École de Technologie Supérieure (ÉTS), Université de Québec, Montreal, QC H3C 1K3, Canada.

Abstract: This paper introduces an innovative consensus algorithm for managing Unmanned Aircraft System Traffic (UTM) through blockchain technology, a highly secure consensus protocol, to allocate airspace. A smart contract was developed on the Ethereum blockchain for allocating airspace. This technique enables the division of the swarm flight zone into smaller sectors to decrease the computational complexity of the algorithm. A decentralized voting system was established within these segmented flight zones, utilizing two primary methodologies: Proof of Work (PoW) and Proof of Stake (PoS). By employing 1000 UAS-S4s across various locations and heading angles, a swarm flight zone was generated. The efficiency of the devised decentralized consensus system was assessed based on error rate and validation time. Despite PoS displaying greater efficiency in cumulative probability for block execution, the comparative analysis indicated PoW outperformed PoS concerning the potential for conflicts among UASs.

Keywords: blockchain; airspace management; consensus protocols; conflict probability

Citation: Hashemi, S.M.; Botez, R.M.; Ghazi, G. Blockchain PoS and PoW Consensus Algorithms for Airspace Management Application to the UAS-S4 Ehécatl. *Algorithms* **2023**, *16*, 472. https://doi.org/10.3390/a16100472

Academic Editor: Abdulsalam Yassine

Received: 6 September 2023
Revised: 27 September 2023
Accepted: 2 October 2023
Published: 7 October 2023

1. Introduction

Air traffic management (ATM) involves the coordination of various tasks to allocate airspaces, control traffic flows, and manage aircraft navigation capacity in a manner that is both secure and efficient while also being cost-effective [1]. This framework can be adapted for swarm-style Unmanned Aerial Systems (UASs) that operate at lower altitudes, particularly in congested and high-risk environments.

One of the most pronounced risks is the potential collision with structures such as tall buildings, towers, and power lines. They are at risk of colliding with other aircraft, which can range from other UAVs to manned aircraft. This risk is particularly heightened in congested airspaces [2]. Complicating the matter further, drones at low altitudes in urban environments can experience disruptions in navigation due to the loss of GPS signals caused by tall structures. The communication between the drone and its operator can also be jeopardized by radio frequency interference. Technical malfunctions are always a possibility, which becomes a significant risk in populated areas. On the technology side, drones can also be susceptible to cybersecurity threats, like hacking.

The Federal Aviation Administration (FAA) has established guidelines and rules for the management of Unmanned Aerial Systems Traffic (UTM) [3]. As the utilization of UASs continues to expand, it becomes imperative to advance UTM algorithms to guarantee the safety of flights while also adhering to budget and timeline constraints [4]. Consequently,

the allocation of safe airspace for UASs emerges as a pivotal undertaking that necessitates the reliability of a well-defined algorithm [5].

Many strategies for allocating airspace in UTM have been designed and used [6]. Airspace allocation can be framed as an optimization challenge with the aim of minimizing a cost function that takes into account safety, energy consumption, and time-related aspects [7]. The significance attributed to each factor within the cost function reflects the priorities for the ecosystem. This study concretely concerns safety in scenarios involving multiple aircraft sharing the same airspace, especially in the context of swarm flight zones [8]. In response to these safety considerations, the UTM problem has been reconfigured using Swarm Dynamic Agents (SDAs) as the foundational framework for the most secure management methodology [9].

The management of Swarm Dynamic Agents involves coordinating a group of autonomous agents (vehicles) towards a common goal. Key aspects include agent communication [10], local decision-making [11], proper movements [12], task allocation [13], adaptability, and optimization. Agents communicate and make decisions based on their perception of the environment and interact to move properly. Tasks are assigned based on agent capabilities, and the system must adapt and recover from changes. Optimization and control ensure efficient performance. This approach revolutionized unmanned vehicle movement management, enabling the decentralized accomplishment of complex tasks.

Management methodologies employed for SDAs have been successfully utilized for allocating spaces to various moving vehicles [14]. These methodologies have found application across Unmanned Underwater Systems (UUS) [15], Unmanned Aerial Systems (UAS) [16], Byzantine robots [17], and unmanned cars [18], and their remarkable ability to ensure safety in dynamic environments has been revealed.

With respect to predetermined trajectories for each UAV, dynamic path planning [19] is needed to solve the SDA problem, as the trajectory of each vehicle can affect the trajectory of the others due to uncertainties. Consequently, there is a need for a consensus algorithm to analyze trajectory for path planning, all the while minimizing the probabilities of conflicts [20] among vehicles. The consensus algorithms that have achieved the greatest success have been established through deterministic, randomized, leader-free, and leader-based methodologies [21]. Consensus algorithms are widespread and used for various purposes. They are frequently applied for data aggregation, clock synchronization, leader election, etc. [22,23]. In cases of asynchronous environments, deterministic strategies [24] might prove inadequate in resolving consensus challenges, particularly when faced with a solitary communication breakdown (link failures between communicating entities). Randomized methodologies are risky for critical tasks such as traffic management due to their association with significant safety concerns [25,26]. A leader-based methodology is more vulnerable to cybersecurity attacks than other methodologies but outperforms other methodologies when security is the main concern. Blockchain provides the most secure framework for the execution of a leader-free consensus algorithm [27].

The P2P blockchain topology is the most reliable coordination methodology that makes decisions based on its decentralized architecture [28]. As the UTM ecosystem requires a consensus mechanism for the reliable allocation of airspaces, the P2P blockchain can perform this task proficiently.

The Linear Consensus Protocol (LCP) is a simple algorithm used in distributed systems [29]. The LCP aims to achieve agreement among participants by progressing through proposal rounds with different phases. It involves the preparation, promise, acceptance, and acknowledgment phases. Participants exchange their messages to gather information, make proposals, and reach a consensus on a value. The challenges of achieving consensus in distributed systems are captured by the famous CAP theorem [30], which states that it's impossible for a distributed system to simultaneously provide consistency (all nodes see the same data), availability (every request receives a response), and partition tolerance (the system continues to operate despite arbitrary message loss or failure of part of the system). The LCP ensures that all participants eventually agree on the same value, even

in the presence of failures or delays. It provides security and availability while simplifying complexity. The LCP, rooted in L1 approximation, balanced asymmetry, absolute infinite flow, and unbounded interaction graphs, has been the most renowned approach for managing SDAs [31]. Nonetheless, the introduction of blockchain technology has exposed vulnerabilities in the LCP, particularly when confronted with soft attacks and hardware failures [32,33].

Utilizing blockchain technology can effectively resolve challenges in managing Swarm Dynamic Agents (SDA), addressing issues such as miscommunications, Sybil, and adversarial attacks [34]. Therefore, we implemented blockchain for UAS management, and the outcomes provided evidence of its enhanced safety compared to the LCP approach [35]. A survey study on safe space allocation in transportation problems using blockchain was performed [36]. Another survey study (focusing on airspace allocation) investigated blockchain methodologies for UTM [37]. Ethereum blockchain was used for Automatic Dependent Surveillance–Broadcast (ADS-B) Air Traffic Management relying on satellite navigation [38]. This approach was able to provide situational awareness, allocate safe airspace, and eventually reduce potential conflicts while overcoming cyber threats. Each of these proposed methodologies used a smart contract, which was based on the Proof of Work (PoW) consensus protocol.

In addition to the PoW consensus protocol, blockchain can be set up based on Proof of History (PoH), Proof of Stake (PoS), Proof of Authority (PoA), and Proof of Time (PoT) for appropriate decision making [39,40]. Each of these techniques has its unique advantages. Indeed, PoW is a consensus algorithm employed in multi-agent systems, necessitating every agent node to validate the accurate completion of the undertaken task [41]. However, the PoW has two major drawbacks: its computational complexity and the lengthy confirmation time. To overcome these drawbacks, the Proof of Stake (PoS) consensus mechanism, which relies on validators that confirm blocks on the blockchain according to their credits, was suggested [42].

In this study, we utilized the blockchain sharding concept for partitioning UAVs. In our previous study, [43] the superiority of the blockchain over the partitioned running of UAVs was proven. We started this study regarding our previous proof and discussed it from different consensus mechanism aspects (PoW and PoS).

The present study has two main objectives. The first objective is to design a reliable smart contract for airspace allocation, and the second is to reconfigure that smart contract with the aim of reducing energy consumption, computational complexity, and validation time. The reliable smart contract is initially designed using the PoW consensus method and is then reconfigured using the PoS consensus method to reduce its computational complexity.

Some of the initial results obtained from error rate and validation time analysis were published as a conference paper in the AIAA Aviation Forum 2023 [44], where a consensus system was developed using Ethereum's blockchain for managing traffic in Unmanned Aircraft Systems (UASs), and to assign safe flying zones for UASs. The airspace was divided into multiple shards, enhancing the efficiency through simpler components. Results indicated that this division method decreased errors and sped up validation. Notably, increasing these shards further enhanced the system's performance. While PoW was found to be more reliable for airspace allocation, PoS offered quicker validation. This article is the developed version of that conference paper. It introduces a novel consensus algorithm and a new partitioning approach and analyses blockchain performance from different points of view.

This paper is organized as follows. Section 2 elaborates on the problem statement and introduces the issues of airspace allocation for Air Traffic Management (ATM). The PoW and PoS consensus methodologies for the airspace allocation are detailed in Section 3, and the simulation results that show the performance of the two methodologies are detailed in Section 4. Finally, Section 5 discusses the benefits and drawbacks of the designed consensus mechanism for airspace allocation and offers some recommendations for future work.

2. Problem Statement

Air Traffic Management (ATM) has traditionally been anchored on centralized systems for the sake of safety and efficiency. However, the advent of technological progress has begun to reshape discussions, putting the feasibility of a distributed ATM ecosystem into focus. Both centralized and distributed models come with their sets of merits and shortcomings. In both approaches, ground stations play a key role in ATM operations. The Air Traffic Management (ATM) ecosystem has three components: Advancement of Air Traffic Services (ATS), Air Traffic Flow Management (ATFM), and Airspace Management (ASM) [45]. Figure 1 shows the ATM ecosystem and the focus of this study.

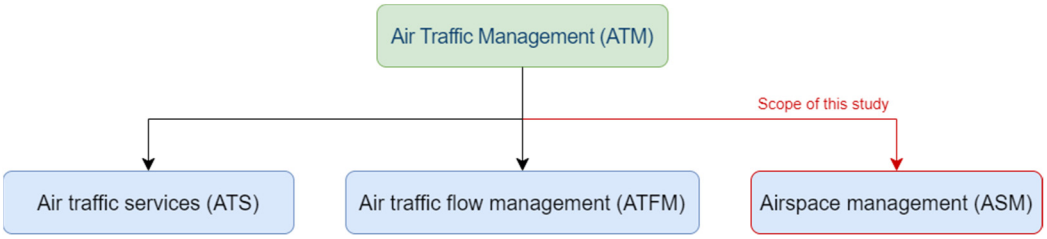

Figure 1. The ATM ecosystem.

The ATS refers to the flight information, air traffic control, and alerting services that prevent collisions between aircraft [46]. The ATFM optimizes the air traffic flow and manages its capacities when demand increases in routes and airdromes, and the ASM is concerned with the allocation of airspaces to aircraft with the aim of reducing the risk of their conflicts [47]. The study presented in this paper focuses on improving the ASM system, and our main objective is to design and develop a reliable and efficient algorithm for airspace allocation and management.

2.1. Airspace Allocation

It is assumed that a flight area is composed of cubic airspaces in which several UASs are flying through their determined trajectories. Figure 2 shows the flight area divided in cubic airspaces $C_{(x,y,z)}$, where u, v, and w denote their number of cubes along the x, y, z axes, respectively.

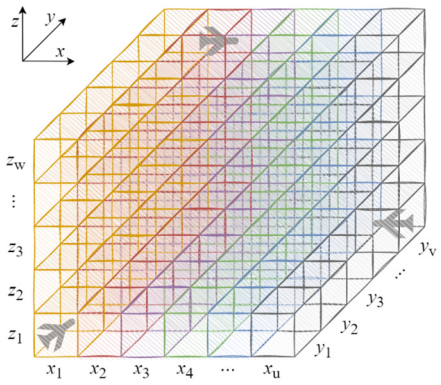

Figure 2. Flight area divided into cubic airspaces [43].

The flight area 3D size is defined by the product $u \times v \times w$, where u, v, and w are positive integers. UASs are located in cubes according to their GPS coordinates (i.e., latitude, longitude, and altitude), and they fly through these cubes following desired trajectories. To avoid conflicts, UASs should never be located in the same cube. Therefore, the supposed ASM must assign a safe cube $C_{(x,y,z)}$ coordinates to each UAS. In our case study, each cube size was considered 100 m $\times$ 100 m $\times$ 100 m in the cubic airspace. The

cubic airspace was constructed by a 16 km^2 squared area and 6 km height (regarding the UAS-S4 operating range).

Relying on the latitude and longitudes of the Earth, the airspaces can be mapped to cubes, and then the cube's coordinates are normalized. We first convert the spherical coordinates (latitude α and longitude β) to Cartesian coordinates for a unit sphere:

$$\begin{cases} x = \cos\alpha.\cos\beta \\ y = \cos\alpha.\sin\beta \\ z = \sin\alpha \end{cases} \quad (1)$$

In order to project these coordinates onto a cube, our approach is a direct mapping wherein you multiply the coordinates based on the furthest point from the origin on the unit sphere to the cube's sides. The method described will project points from the sphere directly outwards to the enclosing cube. Given a point $(\acute{x}, \acute{y}, \acute{z})$ on the unit sphere, the corresponding point on the enclosing cube can be found by

I. Find the axis with the maximum absolute coordinate value:

$$m = \max\left(|\acute{x}|, |\acute{y}|, |\acute{z}|\right) \quad (2)$$

II. Scale the point by the inverse of this value to project it onto the cube:

$$\begin{cases} x = \acute{x}/m \\ y = \acute{y}/m \\ z = \acute{z}/m \end{cases} \quad (3)$$

Cubic airspace considers population and obstacle information that are stored on the blockchain. The verification logic is based on the consensus protocol explained as Algorithm 1, which relies on the conflict probability analyses explained as the conflict solution.

Algorithm 1 Consensus procedure

Input: Application for propagation of blocks to all UASs
Output: Application is accepted/refused
Confirmation process of consensus algorithm ()
    $h = 0$
    waiting time $\geq 2 * T$
    while $h \leq h_{max}$ **do**
        Send agreement to neighbor UASs
        if confirmed by more than 51% nodes **then**
            execute the new block on the chain
            break
        else
            $h = h + 1$
            Choose $R \in \left[0,\ 2^h - 1\right]$ randomly and wait for $R * T$
        end if
    end while
    if $h > h_{max}$ **then**
        airspace rejection, and block execution is refused
    end if
end process

2.2. Communication for a Reliable ASM

Reliable communication is required for effective data sharing among aircraft, and it allows them to make decisions for airspace allocation accordingly. P2P topology is the most efficient communication architecture in terms of reliability [48]. Figure 3 shows the P2P communication between an aircraft and a ground station.

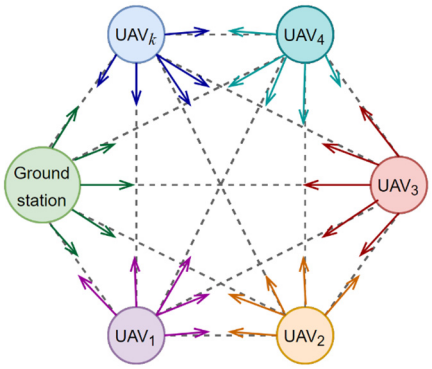

Figure 3. P2P communication between aircraft and a ground station [43].

Figure 3 illustrates a ground station and a k number of UAVs in a flight zone. Each aircraft sends its GPS data directly to all other aircraft, and each receives the GPS data from the others. Therefore, all aircraft are informed of the GPS data corresponding to all of them.

In this methodology, the ground station acts as a supervisor, and each aircraft is considered a local node on the blockchain, which provides a distributed protocol for airspace allocation. Blockchain can serve a large number of local nodes (UAVs), while the limitation arises from other factors (e.g., regulations, size of the flight zone, the purpose of the UAVs, etc.) in advance. Although several ground stations can be involved, relying on the support of the blockchain, one grand station is enough for supervision.

The ground station is the third party in a conventional ASM, which is considered the supervisor. In other words, UAVs perform the full propagation of data sharing by relying on the P2P topology, and the ground station is a redundant validator used to improve reliability and ground supervision. An algorithm (a smart contract), using shared GPS data, evaluates the aircraft trajectories, makes decisions, and then allocates safe airspaces. This process is shown in Figure 4.

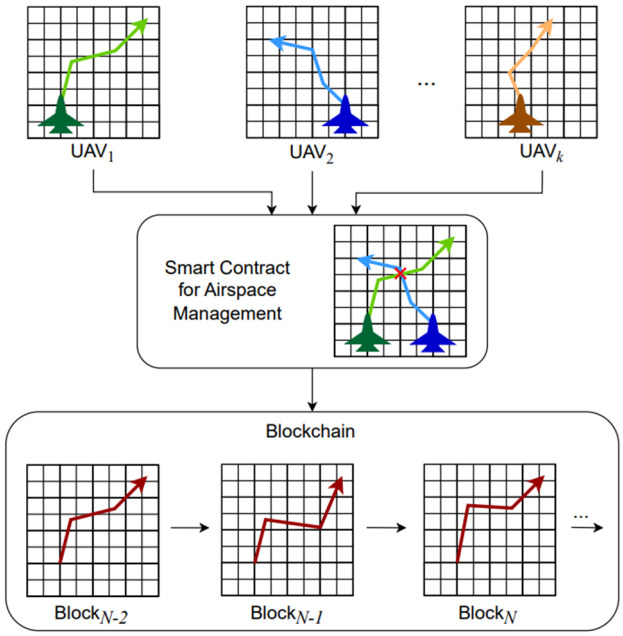

Figure 4. Conflict detection and airspace allocation procedure.

Figure 4 shows the locations and trajectories for the k number of UAVs collected by a smart contract. An imaginary mesh was used to create sectors along the x and y axes at a specific altitude. Aircraft are flying through predetermined trajectories while safe sectors are allocated to them. The smart contract collects data containing allocated sectors

to all UAVs. Its algorithm detects potential conflicts and allocates safer sectors to UAVs such that the risk of conflict is minimized. Allocated airspaces conduct a safe path for each UAS at each timestamp. A timestamp is a digital record that indicates the date and time at which a specific event (airspace allocation) was created, modified, or recorded. It serves as a chronological reference point, often represented in a standardized format, to establish the sequence and timing of events. Then, the conduct-safe trajectories are stored in a blockchain and are used to compute airspaces in further steps. The smart contract algorithm and the associated blockchain for ASM are explained in the following section.

3. Methodology

It is assumed that a ground station is in charge of controlling n UASs. Raising the quantity of UASs within a designated airspace increases the probability of encountering conflicts and collisions [49]. Better airspace management is needed to allow UASs to fly safely. A smart contract was developed for this airspace management, implemented by incorporating sharding, with the flight zone composed of m districts, as depicted in Figure 5.

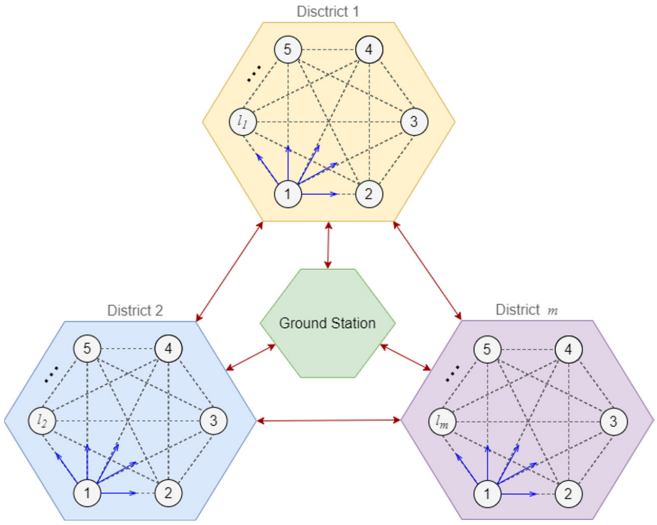

Figure 5. A sharded flight zone set up for the Unmanned Traffic Management (UTM) system.

Figure 5 depicts a P2P communication structure among UASs. The flight area is partitioned into m districts, with each district forming a sub-flight zone hosting l_m number of UASs (where l number of UASs are flying in a district m). Relying on the sharding concept [50], which involves scaling the blockchain network, a smart contract can categorize aircraft flights into various sub-flight zones (districts) based on their respective locations. The districts are considered to be the same size. A flight zone sharding can reduce the computational complexity raised by the smart contract. Moreover, the ground station does not need to communicate directly with aircraft; instead, it relies on the smart contract obtained from each individual district for its required data.

A smart contract on the blockchain is a self-executing digital agreement where the conditions and terms are embedded within lines of code. These contracts automatically enforce and trigger actions when predefined criteria are satisfied, eliminating the need for intermediaries and ensuring a transparent, tamper-proof process. In the realm of space allocation, smart contracts can be harnessed to manage and allocate spaces based on specific rules [51].

In the context of a sharding flight zone, the utilization of a smart contract is pivotal for establishing a consensus framework. Consequently, a voting platform was devised utilizing the Ethereum blockchain [52] to ensure the secure allocation of airspace to Unmanned Aerial Systems (UASs) and to prevent conflicts. Figure 6 illustrates the process validating the secure allocation of airspaces to UASs.

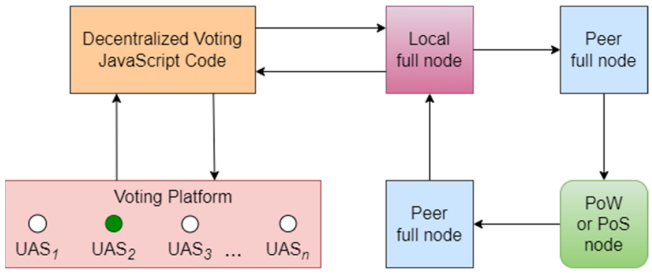

Figure 6. Airspace allocation using the voting approach validated through the PoW or PoS [44].

Figure 6 shows a decentralized process for airspace allocation and validation. Within this process, the voting platform gathers votes from UASs, which include feasible airspaces for projected flight paths. Subsequently, these collected votes [53] are transmitted to the decentralized JavaScript voting code, which relies on the smart contract. The allocated airspace is then sent to the local full node associated with the chosen district. The confirmed votes for airspace allocation are distributed to the peer full node linked to the corresponding district. The peer full node performs an identical validation process, followed by the submission of votes to PoW or PoS validators, contingent upon the chosen PoW/PoS strategy. The PoW/PoS protocol node generates a new block and transmits it to the peer full node. The local full node accepts the verified block from the peer full node and forwards it back to the decentralized voting code. Ultimately, the consensus framework ensures the safety of the allocated airspaces to all UASs.

A UAV flight may encounter various uncertainties [54] that require resolution advice and lead to trajectory modifications, which, directly and indirectly, may affect other UAVs' trajectories. The consensus mechanism should, therefore, update agreements and allocate safe airspaces accordingly.

The most notable feature of the blockchain methodology is its priority level consideration, which allows blockchain-based consensus protocols to perform agents' agreements while considering their corresponding priority levels. Figure 7 illustrates the confirmation process for airspace allocation for trajectory execution.

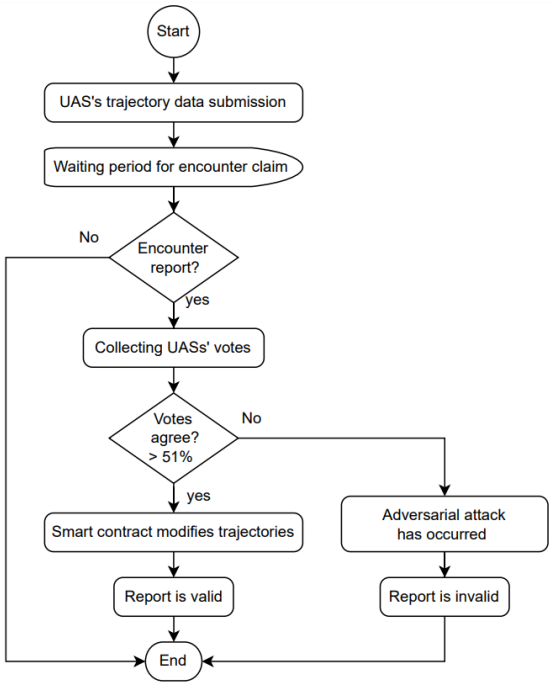

Figure 7. Airspace arrangement and resolution.

It is assumed that UASs are considered in a flight zone while, simultaneously, their trajectory data [55] are recorded on the blockchain. During the timestamp for submitting updated trajectories, any node associated with a UAS may report a conflict with another UAS. If an encounter situation is claimed, a vote collection is needed; otherwise, UASs are not in encounter situations. When a conflict is reported, all nodes check the stored trajectory data (on the blockchain) corresponding to the UASs. If more than 51% of nodes confirm the conflict, then the smart contract must rearrange allocated airspaces. Otherwise, the report is assumed to be invalid due to an adversarial attack [56]. Algorithm 1 explains the consensus procedure in detail.

The blockchain propagates blocks containing the registered trajectory data to all UASs. As explained in Algorithm 1 for a consensus confirmation, the waiting time is twice more than the end-to-end delay time (T) at each timestamp to avoid duplicated block generation. As long as UASs respect the boundary on generating the maximum number of blocks (h_{max}), the agreement is sent to neighboring UASs. If more than 51% of nodes approve, the new block on the chain is executed. Otherwise, for a number of $h+1$ generated blocks, a random number (R) is selected such that $R \in \left[0,\ 2^h - 1\right]$, and a waiting time occurs once again for $R * T$. This process continues forward as long as the boundary on generating the maximum number of blocks (h_{max}) is respected. Once this threshold is surpassed, the block execution is refused, and airspace allocation is denied.

4. Results and Discussion

Blockchain-based airspace allocation methodology was tested and validated by relying on our UAS-S4 trajectory database that was generated using a flight dynamics model and its corresponding controller [57]. The UAS-S4 is shown in Figure 8, and Table 1 lists its specifications.

Figure 8. Hydra Technologies UAS-S4 Ehécatl [44].

Table 1. The UAS-S4 geometrical and flight data specifications.

Specification	Value
Wing area	2.3 m^2
Wingspan	4.2 m
Mean aerodynamic chord	0.57 m
Total length	2.5 m
Maximum take-off weight	80 kg
Empty weight	50 kg
Loitering airspeed	35 knots
Maximum speed	135 knots
Operational range	120 km
Service ceiling	15,000 ft

We applied our approach within an aerial region, conducting tests with both 100 and 1000 UAS-S4 under the control of an adaptive fuzzy control algorithm [58]. For experimental studies, we took into account a cubic flight area of x = 16 km, y = 16 km, and z = 6 km. The UAS-S4s were arranged with predetermined latitudinal and longitudinal coordinates,

altitudes, and heading angles. The forthcoming paths of the UAS-S4s were then predicted by an autoencoder [43] while the smart contract was validating the allocated airspaces. The utilized validators were following the PoW and PoS consensus strategy. Table 2 provides a comparison of the performance of both algorithms concerning validation time and error rate.

Table 2. The airspace allocation performance using The PoW and PoS.

Consensus Mechanism	Number of UAS	Number of Shards	Validation Time (ms)	Error Rate %
Proof of Work (PoW)	100	4	21	7.7
		16	18	7.2
	1000	4	420	9.3
		16	365	8.8
Proof of Stake (PoS)	100	4	11	13.1
		16	8	12.6
	1000	4	121	17.6
		16	101	16.9

As shown in Table 2, consensus algorithms, including Proof of Stake (PoS) and Proof of Work (PoW), are considered. We examined airspace areas with 100 and 1000 UAS-S4. We partitioned each airspace into 4 and 16 shards. Based on the findings presented in Table 2, one can deduce that PoW demonstrated greater accuracy compared to PoS, given its lower error rate with an equivalent number of UAS-S4s. However, PoW exhibited longer validation times compared to PoS.

Table 2 illustrates that within a flight region partitioned into four districts and hosting 100 UASs, the error rates were 13.1% for PoS and 7.7% for PoW, while their validation times were 11 ms and 21 ms, respectively. Additionally, it is worth mentioning that an increased quantity of UASs within a flight zone results in higher error rates and longer validation times for a particular consensus mechanism, irrespective of the shard count. For instance, by using the PoS consensus algorithm (considering 16 shards), raising the UAS count from 100 to 1000 leads to a 4.3% increase in the error rate (from 12.6% to 16.9%) and a 93 ms extended validation time (from 8 ms to 101 ms). On the contrary, increasing the shard count (from 4 to 16) and using either PoW or PoS while keeping the UAS quantity constant enhances the algorithm's performance, resulting in decreased error rates and shorter validation times. For example, when employing PoW in a flight area with 1000 UASs, the expansion of shard count from 4 to 16 led to a 0.5% decrease in the error rate (reducing it from 9.3% to 8.8%) and shortened the validation time by 55 ms (from 420 ms to 365 ms). The maximum acceptable error rate was considered 20%. Larger error rates require a reduction in the number of UAS-S4 in districts or an increase in the number of shards.

While PoS outperformed PoW in terms of validation time, their performances in terms of error rates need to be evaluated. With this aim, the probability analysis of conflicts between UASs has been carried out, where the lower probability of conflict means the lower error rate of a consensus protocol.

Incorporating Gaussian distribution probability functions into the geometrical approach of conflict probability estimation offers a nuanced method to predict UAV conflicts while accounting for uncertainties in UAV positions and velocities. The Gaussian model, with its probabilistic nature, allows for a more sophisticated understanding and calculation of potential conflicts, dynamically adjusting to the movements and position updates of the UAVs. However, care must be taken to ensure the assumptions of normality are met and sufficient computational resources are available for real-time calculations.

For considering conflict probability as the performance metric, firstly, it is needed to determine encounter situations between UASs. Figure 9 shows in 2D the conflict between two UASs flying at a specific altitude.

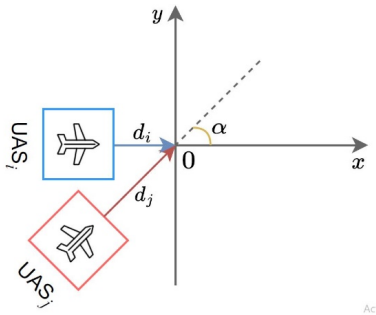

Figure 9. Two-dimensional conflict situation between two UASs [59].

As shown in Figure 9, two aircraft (UAS$_i$ and UAS$_j$) are coming to a probable collision area, where α denotes the difference between their heading angles, and their distance to the collision point 0 is represented as d_i and d_j, respectively.

The Bayesian geometrical approach [59] was used for computing conflict probability between UASs, which is represented by Equation (4):

$$P_{conflict} = \frac{1}{2}\left(f\left(\frac{s-\mu}{\sigma\sqrt{2}}\right) - f\left(\frac{-s-\mu}{\sigma\sqrt{2}}\right)\right) \tag{4}$$

where s is the minimum allowed aircraft horizontal separation in an airspace. The minimum distance is normally distributed with the mean μ and variance σ^2, and f can be obtained from Equation (5) [59].

$$f(x) = \frac{2}{\sqrt{\pi}}\int_0^x exp(-t^2)\,dt \tag{5}$$

The mean μ and variance σ^2 are computed through Equation (6) [59].

$$\mu = \lambda\left(d_j - \rho d_i\right), \;\; \sigma^2 = \lambda^2\overline{a}(t)^2\left(1+\rho^2\right) \tag{6}$$

and Equation (7) [59].

$$\lambda = \frac{\sin(\alpha)}{\sqrt{\rho^2 - 2\rho\cos(\alpha)+1}}, \;\; \rho = \frac{v_j}{v_i} \tag{7}$$

where UAS$_i$ and UAS$_j$ are flying to the conflict point 0 with v_i and v_j speeds; their distance to the conflict point 0 is represented by d_i and d_j, respectively.

While PoS exhibited superior performance in validation time, PoW surpassed PoS in terms of error rate, which resulted in a lower probability of conflict between UASs using the PoW methodology, as presented in Figure 10.

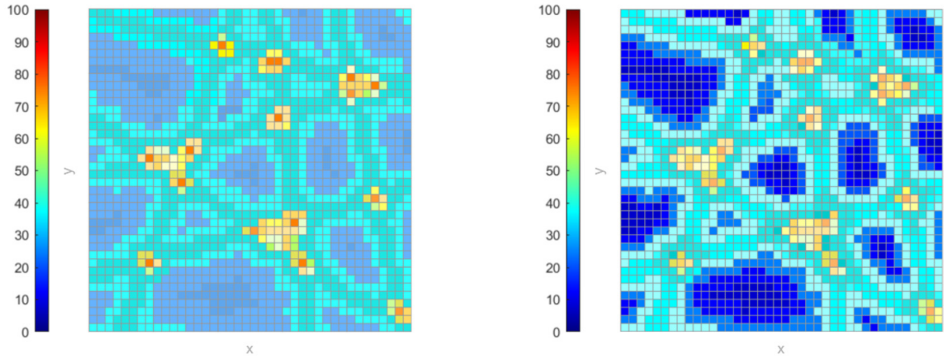

Figure 10. Probability of conflicts according to the PoS (left) and the PoW (right) methodologies [44].

Figure 10 depicts the probability of conflicts among UASs at a specific altitude. Each cube size was considered to be 100 m $*$ 100 m $*$ 100 m in the cubic airspace. One hundred UAS-S4s were arranged at $x = 4 - 8$ km (latitude), $y = 8 - 12$ km (longitude), $z = 3.75$ km (altitude) considering 16 shards. This figure was extracted from the 10th shard, in which 7 UAS-S4s had conflicts in 16 km^2 composed of 1600 airspaces sector (40 sectors along the x-axis and 40 sectors along the y-axis). Comparing the plot on the left (corresponding to PoS) with the one on the right (related to PoW) confirms that PoW outperformed PoS in terms of conflict probabilities and safety. The average conflict probability for the PoS was 34.1%, and for the PoW, it was 25.4%.

The blockchain is a safe consensus protocol that can store trajectory data in an untrusted environment, even when nodes corresponding to new UAVs are added to the P2P topology. Figure 11 illustrates the nodes' (UAVs') agreement approval probability in order to assess the blockchain performance for block execution on the chain. It shows the cumulative execution probability [60] when the number of blocks is increased. The percentile of executed confirmations can be estimated by using the percentile analysis on the generated blocks. This metric is utilized to evaluate the risk of confirmation failures by using the ground station node.

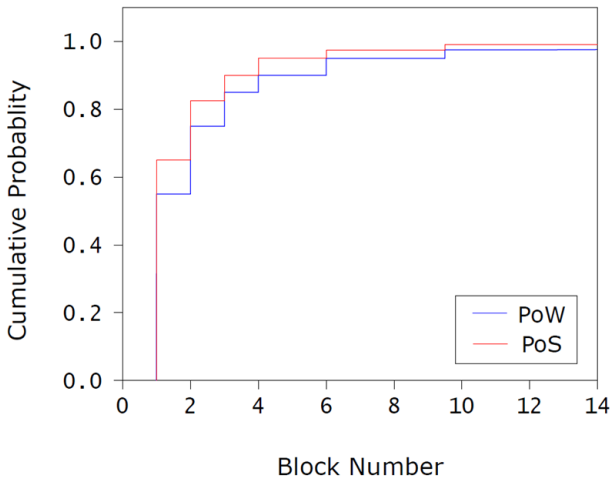

Figure 11. Cumulative execution probability with an increasing number of blocks using the PoW (blue) and PoS (red) consensus protocols.

According to Figure 11, the cumulative probabilities of a successful confirmation are 0.547 (using the PoW consensus protocol) and 0.663 (using the PoS consensus protocol) after their first block generation. After generating four blocks, this value settles at 0.901 (for the PoW) and is higher than 0.953 (for the PoS). The comparison of the cumulative execution probability using the PoW and PoS consensus protocols reveals the superiority of the PoS. The PoS is therefore considered a priority for some UASs to ensure the execution of a new block on the chain with higher probability from this point of view, while the PoW outperforms the PoS in terms of reliability.

The blockchain designed for consensus processing can also be assessed from the perspective of reliability. With this aim, the blockchain's error rates while the blocks are generated were considered as performance indices. The error rates were measured in relationship to the block execution failures while PoW and PoS consensus mechanisms were in operation. This experiment was performed regarding both leader-based and leader-free approaches, in which the leader-free consensus protocol allowed UAS-4s to reach an agreement for airspace allocation without relying on a central authority, and vice versa for the leader-based protocol [61]. The error rates using different consensus protocols and the number of UASs are listed in Table 3.

Table 3. Blockchain reliability analysis according to the control by various consensus mechanisms and with 3 scales of UAS numbers.

	Error Rates %			
Number of UASs	**Leader-Free**		**Leader-Based**	
	PoW	**PoS**	**PoW**	**PoS**
10	0.21	0.43	0.32	0.67
100	0.47	0.88	0.65	1.31
1000	0.96	1.75	1.34	2.71

As shown in Table 3, a flight zone was arranged using three different total numbers of UASs (10, 100, and 1000) for the blockchain reliability analysis. The airspace was partitioned into five shards. The results show that a consensus process with a leader reduces the reliability of both the PoW and the PoS, while leader-free consensus is more reliable as the error rate is higher. The results also reveal that the error rates when using PoW's leader-free consensus mechanism were lower than those of PoS, regardless of the number of UASs in the flight zone. Therefore, the PoW leader-free consensus strategy was better than PoS for airspace allocation.

5. Conclusions

This paper outlines the development of a consensus framework utilizing the blockchain for managing Unmanned Aircraft Systems (UASs) traffic. The smart contract was intended to assign secure airspace to the UASs. The consensus algorithm was formulated by drawing from both Proof of Work (PoW) and Proof of Stake (PoS) algorithms. The airspace was divided into multiple districts, where the smart contract's efficiency could be enhanced through reduced computational complexity.

The findings demonstrated that sharding can lead to a decrease in both the error rate and validation time, underscoring the enhancement in consensus protocol performance with a higher number of shards. Considering conflict probabilities with various consensus protocols, the PoW offered a better safe airspace allocation than the PoS algorithm, in which the average probability of conflict for PoW was 8.7% less than PoS. In fact, the PoW outperformed PoS in terms of error rates. On the other hand, the cumulative execution probability with an increasing number of blocks using the PoS was definitely better than the PoW. Additionally, the PoS could provide a shorter validation time. Without paying attention to the utilized proof methodology, the leader-free approach showed lower error rates compared to those of the leader-based approach.

Author Contributions: Conceptualization, S.M.H.; methodology, S.M.H.; software, S.M.H.; validation, S.M.H., R.M.B., and G.G.; formal analysis, S.M.H. and G.G.; investigation, S.M.H.; resources, R.M.B. and G.G.; data curation, S.M.H. and G.G.; writing—original draft, S.M.H.; writing—review and editing, R.M.B. and G.G.; supervision, R.M.B. and G.G.; project administration, R.M.B.; funding acquisition, R.M.B. and G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Canada Research Chair in Aircraft Modeling and Simulation (NSERC) under the contract number: 231679, which made the realization of this research and the publication of this paper possible. Ruxandra Botez is the Canada Research Chair Tier 1 Holder in Aircraft Modeling and Simulation New Technologies.

Acknowledgments: Special thanks are due to the Natural Sciences and Engineering Research Council of Canada (NSERC) for the Canada Research Tier 1 in Aircraft Modeling and Simulation Technologies funds. We would also like to thank Odette Lacasse and Oscar Carranza for their support at ETS, as well as Hydra Technologies' team members Carlos Ruiz, Eduardo Yakin, and Alvaro Gutierrez Prado in Mexico. Finally, we wish to express our appreciation to the Canada Foundation for Innovation CFI, the Ministère de l'Économie et de l'Innovation, and Hydra Technologies for their support of the acquisition of the UAS-S4 at the LARCASE.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Nomenclature

α	The difference between two UAS's heading angles
$C_{(x,y,z)}$	Cubic airspace regarding the latitude x, longitude y, and altitude z axes
d_i	Distance to the conflict point
h	Number of generated blocks
h_{max}	Boundary on the maximum number of blocks generation
k	Number of Unmanned Aerial Systems in a non-sharded flight zone
m	Number of districts (shards)
l_m	Number of Unmanned Aerial Systems in the m^{th} district
n	Number of Unmanned Aerial Systems in a flight zone
s	Minimum allowed horizontal separation for airspace
T	End-to-end delay
v	UAS's speed when flying toward the conflict point
x	Latitude
y	Longitude
z	Altitude
ATM	Air Traffic Management
ASM	Airspace Management
GPS	Global Positioning System
LCP	Linear Consensus Protocol
PoS	Proof of Stake
PoW	Proof of Work
SDA	Swarm Dynamic Agents
UAS	Unmanned Aerial Systems
UTM	Unmanned Aerial Systems Traffic Management

References

1. Sridhar, B.; Sheth, K.S.; Grabbe, S. Airspace complexity and its application in air traffic management. In *2nd USA/Europe Air Traffic Management R&D Seminar*; Federal Aviation Administration: Washington, DC, USA, 1998.
2. Xiang, J.; Liu, Y.; Luo, Z. Flight safety measurements of UAVs in congested airspace. *Chin. J. Aeronaut.* **2016**, *29*, 1355–1366. [CrossRef]
3. Kopardekar, P.H. *Unmanned Aerial System (UAS) Traffic Management (UTM): Enabling Low-Altitude Airspace and UAS Operations*; NASA: Washington, DC, USA, 2014.
4. Hashemi, S.M. *Novel Trajectory Prediction and Flight Dynamics Modelling and Control Based on Robust Artificial Intelligence Algorithms for the UAS-S4*; École de technologie supérieure: Montréal, QC, Canada, 2022.
5. Ball, M.; Donohue, G.; Hoffman, K. Auctions for the safe, efficient, and equitable allocation of airspace system resources. *Comb. Auction.* **2006**, *1*, 507–538.
6. Liu, J.; Shi, Y.; Fadlullah, Z.M.; Kato, N. Space-Air-ground integrated network: A Survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2714–2741. [CrossRef]
7. Aydoğan, E.; Cetek, C. Aircraft route optimization with simulated annealing for a mixed airspace composed of free and fixed route structures. *Aircr. Eng. Aerosp. Technol.* **2022**, *95*, 637–648. [CrossRef]
8. Brust, M.R.; Danoy, G.; Bouvry, P.; Gashi, D.; Pathak, H.; Gonçalves, M.P. Defending against intrusion of malicious UAVs with networked UAV defense swarms. In Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), Singapore, 9 October 2017.
9. Liu, Y.; Passino, K.M. *Swarm Intelligence: Literature Overview*; Department of Electrical Engineering, the Ohio State University: Columbus, OH, USA, 2000.
10. Izadi, H.; Gordon, B.; Zhang, Y. Safe path planning in the presence of large communication delays using tube model predictive control. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010.
11. Ghommam, J.; Rahman, M.H.; Saad, M. Design of distributed event-triggered circumnavigation control of a moving target by a group of underactuated surface vessels. *Eur. J. Control.* **2022**, *67*, 100702. [CrossRef]
12. Ghommam, J.; Saad, M.; Wright, S.; Zhu, Q.M. Relay manoeuvre based fixed-time synchronized tracking control for UAV transport system. *Aerosp. Sci. Technol.* **2020**, *103*, 105887. [CrossRef]
13. Zhou, X.; Yu, X.; Zhang, Y.; Luo, Y.; Peng, X. Trajectory Planning and Tracking Strategy Applied to an Unmanned Ground Vehicle in the Presence of Obstacles. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1575–1589. [CrossRef]

14. Bouffanais, R. *Design and Control of Swarm Dynamics*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 1.
15. Xiong, M.; Xie, G. Simple agents, smart swarms: A cooperative search algorithm for swarms of autonomous underwater vehicles. *Int. J. Syst. Sci.* **2022**, *53*, 1995–2009. [CrossRef]
16. Liu, G.; Chen, L.; Liu, K.; Luo, Y. A swarm of unmanned vehicles in the shallow ocean: A survey. *Neurocomputing* **2023**, *531*, 74–86. [CrossRef]
17. Wardega, K.; von Hippel, M.; Tron, R.; Nita-Rotaru, C.; Li, W. Byzantine Resilience at Swarm Scale: A Decentralized Blocklist Protocol from Inter-robot Accusations. *arXiv* **2023**, arXiv:2301.06977.
18. Bono, A.; Fedele, G.; Franze, G. A Swarm-Based Distributed Model Predictive Control Scheme for Autonomous Vehicle Formations in Uncertain Environments. *IEEE Trans. Cybern.* **2021**, *52*, 8876–8886. [CrossRef] [PubMed]
19. Fiorentino, F. *Path Planning Strategies for Drone Delivery for Life-Saving Pharmaceuticals*; Politecnico di Torino: Torino, Italy, 2021.
20. Hu, J.; Prandini, M. Aircraft conflict detection: A method for computing the probability of conflict based on Markov chain approximation. In Proceedings of the 2003 European Control Conference (ECC), Cambridge, UK, 1–4 September 2003.
21. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Liu, Y. A Survey on the Scalability of Blockchain Systems. *IEEE Netw.* **2019**, *33*, 166–173. [CrossRef]
22. Kenyeres, M.; Kenyeres, J. Distributed Mechanism for Detecting Average Consensus with Maximum-Degree Weights in Bipartite Regular Graphs. *Mathematics* **2021**, *9*, 3020. [CrossRef]
23. Niu, Y.; Yang, T.; Hou, Y.; Cai, S.; Yan, P.; Li, W. Consensus tracking-based clock synchronization for the Internet of Things. *Soft Comput.* **2022**, *26*, 6415–6428. [CrossRef]
24. Carli, R.; Fagnani, F.; Frasca, P.; Zampieri, S. Gossip consensus algorithms via quantized communication. *Automatica* **2010**, *46*, 70–80. [CrossRef]
25. Zhou, X.; Yu, X.; Guo, K.; Zhou, S.; Guo, L.; Zhang, Y.; Peng, X. Safety Flight Control Design of a Quadrotor UAV With Capability Analysis. *IEEE Trans. Cybern.* **2021**, *53*, 1738–1751. [CrossRef]
26. Cestino, D.; Crosasso, P.; Rapellino, M.; Cestino, E.; Frulla, G. Safety Assessment of Pharmaceutical Distribution in a Hospital Environment. *J. Heal. Technol. Manag.* **2013**, *1*, 10–21. [CrossRef]
27. Lashkari, B.; Musilek, P. A Comprehensive Review of Blockchain Consensus Mechanisms. *IEEE Access* **2021**, *9*, 43620–43652. [CrossRef]
28. Kwilinski, A. Implementation of blockchain technology in accounting sphere. *Acad. Account. Financ. Stud. J.* **2019**, *23*, 1–6.
29. Cepeda-Gomez, R.; Olgac, N. An Exact Method for the Stability Analysis of Linear Consensus Protocols With Time Delay. *IEEE Trans. Autom. Control* **2011**, *56*, 1734–1740. [CrossRef]
30. Gilbert, S.; Lynch, N. Perspectives on the CAP Theorem. *Computer* **2012**, *45*, 30–36. [CrossRef]
31. Bolouki, S. *Linear Consensus Algorithms: Structural Properties and Connections with Markov Chains*; École Polytechnique de Montréal: Montréal, QC, Canada, 2014.
32. Tsiulin, S.; Reinau, K.H.; Hilmola, O.-P.; Goryaev, N.; Karam, A. Blockchain-based applications in shipping and port management: A literature review towards defining key conceptual frameworks. *Rev. Int. Bus. Strat.* **2020**, *30*, 201–224. [CrossRef]
33. van Iersel, Q.G.; Murrieta Mendoza, A.; Felix Patron, R.S.; Hashemi, S.M.; Botez, R.M. Attack and Defense on Aircraft Trajectory Prediction Algorithms. In Proceedings of the AIAA AVIATION 2022 Forum, Chicago, IL, USA, 27 June–1 July 2022.
34. Castelló Ferrer, E. The blockchain: A new framework for robotic swarm systems. In *Proceedings of the Future Technologies Conference (FTC) 2018: Volume 2*; Springer: Berlin/Heidelberg, Germany, 2019.
35. Ismail, L.; Materwala, h. A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. *Symmetry* **2019**, *11*, 1198. [CrossRef]
36. Astarita, V.; Giofrè, V.P.; Mirabelli, G.; Solina, V. A Review of Blockchain-Based Systems in Transportation. *Information* **2019**, *11*, 21. [CrossRef]
37. Alladi, T.; Chamola, V.; Sahu, N.; Guizani, M. Applications of blockchain in unmanned aerial vehicles: A review. *Veh. Commun.* **2020**, *23*, 100249. [CrossRef]
38. Hasin, F.; Munia, T.H.; Zumu, N.N.; Taher, K.A. Ads-b based air traffic management system using ethereum blockchain technology. In Proceedings of the 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 27–28 February 2021.
39. Sarmah, S.S. Understanding blockchain technology. *Comput. Sci. Eng.* **2018**, *8*, 23–29.
40. Mohanta, B.K.; Jena, D.; Panda, S.S.; Sobhanayak, S. Blockchain technology: A survey on applications and security privacy Challenges. *Internet Things* **2019**, *8*, 100107. [CrossRef]
41. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016.
42. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 2017.
43. Hashemi, S.M.; Hashemi, S.A.; Botez, R.M.; Ghazi, G. A novel fault-tolerant air traffic management methodology using autoencoder and P2P blockchain consensus protocol. *Aerospace* **2023**, *10*, 357. [CrossRef]
44. Hashemi, S.; Botez, R.M.; Ghazi, G. Comparison Study between PoW and PoS Blockchains for Unmanned Aircraft System Traffic Management. In Proceedings of the AIAA AVIATION 2023 Forum, San Diego, CA, USA, 12–16 June 2023.

45. Prandini, M.; Piroddi, L.; Puechmorel, S.; Brazdilova, S.L. Toward Air Traffic Complexity Assessment in New Generation Air Traffic Management Systems. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 809–818. [CrossRef]
46. Goodliffe, M. The new UK model for air traffic services—A public private partnership under economic regulation. *J. Air Transp. Manag.* **2002**, *8*, 13–18. [CrossRef]
47. Kistan, T.; Gardi, A.; Sabatini, R.; Ramasamy, S.; Batuwangala, E. An evolutionary outlook of air traffic flow management techniques. *Prog. Aerosp. Sci.* **2017**, *88*, 15–42. [CrossRef]
48. Chen, W.; Cai, S. Ad hoc peer-to-peer network architecture for vehicle safety communications. *IEEE Commun. Mag.* **2005**, *43*, 100–107. [CrossRef]
49. Wang, C.J.; Tan, S.K.; Low, K.H. Collision risk management for non-cooperative UAS traffic in airport-restricted airspace with alert zones based on probabilistic conflict map. *Transp. Res. Part C Emerg. Technol.* **2019**, *109*, 19–39. [CrossRef]
50. Liu, Y.; Liu, J.; Salles, M.A.V.; Zhang, Z.; Li, T.; Hu, B.; Henglein, F.; Lu, R. Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. *Comput. Sci. Rev.* **2022**, *46*, 100513. [CrossRef]
51. de Oliveira, I.R.; Matsumoto, T.; Neto, E. Blockchain-based traffic management for Advanced Air Mobility. *arXiv* **2022**, arXiv:2208.09312.
52. Kushwaha, S.S.; Joshi, S.; Singh, D.; Kaur, M.; Lee, H.-N. Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. *IEEE Access* **2022**, *10*, 6605–6621. [CrossRef]
53. Cao, B.; Zhang, Z.; Feng, D.; Zhang, S.; Zhang, L.; Peng, M.; Li, Y. Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digit. Commun. Netw.* **2020**, *6*, 480–485. [CrossRef]
54. Ghommam, J.; Saad, M.; Mnif, F.; Zhu, Q.M. Guaranteed Performance Design for Formation Tracking and Collision Avoidance of Multiple USVs With Disturbances and Unmodeled Dynamics. *IEEE Syst. J.* **2020**, *15*, 4346–4357. [CrossRef]
55. Hashemi, S.M.; Botez, R.M.; Grigorie, T.L. New Reliability Studies of Data-Driven Aircraft Trajectory Prediction. *Aerospace* **2020**, *7*, 145. [CrossRef]
56. Hashemi, S.M.; Hashemi, S.A.; Botez, R.M.; Ghazi, G. Aircraft Trajectory Prediction Enhanced through Resilient Generative Adversarial Networks Secured by Blockchain: Application to UAS-S4 Ehécatl. *Appl. Sci.* **2023**, *13*, 9503. [CrossRef]
57. Hashemi, S.M.; Botez, R.M. A Novel Flight Dynamics Modeling Using Robust Support Vector Regression against Adversarial Attacks. *SAE Int. J. Aerosp.* **2023**, *16*. [CrossRef]
58. Hashemi, S.; Botez, R. Lyapunov-based Robust Adaptive Configuration of the UAS-S4 Flight Dynamics Fuzzy Controller. *Aeronaut. J.* **2022**, *126*, 1187–1209. [CrossRef]
59. Bashllari, A.; Kaciroti, N.; Nace, D.; Fundo, A. Conflict Probability Estimations Based on Geometrical and Bayesian Approaches. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Bellevue, WA, USA, 30 September–3 October 2007.
60. Aiyar, K.; Halgamuge, M.N.; Mohammad, A. Probability distribution model to analyze the trade-off between scalability and security of sharding-based blockchain networks. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021.
61. Borran, F.; Schiper, A. A leader-free byzantine consensus algorithm. In *International Conference on Distributed Computing and Networking*; Springer: Berlin/Heidelberg, Germany, 2010.