A Novel Fuzzy Trust-based Secure Routing Scheme in Flying Ad Hoc Networks

Mehdi Hosseinzadeh[a,b]

E-mail: mehdihosseinzadeh@duytan.edu.vn

Adil Hussein Mohammed[c]

E-mail: adil.mohammed@cihanuniversity.edu.iq

Farhan A. Alenizi[d]

E-mail: fa.alenizi@psau.edu.sa

Mazhar Hussain Malik[e]

E-mail: Mazhar.malik@uwe.ac.uk

Efat Yousefpoor[f]

E-mail: eyousefpoor@iaud.ac.ir

Mohammad Sadegh Yousefpoor[f]

E-mail: ms.yousefpoor@iaud.ac.ir

Omed Hassan Ahmed[g]

E-mail: omed.hassan@uhd.edu.iq

Amir Masoud Rahmani[h,1]

E-mail: rahmania@yuntech.edu.tw

Lilia Tightiz[i,1]

E-mail: liliatightiz@gachon.ac.kr

[a]*Institute of Research and Development, Duy Tan University, Da Nang, Vietnam*
[b]*School of Medicine and Pharmacy, Duy Tan University, Da Nang, Vietnam*
[c]*Department of Communication and Computer Engineering, Faculty of Engineering, Cihan University, Erbil, Kurdistan Region, Iraq*
[d]*Electrical Engineering Department, College of engineering, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*
[e]*School of Computing and Creative Technologies College of Arts, Technology and Environment (CATE) University of the West of England Frenchay Campus, Coldharbour Lane Bristol, BS16 1QY, United Kingdom*
[f]*Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran*
[g]*Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq*
[h]*Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan*
[i]*School of Computing, Gachon University, 1342 Seongnamdaero, Seongnam 13120, Korea*

Abstract

Today, many studies assess vulnerabilities, threats, and attacks in flying ad hoc networks (FANETs) to provide solutions for countermeasures. Protecting FANETs against attackers and coordinating connections are challenging. The purpose of this study is to increase and maintain communication security. In this paper, a fuzzy trust-based secure routing scheme (FTSR) is presented in FANETs. FTSR utilizes two trust assessment mechanisms, namely local trust and path trust. Local trust strategy is a distributed process for finding reliable neighboring nodes and isolating hostile nodes on the network. In this regard, only reliable nodes are allowed to contribute to the path discovery procedure. This lowers the risk of forming fake paths in FANETs. Path trust strategy is responsible for identifying hostile nodes that are not identified in the local trust process. This strategy shows a general view of the trust status of the desired path. To design this mechanism, the source node runs a fuzzy system to select the safest path between source and the destination. Finally, network simulator 2 (NS2) implements FTSR, and the results such as malicious detection rate, packet delivery ratio, packet loss, accuracy, and delay are obtained from the simulation process. These results indicate that FTSR presents better performance compared to TOPCM, MNRiRIP, and MNDA. However, FTSR takes more time to find paths compared to TOPCM.

Keywords: Flying ad hoc networks (FANETs), Fuzzy logic, Cybersecurity, Routing, Artificial intelligence (AI)

1. Introduction

Unmanned aerial vehicles (UAVs), also called drones, have a structure similar to aircraft, but the big difference between UAVs and classic aircraft is that UAVs can perform their missions without a human pilot [1, 2]. UAVs have attracted a lot of attention due to numerous applications in many parts. Additionally, the drone market has experienced huge growth in the past decades. For example, in the United States, the number of UAVs registered by the federal aviation administration (FAA) is 855,860 by May 2022 [3], so that recreational and commercial targets used 63% and 37% of these UAVs, respectively. According to Drone Market Report 2020 in [6], the experts have estimated that software advances will increase their global market from £15.8 billion to more than £30.2 billion in the range of 2020-2025. Additionally, a PwC report presented based on the British gross domestic product (GDP) forecasts that drones will increase £42 billion in revenue, decrease £16 billion in net costs, and provide suitable conditions for more than 628,000 jobs by 2023 [7]. In addition, based on statistics presented in [8], the experts have estimated that the UAV market with a Compound Annual Growth Rate (CAGR) of $14.1 billion in 2020 experiences exponential growth and reaches $21.8 billion in 2027 [9].

Initially, drones were used to carry out military missions. As a result, they gained many capabilities to use in many applications. Today, their applications are not limited to military areas. For example, they can be used in agriculture, media coverage, emergency services, healthcare, atmospheric guidance, object recognition, tracking, monitoring, and data collection [10]. When occurring the COVID-19 epidemic, UAVs perform various missions such as monitoring the movement of vehicles and individuals, preventing gatherings, delivering drugs, and spraying disinfectants. In addition, drones can extend their coverage area using multi-UAV networks and form a flying ad hoc network (FANET). This network allows different UAVs to communicate with each other. For this reason, providing security for the exchanged data between UAVs and protecting them against cyber-attacks are very important research areas, which should be considered. Constrained resources and wireless communications in FANET have created gaps to penetrate, launch attacks, and damage to the network due to cyber threats [11, 12]. UAVs, which are a fundamental part of FANET, have characteristics, including small size, low storage capacity, constrained energy source, and low processing power. To design any security system, a balance between lightweight and efficiency of the designed security system should always be considered. To achieve a strong security system, it is necessary to find the vulnerable points of FANET and attempt to fix them. Attackers use these vulnerable points to penetrate the network in order to carry out destructive activities such as deleting or changing data in the network. These destructive activities can lead to unpleasant events, including the breakdown of UAVs, unsafe landings, and collisions. In FANET, the most important vulnerable points are wireless connections between UAVs, the possibility of physical access to UAVs in the air, dynamic topology, constrained resources, unencrypted GPS data, and the hardware vulnerability of UAVs. These vulnerable points create security gaps in the network and thus provide suitable conditions for attackers to launch various attacks on the network [32, 33].

Trust management is a strong security solution to prevent various attacks in FANET. Trust management means the evaluation of the reliability of UAVs based on their communication history in the network. Based on the evaluations performed on communications and interactions between UAVs, the trust system determines whether an entity is trustworthy or untrustworthy. Recently, many researchers have proposed trust-based routing methods. Routing is the data transfer process between UAVs in FANET. The most important existing routing protocols are static, proactive, reactive, geographic, and hierarchical routing [34, 35, 36].

Based on the points mentioned above, the main challenge in a trust-based routing method is how to ensure network security in FANETs. In a routing process, trust evaluation helps UAVs to detect and neutralize routing attacks such as black hole (BH), wormhole (WH), selective forwarding (SF), and flooding attacks. In BH, the attacker deceives normal UAVs to transmit their data through these hostile nodes [37, 38]. Then, the BH node drops these packets. In WH, one or more attacker nodes deceive normal UAVs to send their packets through the WH tunnel. Then, the attackers use the tunnel to send data packets to the other side of the network and steal data. In addition, in SF, network availability is damaged. This attack is difficult to detect due to the unpredictable behavior of SF nodes because they behave dually. They choose a random time period to conduct their hostile operations on the network. In this period, SF nodes are similar to BH nodes and drop data packets. After this time period, SF nodes have completely normal and correct behavior and send data packets to desired nodes. In a flooding attack, the attacker sends a large number of packets to drain the resources of UAVs and reduce the network bandwidth. This process leads to the abnormal performance of UAVs so that their memory capacity is full and their energy is lost. Therefore, it is very important to design a lightweight and efficient defense system to counteract and neutralize these attacks. Studies indicate the fact that few research works have tackled cyber-attacks and provided efficient security solutions in FANET [39, 40].

In this paper, a fuzzy trust-based secure routing scheme (FTSR) is presented for FANET to detect BH, WH, SF, and flooding attacks in the network. In recent years, these attacks have been responsible for most security events in FANETs. Due to the increasing popularity of this technology and its use in various areas, the importance and necessity of this research have become more and more evident. While researchers do some efforts to respond to the problems and challenges mentioned above in past years, there are still many gaps in relation to the security of these networks. This paper attempts to fill existing gaps by presenting a novel security method based on fuzzy logic. FTSR considers two trust assessment mechanisms, namely local trust and path trust. A local trust strategy is used to find reliable neighboring UAVs. However, this strategy may not be able to distinguish all hostile nodes because some attack-

[1]Corresponding author

ers, such as Grey hole, are trying to hide and not be diagnosed with local trust. Therefore, the detection of these nodes will be done by the path trust strategy. This mechanism presents a general picture of the trust status of the path formed between source and destination. The path trust mechanism is designed using fuzzy logic and implemented in the source node to identify the safest route between source and destination. In general, the main contributions of our work are as follows:

- FTSR suggests a trust-based routing method in FANET. While many existing trust-based routing methods focus solely on one type of attack, FTSR can simultaneously neutralize four different types of the most dangerous attacks in FANETs.
- FTSR uses a local trust strategy to find normal neighboring nodes and isolate malicious nodes in the network. Local trust refers to the trust of a UAV relative to its neighboring nodes. It is obtained from three security parameters, i.e. BH and SF-based local trust, WH-based local trust, and flooding-based local trust. According to this mechanism, each node detects some malicious neighboring and creates a list of potentially malicious UAVs to prevent their effect on the network performance.
- In FTSR, in order to improve security, only nodes that are identified as trustworthy in the local trust evaluation mechanism participate in the route discovery process. This issue reduces the risk of forming fake routes in the network.
- In FTSR, the format of the route request message (RREQ) has been modified and five parameters, including energy change rate, delay, packet loss rate, packet transmission frequency, and packet reception frequency are added to this message. These parameters are used to evaluate the paths formed between source and destination so that the safest path is chosen for sending data.
- In FTSR, fuzzy logic is used for the first time based on our best knowledge of research background to design a path trust mechanism to deal with cyber-attacks against FANETs. In this regard, FTSR proposes a route selection process based on path trust evaluation. This path trust mechanism is a fuzzy system, which includes three inputs, namely BH and SF-based path trust, WH-based path trust, and flooding-based path trust. The output of this fuzzy system determines the trust status of the routes formed between source and destination. The purpose of this process is to detect black hole, flooding, wormhole, and selective forwarding attacks so that unsafe paths are identified and removed from the network. In addition, FTSR finds a safe route with low delay and a small number of hops to improve network performance.
- FTSR is compared with TOPCM, MNRiRIP, and MNDA in terms of hostile node detection rate, packet delivery rate, packet loss rate, accuracy, and delay. This evaluation shows the optimal performance of FTSR in comparison with other routing methods. Although, FTSR requires more time to find paths compared to TOPCM.

The organization of this paper is as follows: Section 2 includes some research works related to secure routing in FANETs. Section 3 illustrates the principal concepts of fuzzy theory due to its use in FTSR. In Section 4, we describe the network and attack models in FTSR. Our scheme is demonstrated in Section 5. The results related to the simulation process are stated in Section 6. Section 7 summarizes the conclusions of this paper.

2. Related works

Francelin et al. in [13] provided an efficient security solution called tunicate swarm political optimization-based deep residual network (TSPO-DRN) to form secure connections in FANETs. This scheme introduces a new metaheuristic algorithm called TSPO, which mixes political optimizer (PO) and tunicate swarm algorithm (TSA). TSPO constitutes routing paths between flying nodes. It applies a fitness function that includes link quality and distance. TSPO-DRN defines three agents, namely evaluator, decision maker, and defender. The evaluator is responsible for monitoring the exchanged data and checking the behavior of UAVs. The result of this evaluation is recorded in the table stored in the source UAV and the report is sent to the decision maker. This agent uses the DRN technique and examines each path in accordance with five factors, for example round trip time, signal power, delivered packets, packet size, and number of input packets to identify malicious paths. Then, the decision maker determines hostile UAVs and sends a warning to the defender. To identify attackers, the defender utilizes test packets to check the formed paths. If there is an attacker on the path, this node will prevent test packets to reach the destination, and the defender will not receive any acknowledgment from the destination UAV. After sending a certain number of the test packets, the defender examines whether the desired path suffers from a lot of lost packets. If yes, it concludes that there are hostile UAVs in this route and eliminates that path.

Buksh et al. in [14] suggested the trust-oriented peered customized mechanism (TOPCM) in FANETs to calculate the trust of UAVs and to identify and separate hostile UAVs on the network. In TOPCM, the routing idea is inspired by AODV, but one difference is that TOPCM modifies the route request and route response packets used in the routing operations. This scheme uses two trust modules, namely evaluator and decision maker to gain the trust value of UAVs along the discovered path. The trust evaluator considers broadcast ID, destination address, next-hop ID, and current node ID to analyze the trust level of UAVs. Next, it records the trust related to each UAV in a trust table. Evaluation operations are carried out by monitoring RREQ and extracting evaluation parameters from it. In TOPCM, when a UAV broadcast RREQ, it must send an ACK message called R packet for the previous-hop UAV. This R packet contains information required to calculate trust. Then, the decision maker is responsible for comparing the trust of UAVs with the threshold trust and identifying the hostile and

honest UAVs based on the result obtained from the evaluation module.

Fotohi et al. in [15] introduced an agent-based self-protective system (ASP-UAVN) to guarantee communication security in UAV networks. The purpose of this system is to discover secure paths between UAVs according to the routing strategy presented in AODV. This approach focuses on three attacks namely selective forwarding, wormhole, and sink hole to improve the packet delivery ratio and the malicious detection rate. ASP-UAVN presents a multi-agent security system inspired by the human immune system (HIS). The system has three trust agents (i.e. evaluator, decision maker, and defender) and a knowledge base. The evaluator is responsible for analyzing the behavior of UAVs in the discovered paths and recording the gathered information in a trust table. In the evaluation operations, the evaluator calculates a probability value that indicates the presence of hostile UAVs in a path. For this reason, it sends hello packets to the destination through the discovered paths and receives the ACK message. Finally, if this probability is greater than a threshold, the relevant path will be destructive, and must be eliminated. Other agents will evaluate the rest of paths in next steps. The decision maker has a direct relationship with the knowledge base and is responsible for examining suspicious paths. The decision maker obtains a threshold value in accordance with delay, packet delivery, packet loss rate, and packet sending rate to separate malicious paths from suspicious paths. Finally, the defender analyzes the UAVs in the desired path to identify and separate hostile UAVs. In this operation, each UAV uses test packets to find fake paths. Finally, the source UAV evaluates paths with low thresholds based on round trip time and the signal power to choose the safest paths for data transfer.

Du et al. in [16] offered the adaptive trust strategy-based lightweight mutual identity authentication scheme (ATS-LIA) in FANETs. ATS-LIA focuses on the energy constraint of UAVs in the network. If the energy level of UAVs is more than an energy threshold, these nodes are allowed to participate in the networking operations, and their trust will be evaluated. Otherwise, these nodes will leave the network to recharge. In this method, the total trust of each UAV is obtained from local trust, global trust, and energy trust, and UAVs with the highest trust level can communicate with the ground station. Then, ATS-LIA provides a lightweight authentication mechanism that uses an elliptic curve encryption technique to confirm the UAV ID. Finally, the performance of ATS-LIA is evaluated based on an accurate security analysis to measure its resistance to various attacks.

Agron et al. in [17] suggested a secure routing scheme based on the ground station (GS) for FANETs. The role of GS in the routing operation is critical because this operation is centralized. GS obtains information such as geographical location, flight time, and link status from UAVs to control network connectivity. It calculates the cost of network links with regard to the spatial information of the nodes. Then, it searches different routes in the network and sends them to the UAVs. This method introduces an authentication technique that combines symmetric and asymmetric cryptography methods. In this technique, symmetric keys and digital signatures are used to encrypt routing messages and ensure communication security, respectively. The purpose of this mechanism is to guarantee data integrity and authentication on the network. The key distribution center (KDC) is responsible for providing public and private keys for each UAV on the network. In addition, the routing operation uses the TWINE algorithm and the packet leash mechanism to tackle wormhole attacks.

Bhardwaj et al. in [18] presented a hierarchical secure routing approach (SecRIP) for FANET. It utilizes the chaotic algae algorithm (CAA) and the dragonfly algorithm (DA) to form clusters and choose cluster heads (CHs), respectively. Clustering has increased scalability and energy efficiency in SecRIP. When forming a cluster, the distance between UAVs is considered in the fitness function to reduce energy consumed inside the cluster. The DA-based CH selection algorithm regards several factors, including consumed energy, network longevity, delay, received signal power, and UAV movement. In SecRIP, CHs are directly connected to each other, and communication security is guaranteed using a Lattice-based cryptosystem called NTRU. SecRIP rises the packet delivery ratio and reduces delay and overhead.

Muruganandam and Manickam in [19] provided a malicious node detection algorithm (MNDA) for mobile ad hoc networks. MNDA applies a stealthy assault location solution to separate fake and actual routing packets in MANET. To achieve this goal, a dynamic malicious node detection algorithm has been used to identify hidden hostile nodes. MNDA employs the DSR routing protocol to discover network paths. On each route, intermediate UAVs act as routers, which forward packets from the source UAV to the destination UAV. In MNDA, each intermediate UAV collects some information such as the packet propagation delay of other intermediate UAVs, and stores it in a table. MNDA employs this information to locate intermediate UAVs and calculate the expected time for incoming the next packet from the source UAV. In the next step, MNDA detects hostile nodes and picks out a suitable path. This method can well identify hostile nodes and prevent the collision of packets. Note that empirical results show that MNDA is efficient.

Rahmani et al. in [20] proposed a modified OLSR named OLSR+ in FANET. OLSR+ seeks to achieve energy efficiency and path stability on the network. In OLSR+, a novel solution has been presented to compute link lifetime. This solution has used factors like link quality, distance, relative speed, and movement direction. In OLSR+, a fuzzy system is introduced to pick out multipoint relays (MPRs). According to this system, nodes, which include high energy, long link lifetime, and high neighbor degree, obtain a greater MPR chance than other UAVs. Moreover, to calculate the routing table, OLSR+ regards path energy and lifetime to form stable paths.

Lee et al. in [21] presented a fuzzy logic-based routing approach in FANETs. In this approach, the authors have attempted to solve the broadcast storm problem in AODV. To solve this problem, each UAV calculates a score, and UAVs that get the required score can participate in broadcasting routing messages. This balances the energy consumption of network UAVs and consequently improves network lifespan. To calculate this score, the direction of UAVs, energy, link quality, and

Table 1: Comparison of related works.

Approach	Benefits	Weaknesses
TSPO-DRN [13]	Using deep learning technique, speed up convergence rate using TSPO, increasing packet delivery ratio, reducing delay, ability to detect selective forwarding, sink hole, and wormhole	Not presenting simulation setting such as initial population, UAV speed, and mobility model, high computational complexity, lack of energy efficiency in the routing operations, high overhead
TOPCM [14]	The ability to identify and isolate hostile nodes in the network, appropriate PDR, high detection accuracy	High overhead, long delay in the path discovery process
ASP-UAVN [15]	Using HIS algorithm to design a self-organized security mechanism, ability to detect wormhole, sink hole and selective forwarding attacks, ability to identify and isolate hostile nodes, high PDR, high accuracy	High computational complexity, long delay in the safe path selection process, high consumed energy
ATS-LIA [16]	Providing a lightweight authentication mechanism, managing consumed energy in the network, decreasing computational overhead and managing routing overhead, ability to detect different attacks in the network	Failure to do sufficient experimental scenarios, not evaluating the malicious detection rate, not examining the trust status of UAVs, not evaluating the consumed energy in the network
Agron et al. [17]	Ability to detect different attacks on the network, presenting a hybrid authentication mechanism, high PDR	Presenting a centralized routing scheme, high routing overhead, low scalability, lack of sufficient testing scenarios, lack of energy efficiency in the routing process
SecRIP [18]	Introducing a cluster-based routing method, high scalability, energy efficiency, improving PDR, reducing delay in the routing process	High computational complexity due to the simultaneous use of two metaheuristic algorithms, not having attack model and not providing a security analysis, not testing the security status of UAVs in the presence of hostile nodes, not evaluating energy consumption or network lifetime
MNDA [19]	Ability to detect different attacks on the network, providing a dynamic malicious detection algorithm to identify hidden hostile nodes, preventing packet collision	Low scalability, low PDR, low detection accuracy, long delay in the routing process, incompatibility with FANET environment
OLSR+ [20]	Considering the energy problem when creating paths and selecting MPRs, extending network lifetime, trying to create stable paths, introducing a new technique to calculate network lifetime	Failure to test network performance with regard to the speed of UAVs, high routing overhead, not presenting a proper security mechanism to ensure secure data transfer
Lee et al. [21]	Solving the broadcast storm problem, balancing the energy consumption of UAVs and extending network lifetime, preventing path failure, and replacing failed paths quickly	Failure to evaluate the network performance with regard to the speed of UAVs, high routing overhead, the lack of secure connections between UAVs

distance are considered. Additionally, this method uses a fuzzy structure to pick out more suitable paths. It has presented a suitable solution to prevent path failure. In addition, when failing paths, these routes are quickly replaced with novel paths so that the data transfer process will not be disrupted.

Table 1 summarizes the related works and expresses their benefits and weaknesses. Based on the trust-based routing solutions reviewed in this section, it can be seen that most of these methods focus only on one type of attack. While the proposed method, can simultaneously neutralize and deal with four types of the most dangerous attacks, namely BH, WH, SF, and flooding in FANET. To achieve this goal, FTSR considers two trust evaluation mechanisms, i.e. local trust mechanism and path trust mechanism. This local trust strategy is applied to find trusted neighbor nodes. However, the local trust mechanism may not be able to detect all hostile nodes because some attackers like SF try to hide themselves in the network. Therefore, these nodes are detected using the path trust mechanism, which presents a global view of the trust status of the path formed between source and destination. It is designed using a fuzzy system and is embedded in the source node to identify the safest route between source and destination.

3. Basic concept

Fuzzy logic (FL) is an appropriate and accurate strategy based on approximate and ambiguous data [22, 23]. Nowadays, fuzzy-based routing approaches, for example [20], [21], and [24] use fuzzy theory to obtain routing paths and make the best routing decisions. Fuzzy sets are different from classic sets because they allow an element to be partially a member of a set. Usually, a fuzzy system (FS) has four components, namely fuzzifier, fuzzy rules, inference engine, and defuzzifier. Fuzzification categorizes numerical scales into fuzzy sets. To achieve this goal, it defines linguistic variables and a variety of membership functions (MFs) such as triangular, trapezoidal, and Gaussian, and converts crisp data into suitable linguistic values. The task of MFs is to make a map of input variables and their membership degree. Fuzzy rules illustrate linguistic propositions that indicate the input-output relationships. Experts empirically define them. A fuzzy rule is displayed as "**IF** antecedent(s) **THEN** consequent(s)", so that antecedents and consequents represent input space and output space, respectively. They are defined through a combination of fuzzy sets. The inference engine makes an approximate argument for achieving the desired solution. The inference process maps fuzzy inputs to fuzzy rules to produce fuzzy output. The most important inference methods are Min-Max, averaging, and clipped center of gravity. Finally, the choice of defuzzification techniques is very important and effective in the speed and accuracy of a fuzzy system. Defuzzification determines how to extract the crisp value from the fuzzy output. The most famous defuzzification methods are centroid, bisector, Mean of Maximum (MoM), Largest of Maximum (LoM), and Smallest of Maximum (SoM). The centroid method is commonly used for defuzzification operations because it provides more reliable results than others [25].

4. System model

This section explains the system model used in FTSR. It consists of two main parts: network model and attack model.

4.1. Network model

In FTSR, FANET includes N flying nodes (i.e. U_i, $i = 1,2,\ldots,N$) and a ground control station (GCS). The UAVs are

randomly scattered on the network. Moreover, there are a number of hostile UAVs on this network. Figure 1 depicts the network model in FTSR. In this model, each UAV is equipped with a communication module to connect with GCS and other flying nodes on the network. All network nodes utilize the IEEE.802.11a standard in their MAC layer because it provides a proper bandwidth and supports highly dynamic topology [26]. Each UAV has a unique ID and connected to GPS to inform its location on the network. FTSR utilizes the air-to-air (A2A) channel. Since there are a lot of missing packets in less fading, this model can be stated according to the free space model. Thus, Equation 1 calculates the path loss of the A2A channel [27].

$$PLAA\left(d_{ij}\right) = \beta 10\log_{10}^{d_{ij}} + \alpha \tag{1}$$

where β is the path loss exponent. When the propagation model is free space, its value is $\beta = 2$. Also, the path loss of the reference point is α, which is computed using Equation 2:

$$\alpha = 10\log_{10}^{\left(\frac{4\pi w}{c}\right)} \tag{2}$$

Where w and $c = 3 \times 10^8\ m/s$ are the carrier frequency and the light speed, respectively. Additionally, the distance between U_i and U_j is displayed as d_{ij} and calculated by Equation 3.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{3}$$

So that (x_i, y_i, z_i) and (x_j, y_j, z_j) are the locations of U_i and U_j, respectively.

When transferring l bits from U_i to U_j, the consumed energy of U_i (sender) and U_j (recipient) is computed according to Equations 4 and 5, respectively.

$$E_{tx}\left(l, d_{ij}\right) = l \times E_{elec} + l \times \varepsilon_{fs} \times d_{ij}^2 \tag{4}$$

$$E_{rx}\left(l\right) = l \times E_{elec} \tag{5}$$

Where E_{elec} that the energy used by the recipient/sender circuit to receive/send one bit. ε_{fs} refers to the signal amplifier coefficient in free space.

4.2. Attack model

In FANET, UAVs use wireless communication channels to exchange information between themselves. This can provide conditions for hostile attackers to penetrate and consequently cause serious security damage to the network because these attackers are able to launch different attacks to disrupt the data transfer and routing processes. This subject indicates the importance and necessity of effective solutions to differentiate and separate malicious nodes in the network and ensure that reliable nodes execute the routing operations. FTSR attempts to detect and prevent attacks such as black hole (BH), wormhole (WH), selective forwarding (SF), and flooding.

- **Black hole attack:** In FANET, BH nodes declare that they have zero-cost paths and encourage UAVs to transfer their data through this fake path. In BH, the attacker eliminates all packets transferred through this fake path [28, 29]. This attack is trying to disconnect the links between UAVs and trick them to transmit their packets aimlessly in a long time in order to lose their energy and shut down due to the limited power supply. Figure 2 depicts a BH attack.

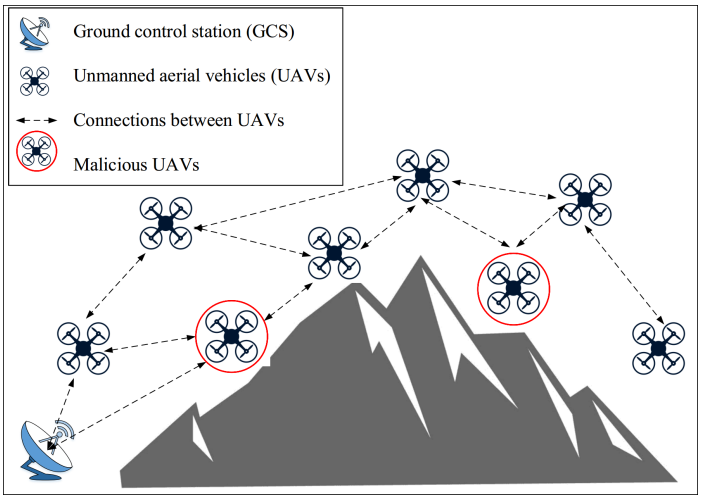

Figure 1: Network model in FTSR.

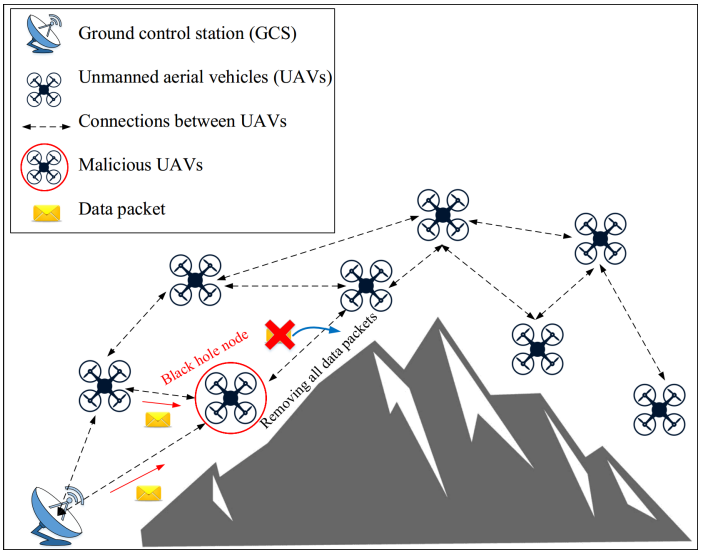

Figure 2: Black hole attack

- **Wormhole attack:** WH is a severe threat in FANET. Usually, two hostile nodes start such an attack and create a WH tunnel between themselves. See Figure 3. In WH tunnels, two hostile nodes are falsely claiming to be so close together. The tunnel is created to transfer the data between the hostile nodes, and the attackers claim that they have the fastest path to the desired area to persuade other UAVs for transferring their data through the WH tunnel [28, 29]. The attack can be carried out by hostile UAVs that have more resources than ordinary UAVs. In WH, the attackers forge a short and efficient path to trick

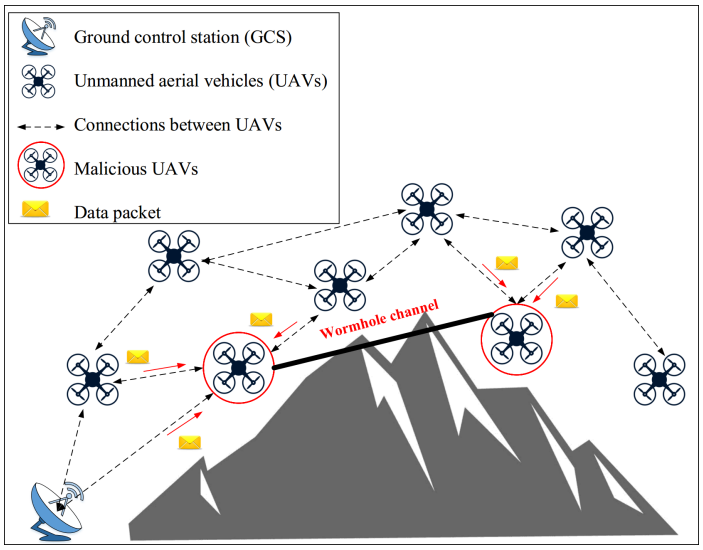

Figure 3: Wormhole attack

UAVs and attract traffic. Thus, the hostile node is able to track the communication of the transmitter UAV, hear and copy its data packets, and manipulate or collect the network traffic. WH could be the basis of other attacks, including Man in the Middle [30]. Therefore, dealing with WH is very important.

- **Selective forwarding:** In SF, also called the Greyhole attack, upon receiving a route request packet, the hostile node sends a fake route reply packet to the source UAV to create an unsafe path. Then, the SF node selectively removes some packets and sends other packets. BH is a simple form of SF and eliminates all passing packets [28, 29]. Figure 4 depicts a SF attack. This attack can be implemented in two ways: removing a specific type of packet or deleting packets sent to a specific destination. This attack is trying to disconnect communication links between UAVs and divert packets to a particular destination.

- **Flooding attack:** In this attack, the hostile node regularly sends fake route requests to the desired UAV to empty its battery and fill its memory because the attacker knows that UAVs save much information about these fake requests [28, 29]. Thus, the desired UAV will inevitably reject actual requests sent by legal UAVs because the capacity of its memory is over. Additionally, the attack increases the energy used by the desired UAV and accelerates its death. This affects network longevity. Due to the limited power supply in the UAVs, it is very important to deal with these attacks. Figure 5 depicts a flooding attack.

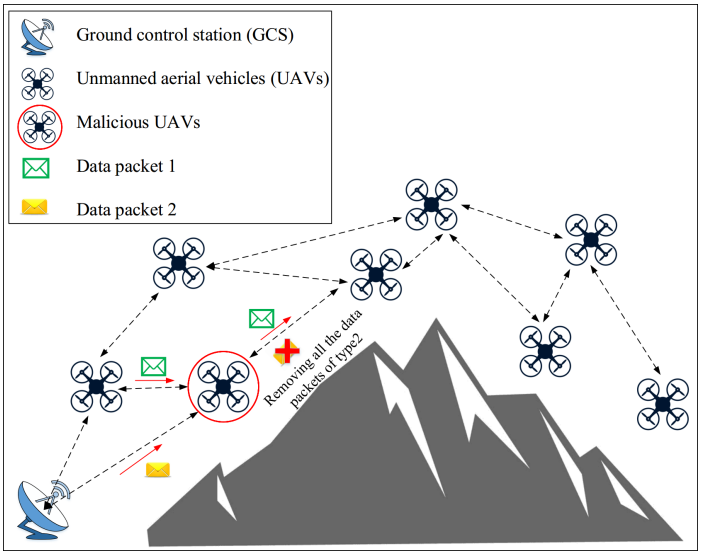

Figure 4: Selective forwarding attack

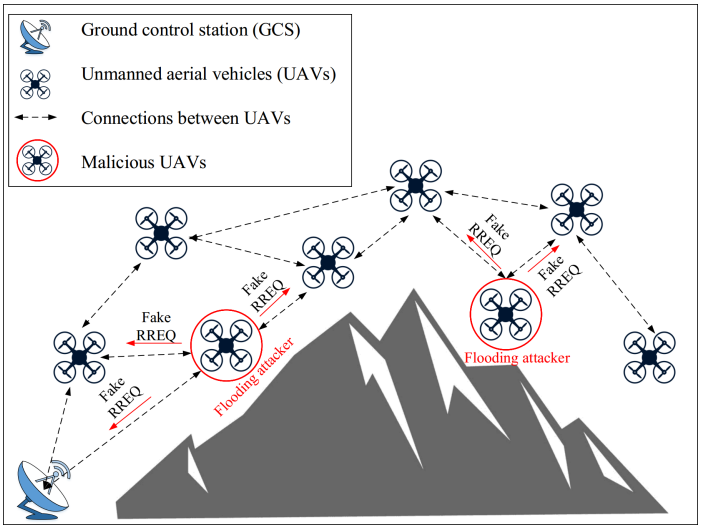

Figure 5: Flooding attack

5. Proposed method

This section describes the fuzzy trust-based secure routing approach (FTSR) for flying ad hoc networks. FTSR contains

Byte 0	Byte 1	Byte 2	Byte 3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Hello message time	Hello message ID	ID_i	E_i^t
(x_i^t, y_i^t, z_i^t)			T_{queue}

Figure 6: The format of hello message.

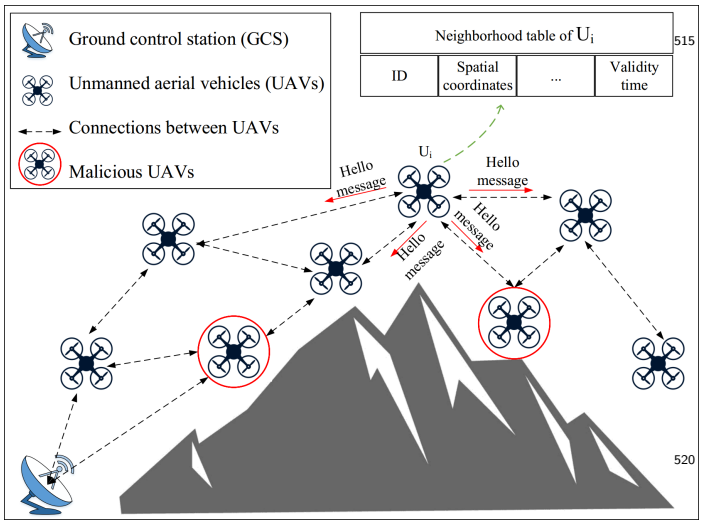

Figure 7: The broadcast of hello messages and the creation of the neighbor table.

the following phases:

- Neighbor discovery
- Local trust assessment
- Path discovery
- Path trust-based route selection
- Path maintenance

5.1. Neighbor discovery phase

Initially, each UAV, for example, U_i ($i = 1,2,...,N$ so that N is the number of UAVs in the network) regularly distributes a hello message to its neighbors. It involves the status information of U_i, for example, identifier (ID_i), queuing delay (T_{queue}), remaining energy (E_i^t), and location (x_i^t, y_i^t, z_i^t). The format of this message is shown in Figure 6. Upon getting hello messages from other UAVs, U_i constructs a neighbor table to store their status information. See this process in Figure 7. To simplify this figure, only the broadcast of hello messages by U_i is shown in Figure 7. Moreover, the format of the neighbor table is presented in Table 2.

Table 2 holds the status information about the location and energy of neighboring UAVs such as U_j. Additionally, it has three important fields, namely link delay (T_{ij}^t), local trust (LT_{ij}^t), and potentially malicious UAVs (PMUs).

- **Link delay ($\mathbf{T_{ij}^t}$):** This field holds the time needed to transfer a hello message from U_i to U_j at the current moment (i.e. t). T_{ij}^t considers various delay scales like propagation (T_{prop}^t), queuing (T_{queue}^t), media access (T_{mac}^t), and transmission (T_{trans}^t). According to the scales mentioned above, Equation 6 calculates T_{ij}^t.

$$T_{ij}^t = T_{prop}^t + T_{queue}^t + T_{mac}^t + T_{trans}^t \tag{6}$$

so that T_{prop}^t is a delay scale for indicating the time when data is transmitted through the wireless media. It is directly proportional to the transfer distance and obtained from Equation 7.

$$T_{prop}^t = \frac{Dist^t(U_i, U_j)}{c} \tag{7}$$

so that $c = 3 \times 10^8$ m/s and $Dist^t(U_i, U_j)$ refer to the light speed and the distance from U_i to U_j at the current time t, respectively.

$$Dist^t(U_i, U_j) = \sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2 + \left(z_i^t - z_j^t\right)^2} \tag{8}$$

where $\left(x_j^t, y_j^t, z_j^t\right)$ and $\left(x_j^t, y_j^t, z_j^t\right)$ are gotten from hello messages and indicate the locations of U_i and U_j at the moment t, respectively. Also, T_{queue}^t comes from hello messages and expresses the time when the packet arrives at the start of the buffer queue. T_{mac}^t is the time of media access and estimated using ACK packets.

$$T_{mac}^t = t_{ACK} - t_S \tag{9}$$

where t_{ACK} and t_S are two moments for indicating the ACK reception time and the packet sending time, respectively. In addition, T_{trans}^t indicates the data transfer time, which is obtained from Equation 10.

$$T_{trans}^t = \frac{msg_{size}}{br} \tag{10}$$

where msg_{size} and br illustrate the length of messages and the data transfer rate, respectively.

- **Local trust ($\mathbf{LT_{ij}^t}$):** This field holds the trust value of U_i relative to a neighboring node, like U_j. LT_{ij}^t is gained from a local and decentralized manner. After calculating this parameter, U_i can detect and remark some hostile neighboring nodes so that these hostile nodes cannot affect network performance. We explain how to get this parameter in Section 5.2.

- **Potentially malicious UAVs (PMU):** It contains a list of binary values so that PMU_j corresponds to U_j. It means whether U_j has been remarked as a hostile UAV in the local trust evaluation process. After calculating local trust (LT_{ij}^t) in Section 5.2, this value is standardized by Equation 11.

$$LT_{ij}^{standard} = \frac{LT_{ij}^t - \mu_{ij}}{\sigma_{ij}} \tag{11}$$

Table 2: The format of neighbor table.

Identifier	Location	Remaining energy	Link delay	Local trust	Potentially Malicious UAVs (PMUs)	Valid interval
ID_j	$\left(x_j^t, y_j^t, z_j^t\right)$	E_j^t	T_{ij}^t	LT_{ij}^t	0/1	VD_t

where μ_{ij} and σ_{ij} are the mean and standard deviation of LT_{ij}^t, they will be calculated according to Equations 12 and 13, respectively.

$$\mu_{ij} = \frac{1}{N_i} \sum_{j \in Nei}^{N_i} LT_{ij}^t \tag{12}$$

$$\sigma_{ij} = \sqrt{\frac{1}{N_i} \sum_{j \in Nei}^{N_i} \left(LT_{ij}^t - \mu_{ij}\right)^2} \tag{13}$$

In Equations 12 and 13, N_i and Nei represent the number of neighbors and the neighbor set of U_i, respectively. Now, if LT_{ij}^t is less than μ_{ij} (i.e. $LT_{ij}^{standard} < 0$), U_i remarks U_j as a hostile node and puts $PMU_j = 1$. Otherwise, if $LT_{ij}^{standard} \geq 0$, U_i remarks U_j as an honest node and puts $PMU_j = 0$. When $PMU_j = 1$, it means that U_j is a hostile node, and U_i does not transfer any route request message (RREQ) to this node and ignores all RREQs or RREPs received from this node. Thus, U_i isolates U_j. Now, if $PMU_j = 0$, this means that U_i has identified U_j as an honest node in the local trust assessment phase (Section 5.2). In this case, this node can participate in the routing process.

Algorithm 1 expresses the pseudo code of the neighbor discovery phase. The time complexity of this algorithm is calculated below.

Line 1 of Algorithm 1 includes a *while* loop, which is constantly repeated during the simulation period. The number of its iterations is equal to the simulation time. Inside the *while* loop, an *IF* command is intended to determine the broadcast time of the hello message (Lines 2-22). This *IF* command includes the following commands.

- A *For* loop (in lines 3-6) is repeated N times. The purpose of this loop is to spread hello messages by UAVs. This loop consists of two commands (Lines 4 and 5) that have fixed run times, c_1 and c_2.

$$T_{For}(n) = N(c_1 + c_2) \tag{14}$$

Assume that there is a fixed number c so that $c \geq c_1 + c_2$.

$$T_{For}(n) = N(c_1 + c_2) \leq cN \tag{15}$$

- An *IF* condition (in lines 7-21) is used to enter the information of neighboring UAVs in the neighbor table and includes the following commands:
  - Four commands (in lines 8-11) with fixed run times r_1, r_2, r_3, and r_4.
  - A command (in line 12) whose run time depends on Equation 6. This equation runs at a fixed execution time r_5.
  - A command (in line 13) that depends on Algorithm 2 whose time complexity is $O(N_i)$.
  - The command (in line 14) whose implementation time is determined by Equation 11. This equation depends on the number of neighbors of each UAV (i.e. N_i).
  - An *IF-ELSE* command (lines 15-19) that has a fixed run time r_6.

Therefore, the overall run time of this *IF* condition is calculated as follows:

$$T_{IF}(n) = r_1 + r_2 + r_3 + r_4 + r_5 + N_i + N_i + r_6 \tag{16}$$

Suppose that there is a fixed number like r:

$$T_{IF}(n) = r_1 + r_2 + r_3 + r_4 + r_5 + N_i + N_i + r_6 \leq rN_i \tag{17}$$

As a result, the overall time complexity of Algorithm 1 is equal to:

$$T(n) = t_s\left(T_{For}(n) + T_{IF}(n)\right) = t_s(cN + rN_i) \tag{18}$$

Note that $N_i \ll N$,

$$T(n) = t_s(cN + rN_i) \leq t_s N(c + r) \tag{19}$$

Therefore, the time complexity of Algorithm 1 is $O(t_s N)$.

5.2. Local trust assessment phase

This section explains the local trust assessment phase, which is a distributed process. Local trust (LT_{ij}^t) represents the trust value of U_i relative to a neighboring node like U_j. After calculating LT_{ij}^t, U_i can detect some hostile neighboring UAVs and does not allow them to affect network performance. As explained in the attack model (Section 4.2), FTSR focuses on BH, WH, SF, and flooding attacks. Thus, UAVs are evaluated in accordance with three security scales, namely BH and SF-based local trust (LT_{ij}^{BH-SF}), WH-based local trust (LT_{ij}^{WH}), and flooding-based local trust (LT_{ij}^{F}).

- **BH and SF-based local trust ($\mathbf{LT_{ij}^{BH-SF}}$):** When black hole (BH) or selective forwarding (SF) attacks occur on the network, the most common solution to detect these attacks is to analyze the packet loss rate (PLR) and the packet reception frequency (PRF) in UAVs. PLR determines how many hello packets are lost. Additionally, PRF indicates the frequency of hello packets received by U_j at interval $[t, t + \Delta t]$. A BH or SF node experiences a high PLR and a low PRF. Therefore, the evaluation of

Algorithm 1 Neighbor discovery process

Input: U_i: UAVs in the networks so that $i = 1,2,...,N$
$Time_{Hello}$: Hello broadcast time
$HT = 0$: Timer for hello message
$t = 0$: Timer for simulation process
t_s: Simulation time
Output: Neighbor table
Begin
while $t \leq t_s$ **do**
  if $HT = Time_{Hello}$ **then**
    for $i = 1$ to N **do**
      $\mathbf{U_i}$**:** Transfer a hello message to its neighboring nodes;
      $HT = 0$;
    end for
    if U_i receives a hello message from a new neighbor such as U_j **then**
      $\mathbf{U_i}$**:** Add a new entry to its neighbor table;
      $\mathbf{U_i}$**:** Extract ID_j from the hello message and insert it into the node ID field of the neighbor table;
      $\mathbf{U_i}$**:** Extract $\left(x_j^t, y_j^t, z_j^t\right)$ from the hello message and record it in the location field in the neighbor table;
      $\mathbf{U_i}$**:** Obtain E_j^t from the hello message and record it in the residual energy field of the neighbor table;
      $\mathbf{U_i}$**:** Calculate T_{ij}^t based on Equation 6 and insert it into the link delay field in the neighbor table;
      $\mathbf{U_i}$**:** Determine LT_{ij}^t according to Algorithm 2 and record it in the local trust field of the neighbor table;
      $\mathbf{U_i}$**:** Standardize LT_{ij}^t based on Equation 11;
      if $LT_{ij}^{standard} < 0$ **then**
        $PMU_j = 1$;
      else
        $PMU_j = 0$;
      end if
      $\mathbf{U_i}$**:** Insert PMU_j into the potentially malicious UAV field in the neighbor table;
    end if
  end if
  $HT = HT + 1$;
end while
End

these two scales can help U_i to be informed of the presence of a BH or SF node around itself. This evaluation is performed by Equation 20.

$$\begin{aligned} LT_{ij}^{BH-SF} &= \Psi\left(1 - PLR_j\right) + \left(1 - \Psi\right) PRF_j \\ &= \Psi\left(1 - \frac{msg_j^{dropped}}{msg_{total}}\right) + \left(1 - \Psi\right)\left(\frac{msg_j^{received}}{\Delta t}\right) \end{aligned} \tag{20}$$

where $msg_j^{dropped}$ and msg_{total} are the number of missing hello packets and all packets, respectively. Moreover, $msg_j^{received}$ is the number of packets gotten at $[t, t+\Delta t]$. Ψ is a weight coefficient and shows the effect of PLR_j and PRF_j on the value of LT_{ij}^{BH-SF}. Ψ is an adjustable weight in $[0,1]$. If the value of Ψ is less than one and approaches zero, PRF_j will have a greater effect on LT_{ij}^{BH-SF} than PLR_j, so if $\Psi = 0$, then LT_{ij}^{BH-SF} is only evaluated based on PRF_j, and PLR_j has no effect on it. In contrast, if the value of Ψ is close to one, PLR_j has more effect on LT_{ij}^{BH-SF} than PRF_j, so if $\Psi = 1$, LT_{ij}^{BH-SF} depends only on PLR_j. In this paper, Ψ is set to 0.5 so that PLR_j and PRF_j have the same effect on LT_{ij}^{BH-SF}.

- **WH-based local trust ($\mathbf{LT_{ij}^{WH}}$):** When the WH attack occurs in the network, the tunnel formed between two hostile nodes does not allow the UAVs, which are close to the attackers to find valid paths because the WH node claims that it has a short path to the destination. These attacks increase network congestion, queuing delay, and lost packets. Thus, the scales mentioned above increase the need to re-transfer packets on the network. An appropriate solution to detect such an attack is to evaluate the link delay, the packet transfer frequency (PTF), PRF, and PLR. PTF indicates the frequency of sent packets at time $[t, t+\Delta t]$. This evaluation is done through Equation 21.

$$\begin{aligned} LT_{ij}^{WH} &= \omega_1\left(1 - \frac{T_{ij}^t}{\max\limits_{j \in Nei} T_{ij}^t}\right) + \omega_2\left(1 - PLR_j\right) + \\ &\omega_3\left(1 - PTF_j\right) + \omega_4 PRF_j \\ &= \omega_1\left(1 - \frac{T_{ij}^t}{\max\limits_{j \in Nei} T_{ij}^t}\right) + \omega_2\left(1 - \frac{msg_j^{dropped}}{msg_{total}}\right) + \\ &\omega_3\left(1 - \frac{msg_j^{transferred}}{\Delta t}\right) + \omega_4\left(\frac{msg_j^{received}}{\Delta t}\right) \end{aligned} \tag{21}$$

where Nei is the neighbor set of U_i, T_{ij}^t is the delay between U_i and U_j. $msg_j^{dropped}$ is the number of missing packets in U_j, and msg_{total} is the total number of packets. Moreover, $msg_j^{transferred}$ and $msg_j^{received}$ indicate the total number of sent and received packets at $[t, t+\Delta t]$, respectively. ω_1, ω_2, ω_3, and ω_4 are weight coefficients, and their value is limited to $[0,1]$ and $\Sigma_{i=1}^4 \omega_i = 1$. Each coefficient shows the effect of the corresponding parameter on the value of LT_{ij}^{WH}. If ω_1 has a larger value (closer to one) than other coefficients, then one-hop delay (T_{ij}^t) is the most influential parameter on LT_{ij}^{WH}. In addition, if ω_2 has a greater value than other weights, PLR_j is known as the most effective parameter in determining LT_{ij}^{WH}. Likewise, if $\omega_3 = 1$, LT_{ij}^{WH} depends only on PTF_j. In addition, if $\omega_4 = 1$, LT_{ij}^{WH} will be determined based on PRF_j. In this paper, it is assumed that these parameters, including PLR_j, T_{ij}^t, PTF_j, and PRF_j have the same effect on LT_{ij}^{WH}, and as a result, four weight coefficients have the same value ($\omega_1 = \omega_2 = \omega_3 = \omega_4 = \frac{1}{4}$).

- **Flooding-based local trust ($\mathbf{LT_{ij}^{F}}$):** When a flooding attack occurs in the network, the hostile node experiences a high PTF. Thus, the memory of the target nodes will be full, and consequently, the consumed energy and delay will be increased in the network nodes. Therefore, the solution to detect the attack is to evaluate the energy change rate (ECR), delay (T_{ij}^t), and PTF. This evaluation is performed based on Equation 22.

$$\begin{aligned} LT_{ij}^{F} &= \lambda_1\left(1 - ECR_j\right) + \lambda_2\left(1 - \frac{T_{ij}^t}{\max\limits_{j \in Nei} T_{ij}^t}\right) + \lambda_3\left(1 - PTF_j\right) \\ &= \lambda_1\left(1 - \frac{E_j^{t-1} - E_j^t}{t - (t-1)}\right) + \lambda_2\left(1 - \frac{T_{ij}^t}{\max\limits_{j \in Nei} T_{ij}^t}\right) + \\ &\lambda_3\left(1 - \frac{msg_j^{transferred}}{\Delta t}\right) \end{aligned} \tag{22}$$

where E_j^{t-1} and E_j^t are the remaining energy of U_j in two moments $t-1$ and t, respectively. In addition, Nei indicates the set of neighbors of U_i, T_{ij}^t is the delay between U_i and U_j. It is gotten from the neighbor table.

$msg_j^{transferred}$ indicates the total number of sent packets at $[t, t+\Delta t]$. λ_1, λ_2, and λ_3 are weight coefficients and show the effect of three parameters, ECR_j, T_{ij}^t, and PTF_j, on the value of LT_{ij}^F, respectively. These weight coefficients are limited to $[0,1]$ and $\Sigma_{i=1}^{3}\lambda_i = 1$. If $\lambda_1 = 1$, LT_{ij}^F will be determined only based on ECR_j, and other parameters do not affect its value. Now, if λ_2 has a larger value than other weight coefficients, then T_{ij}^t is known as the most effective parameter in determining LT_{ij}^F. Similarly, if $\lambda_3 = 1$, LT_{ij}^F depends only on PTF_j. In this paper, it is assumed that ECR_j, T_{ij}^t, and PTF_j have the same effect on LT_{ij}^F, and as a result, three weight coefficients have the same value (i.e. $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$).

Finally, Equation 23 calculates the direct trust of U_i relative to U_j (i.e. LT_{ij}^{direct}) according to three security scales, LT_{ij}^{BH-SF}, LT_{ij}^{WH}, and LT_{ij}^F.

$$LT_{ij}^{direct} = \min\left(LT_{ij}^{BH-SF}, LT_{ij}^{WH}, LT_{ij}^F\right) \tag{23}$$

FTSR utilizes the window mean with exponentially weighted moving average (WMEWMA) to renew LT_{ij}^{direct}. WMEWMA intends a window whose length is w. It records the last w values of LT_{ij}^{direct}. As a result, U_i does not rely only on the present value of LT_{ij}^{direct} and uses its historical values to achieve a better view of the trust level. LT_{ij}^{direct} is updated according to Equation 24.

$$LT_{ij}^{direct}(q) = (1-\Phi)\frac{\sum_{k=q-w}^{q-1} LT_{ij}^{direct}(k)}{w} + \Phi LT_{ij}^{direct}(t) \tag{24}$$

where Φ shows the effect of the current value and the historical values of LT_{ij}^{direct} in updating the value of LT_{ij}^{direct}. In general, Φ is adjustable in $[0,1]$. If Φ has a large value and close to one, then LT_{ij}^{direct} is updated only based on its current value, and if Φ is close to zero, the historical values of LT_{ij}^{direct} have a greater effect on its updated value. This weight coefficient is determined based on the stability value of LT_{ij}^{direct}. If it is unstable, Φ approaches one so that the updated value of LT_{ij}^{direct} is determined based on its present value. While if LT_{ij}^{direct} is stable, Φ will be close to zero so that the historical values play a stronger role in the updating process.

Next, the indirect trust of U_i relative to U_j ($LT_{ij}^{indirect}$) is gained. It represents a trust analysis by the recommenders, which are common and trusted neighboring UAVs whose trust level is higher than $LT_{threshold}$. Suppose $R = \{n_{Recommender}^1, n_{Recommender}^2, ..., n_{Recommender}^p\}$ contains p recommenders between U_i and U_j. In this case, $LT_{ij}^{indirect}$ is obtained using Equation 25.

$$LT_{ij}^{indirect} = \frac{1}{p}\sum_{x\in R}^{p}\left(LT_{ix}^{direct} \cdot LT_{xj}^{direct}\right) \tag{25}$$

where LT_{ix}^{direct} and LT_{xj}^{direct} are two direct trust values of U_i to U_x and U_x to U_j, respectively. Moreover, U_x is a recommender between U_i and U_j. Finally, local trust is achieved based on the sum of direct and indirect trusts through Equation 26.

$$LT_{ij}^{total} = \gamma LT_{ij}^{direct} + (1-\gamma)LT_{ij}^{indirect} \tag{26}$$

so that γ shows the effect of LT_{ij}^{direct} and $LT_{ij}^{indirect}$ in determining LT_{ij}^{total}. In general, $\gamma \in [0,1]$ is an adjustable coefficient. If γ is close to zero, $LT_{ij}^{indirect}$ has more effect on LT_{ij}^{total} than LT_{ij}^{direct}, so if $\gamma = 0$, then LT_{ij}^{total} will only be evaluated based on $LT_{ij}^{indirect}$, and LT_{ij}^{direct} has no effect on it. In contrast, if γ approaches one, LT_{ij}^{direct} has a greater effect on LT_{ij}^{total} than $LT_{ij}^{indirect}$, so if $\gamma = 1$, LT_{ij}^{total} depends only on LT_{ij}^{direct}. In this paper, γ is set to 0.5 so that LT_{ij}^{direct} and $LT_{ij}^{indirect}$ have the same effect on LT_{ij}^{total}.

Algorithm 2 shows the pseudo-code related to this process. The time complexity of this algorithm is calculated below.

This algorithm consists of seven commands (lines 1-7) whose run times are determined by Equations 20 to 26.

- The run time of Equations 20 to 22 depends on the number of neighboring nodes (i.e. N_i).
- Equation 23 has a fixed execution time, i.e. $O(1)$.
- Equation 24 depends on the window length w.
- Equation 25 depends on the number of recommenders between U_i and U_j (i.e.p). Note that $p \ll N_i$, consequently the time complexity of Equation 25 is $O(N_i)$.
- Equation 26 has a fixed run time, i.e. $O(1)$.

According to the above items, time complexity of Algorithm 2 is $O(N_i)$.

Algorithm 2 Local trust evaluation

Input: U_i: UAVs in the networks so that $i = 1, 2, ..., N$
Output: LT_{ij}^t: Local trust between U_i and U_j
Begin
1: $\mathbf{U_i}$: Calculate LT_{ij}^{BH-SF} based on Equation 20;
2: $\mathbf{U_i}$: Compute LT_{ij}^{WH} according to Equation 21;
3: $\mathbf{U_i}$: Obtain LT_{ij}^F from Equation 22;
4: $\mathbf{U_i}$: Calculate LT_{ij}^{direct} according to Equation 23;
5: $\mathbf{U_i}$: Update LT_{ij}^{direct} using the WMEWMA scheme based on Equation 24;
6: $\mathbf{U_i}$: Compute $LT_{ij}^{indirect}$ according to Equation 25;
7: $\mathbf{U_i}$: Calculate LT_{ij}^{total} by using Equation 26;
End

5.3. Path discovery phase

Suppose the source node (U_S) searches for a suitable path to the destination (U_D) to transfer its data, but it does not find such a path. In this case, U_S starts a path discovery procedure and makes a route request (RREQ) message. Then, it distributes the RREQ to its neighboring UAVs, except nodes that have $PMU = 1$. As a result, attackers are separated on the network to reduce the risk of fake paths in the network. This procedure is depicted in Figure 8.

The structure of RREQ is illustrated in Figure 9. In the following, the fields of this message are described.

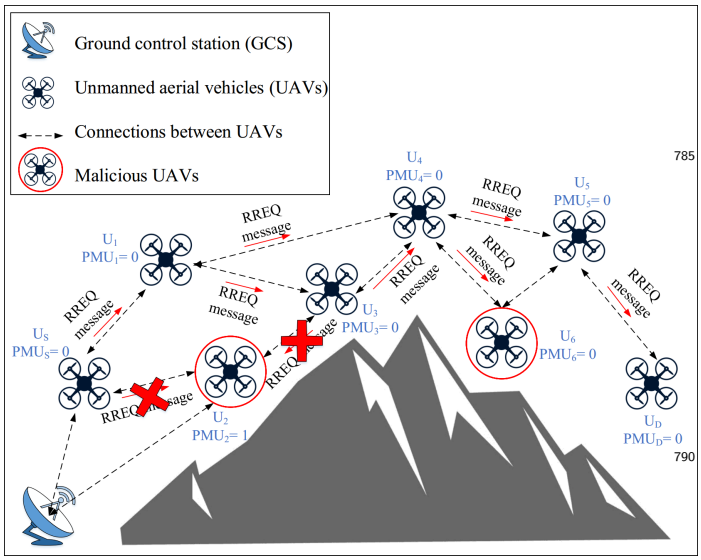

Figure 8: Path discovery procedure by sending RREQs.

- **Message type:** If this field is set on one, this control message is a RREQ.
- **H_c:** It holds the hop count related to a path from U_S to the current UAV. U_S sets this field on one. Then, one unit is added to this field at each hop. The aim of this scale is that the attacker may deceive U_S by announcing the path with a low hop count than legal paths. U_S can identify the fake paths by evaluating this scale.
- **Message ID:** It is a special ID associated with the address of U_S to control the message and prevent repeated messages.
- ΔE_R: This field is the variance of the energy change level of UAVs in the desired path between U_S and U_D. If a route has a high ΔE_R, the UAVs available in this path experience unbalanced energy consumption. Thus, there is likely a flooding attack on this path. As a result, the source node prefers to select paths whose ΔE_R is close to zero. This scale is calculated below.

$$\Delta E_R = \underset{U_i \in Route_k}{\sigma^2}(ECR_i) = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1}\left(ECR_i - \mu^{ECR}_{Route_k}\right)^2 \tag{27}$$

where U_i indicates the UAVs available in the present path ($Route_k$). In addition, ECR_i is the energy change rate of U_i in $[t-1,t]$ so that $ECR_i = \left(\frac{E_i^{t-1}-E_i^t}{t-(t-1)}\right)$. E_i^{t-1} and E_i^t indicate the remaining energy in two moments $t-1$ and t, respectively. Additionally, $\mu^{ECR}_{Route_k}$ is the average energy change of UAVs available in $Route_k$ so that $\mu^{ECR}_{Route_k} = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1} ECR_i$. H_c also indicates hop count along $Route_k$.

- **T_R:** This field holds the end-to-end delay in $Route_k$. It is equal to the sum of the link delays between the intermediate UAVs available in $Route_k$. When T_R is low in a route, it is more appropriate for sending data because hostile nodes cannot be in this route. This scale is obtained from Equation 28.

$$T_R = \sum_{U_i,U_j \in Route_k}^{Destination} T_{ij}^t \tag{28}$$

where U_i and U_j represent two consecutive nodes on $Route_k$. $Destination$ is U_D, and T_{ij}^t indicates the link delay between U_i and U_j in $Route_k$. It is calculated using the information stored in the neighbor table.

- **PLR_R:** This field holds the packet loss rate in a path. It is equal to the maximum PLR in the nodes available on $Route_k$. Unsecure paths have a lot of PLR. As a result, this scale can help diagnose these paths on the network. This value is calculated using Equation 29.

$$PLR_R = \max_{U_i \in Route_k}(PLR_i) = \max_{U_i \in Route_k}\left(\frac{msg_i^{dropped}}{msg_{total}}\right) \tag{29}$$

where $msg_i^{dropped}$ and msg_{total} are the number of packets deleted in U_i and the total number of packets, respectively.

- **PTF_R:** This field holds the packet transfer frequency in a path. It is proportional to the mean of the PTF of UAVs available in $Route_k$ and is obtained from Equation 30. Safe routes have lower PTF. In contrast, if a route contains hostile nodes, the possibility of PLR will increase, and the need to re-transfer the packets will increase.

$$PTF_R = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1} PTF_i = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1}\left(\frac{msg_i^{transferred}}{\Delta t}\right) \tag{30}$$

where $msg_j^{transferred}$ represents the number of sent packets in the interval $[t,t+\Delta t]$. H_c also indicates the hop count in $Route_k$.

- **PRF_R:** This field holds the packet reception frequency of a route. It is equal to the mean of PRF of intermediate UAVs in $Route_k$ and is calculated through Equation 31. Secure paths have more PRF. In contrast, the presence of hostile nodes in a path increases the likelihood of losing the packets and lowers PRF_R.

$$PRF_R = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1} PRF_i = \frac{1}{H_c+1}\sum_{U_i \in Route_k}^{H_c+1}\left(\frac{msg_i^{received}}{\Delta t}\right) \tag{31}$$

so that $msg_j^{received}$ indicates the total number of packets received in the interval $[t,t+\Delta t]$. H_c also indicates the hop count in $Route_k$.

Message Type	H_c	ΔE_R	T_R
PLR_R	PRF_R	PTF_R	
Message ID			
Destination IP Address			
Destination Sequence Number			
Source IP Address			
Source Sequence Number			

Figure 9: The structure of RREQ.

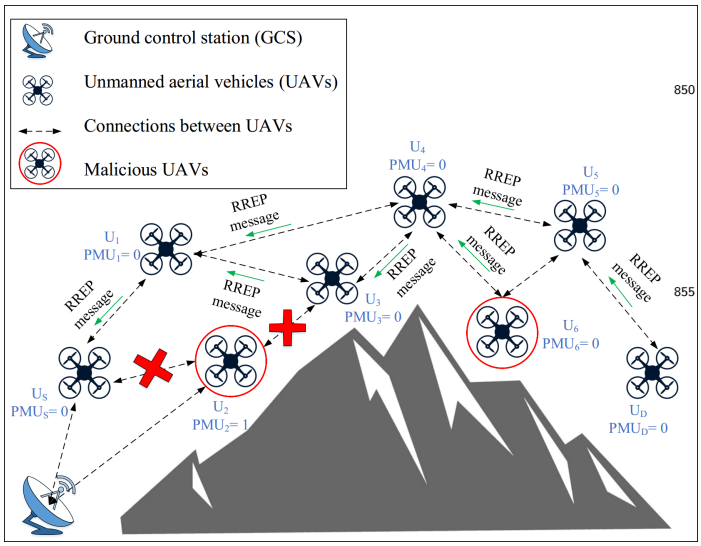

Figure 10: Sending the RREP message to U_S.

- **Source IP address:** It holds the address of U_S that forwards RREQ.
- **Destination IP address:** It holds the address of U_D.
- **Source sequence number:** It ensures that the information of the reverse path to U_S is fresh.
- **The destination sequence number:** Before U_S chooses a path, the field ensures that the path is fresh.

When a neighboring node receives this RREQ message, it first ensures that it is not repetitive. Then, it broadcasts the RREQ again for its neighbors on the network. This procedure is repeated until the RREQ message arrives U_D or the node with a valid path to U_D. Note that the local trust presented in Section 5.2 can detect some hostile nodes. For example, in Figure 8, U_2 was identified as a hostile node, and no RREQ message is sent to it. While some attackers, such as Grey hole, may hide and not be diagnosed with local trust. For example, the hostile node U_6 has not been identified as an attacker. It has participated in the routing process. The detection of these nodes is done at the path trust phase described in Section 5.4. After the RREQ messages arrive at the destination or node with a valid path, this node prepares a route reply (RREP) message and sends it to U_S to identify different routes to U_D. Figure 10 depicts this procedure.

The pseudo-code related of the procedure is presented in Algorithm 3. In the following, the time complexity of this algorithm is analyzed. This algorithm uses an *IF* condition (in lines 1-14), which includes the following commands:

- Two commands (lines 2 and 3) that have fixed run times, b_1 and b_2, respectively.
- A *while* loop (in lines 4-10), which is repeated N time in the worst case. So that N is the total number of UAVs in the network. This loop contains an *IF-ELSE* condition (in lines 5-9) with a fixed run time b_3.
- Two commands (in lines 11 and 12), which have fixed run times b_4 and b_5, respectively.
- A command (line 13) whose run time is dependent on Algorithm 4, whose time complexity is $O(N)$.

Therefore, the time complexity of this algorithm is calculated as follows:

$$T(n) = b_1 + b_2 + b_3 N + b_4 + b_5 + N \tag{32}$$

Consider a fixed number $b \geq b_1 + b_2 + b_3 + b_4 + b_5$:

$$T(n) = b_1 + b_2 + b_3 N + b_4 + b_5 + N \leq bN \tag{33}$$

As a result, the time complexity of Algorithm 3 is $O(N)$.

Algorithm 3 Route discovery process

Input: U_i: UAVs in the networks so that $i = 1, 2, ..., N$
U_S: Source UAV
U_D: Destination UAV
Output: Forming a path between U_S and U_D.
Begin
1: **if** U_S wants to send its data packet to U_D **and** U_S does not have any path to U_D in its routing table **then**
2: **U_S:** Produce a route request message (RREQ);
3: **U_S:** Send RREQ to its neighboring UAV, for example U_j;
4: **while** RREQ is received by U_D **or** RREQ is received by a UAV having a path to U_D **do**
5: **if** $PMU_j = 1$ **then**
6: **U_j:** Delete RREQ;
7: **else if** $PMU_j = 0$ **then**
8: **U_j:** Send RREQ to its neighbors;
9: **end if**
10: **end while**
11: **U_D:** Generate a route reply (RREP) message;
12: **U_D:** Send RREP to U_S;
13: **U_S:** Select the safest route from the paths between U_S and U_D based on Algorithm 4;
14: **end if**
End

5.4. Fuzzy trust-based path selection phase

In this section, U_S evaluates the various paths found in the path discovery phase with regard to a fuzzy-based path trust process to decide on the safest route. In Figure 11, we assume that the three paths have been found between U_S and U_D: $Route(1): U_S - U_1 - U_4 - U_5 - U_D$, $Route(2): U_S - U_1 - U_3 - U_4 - U_5 - U_D$, and $Route(3): U_S - U_1 - U_3 - U_6$. Among these paths, $Route3$ is a fake route that includes U_6. Each path is known by six scales, H_c, ΔE_R, PLR_R, PTF_R, PRF_R, and T_R. They can be extracted from RREQ.

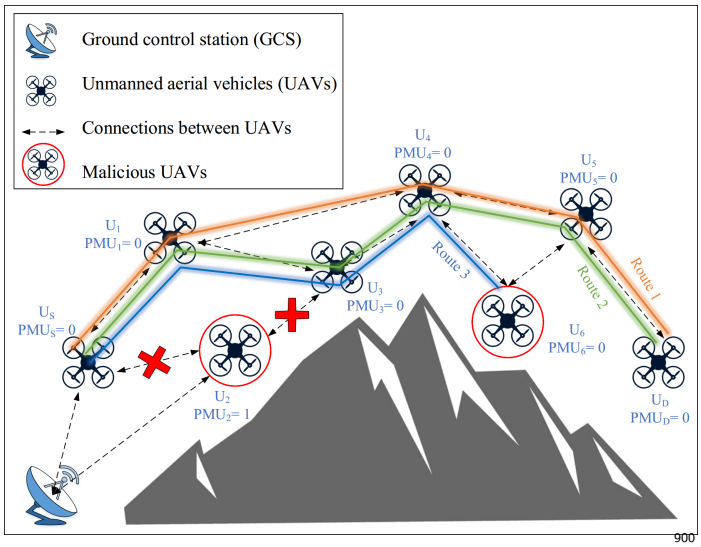

Figure 11: Finding various path between U_S and U_D.

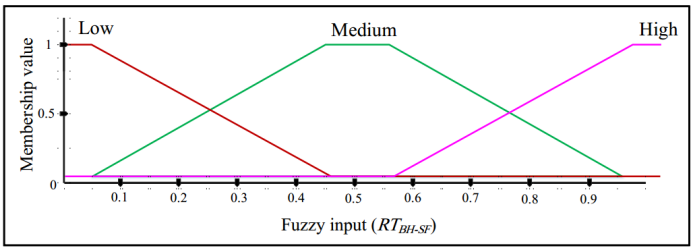

Figure 12: The diagram of MF associated with RT_{BH-SF}.

According to these scales mentioned above, the evaluation of the path trust (RT_R) is based on a fuzzy system. This system is responsible for evaluating the trust level of paths in accordance with these six scales. This fuzzy system follows the Mamdani fuzzy inference. It consists of three inputs, namely BH and SF-based path trust (RT_{BH-SF}), WH-based path trust (RT_{WH}), and flooding-based path trust (RT_F). Also, this system has an output, namely path trust (RT_R), and a knowledge base. Note that U_S is responsible for implementing the fuzzy system and determining the trust level of different paths.

5.4.1. Fuzzy inputs

In FTSR, the fuzzy trust system includes three inputs, namely RT_{BH-SF}, RT_{WH}, and RT_F.

- **BH and SF-based path trust ($\mathbf{RT_{BH-SF}}$):** When a black hole attack or a selective forwarding attack occurs on a path, it experiences a high PLR and low PRF. Therefore, RT_{BH-SF} is achieved according to Equation 34.

$$RT_{BH-SF} = \frac{\left(\frac{PRF_R}{\max\limits_{R\in RouteSet}(PRF_R)}\right)}{\left(\frac{PLR_R}{\max\limits_{R\in RouteSet}(PLR_R)}\right)} \tag{34}$$

  where R indicates the present path that must be assessed. $RouteSet$ is a set of all paths found between U_S and U_D. The diagram of the membership function (MF) associated with RT_{BH-SF} is illustrated in Figure 12. According to this figure, RT_{BH-SF} contains three modes namely low, medium, and high.

- **WH-based path trust ($\mathbf{RT_{WH}}$):** When a wormhole attack occurs on a path, the fake path shows lower hops than legal paths. While this fake path increases the end-to-end delay, PLR, and PTF and lowers PRF. According to [31], the authors prove that the number of hops in a legal path must be $\frac{D(U_S,U_D)}{R_T} \leq H_c \leq 2\left(\frac{D(U_S,U_D)}{R_T}\right)$ so that $D(U_S,U_D)$ is the distance between U_S and U_D, and R_T indicates the transmission radius. Now, if H_c of a path is less than $\frac{D(U_S,U_D)}{R_T}$, there is likely a WH attack on this path. While if H_c of a path is more than $2\left(\frac{D(U_S,U_D)}{R_T}\right)$, it will have an adverse effect on network performance. In these two cases, it lowers the WH-based path trust (RT_{WH}). As a result, RT_{WH} can be calculated using Equation 35.

  where R is the present path that must be assessed. $RouteSet$ indicates a set of the discovered paths between U_S and U_D. The diagram of MF associated with RT_{WH} is presented with RT_{WH} in Figure 13. As shown in this figure, RT_{WH} contains three modes, namely low, medium, and high.

- **Flooding-based path trust ($\mathbf{RT_F}$):** When a flooding attack occurs on the path, it increases PTF, ECR, and the end-to-end delay in that path. Therefore, RT_F is calculated through Equation 36.

$$RT_F = \frac{1}{\left(\frac{ECR_R}{\max\limits_{R\in RouteSet}(ECR_R)}\right) + \left(\frac{T_R}{\max\limits_{R\in RouteSet} T_R}\right) + \left(\frac{PTF_R}{\max\limits_{R\in RouteSet} PTF_R}\right)} \tag{36}$$

  where R and $RouteSet$ are the current path and the set of all paths found between U_S and U_D, respectively. The diagram of MF associated with RT_F is shown in Figure 14. According to this figure, RT_F contains three modes, namely low, medium, and high.

5.4.2. Fuzzy output

The output of this system is the path trust (RT_R), which is a combination of three fuzzy inputs (i.e. RT_{BH-SF}, RT_{WH}, and RT_F). It includes seven modes (i.e. extremely low, very low, low, medium, high, very high, and extremely high). The diagram of MF related to RT_R is shown in Figure 15.

5.4.3. Knowledge base

The proposed system considers the rules presented in Table 3. For example, Rule 1 will be stated below.

Rule 1: IF RT_{BH-SF} is *Low* **AND** RT_{WH} is *Low* **AND** RT_F is *low* **THEN** RT_R is *Extremely low*.

Algorithm 4 expresses the pseudo-code related to the path selection procedure. In the following, the time complexity of

$$RT_{WH} = \begin{cases} \dfrac{\left(\dfrac{PRF_R}{\max\limits_{R\in RouteSet}(PRF_R)}\right)+\left(\dfrac{H_c}{\max\limits_{R\in RouteSet}(H_c)}\right)}{\left(\dfrac{T_R}{\max\limits_{R\in RouteSet}T_R}\right)+\left(\dfrac{PLR_R}{\max\limits_{R\in RouteSet}(PLR_R)}\right)+\left(\dfrac{PTF_R}{\max\limits_{R\in RouteSet}(PTF_R)}\right)}, & \dfrac{D(U_S,U_D)}{R} \le H_c \le 2\left(\dfrac{D(U_S,U_D)}{R}\right) \\ \dfrac{\left(\dfrac{PRF_R}{\max\limits_{R\in RouteSet}(PRF_R)}\right)}{\left(\dfrac{T_R}{\max\limits_{R\in RouteSet}T_R}\right)+\left(\dfrac{PLR_R}{\max\limits_{R\in RouteSet}(PLR_R)}\right)+\left(\dfrac{PTF_R}{\max\limits_{R\in RouteSet}(PTF_R)}\right)+\left(\dfrac{H_c}{\max\limits_{R\in RouteSet}(H_c)}\right)}, & Otherwise \end{cases} \tag{35}$$

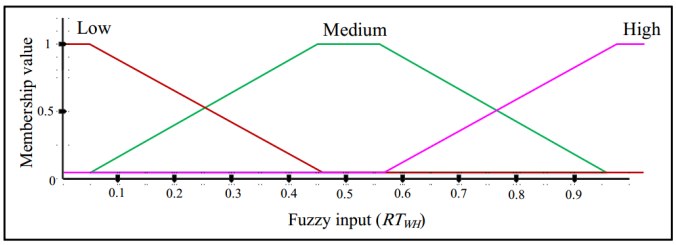

Figure 13: The diagram of MF associated with RT_{WH}.

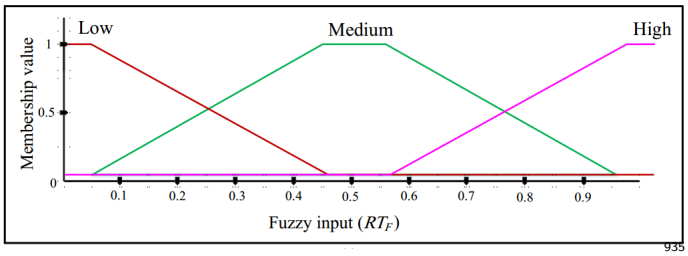

Figure 14: The diagram of MF related to RT_F.

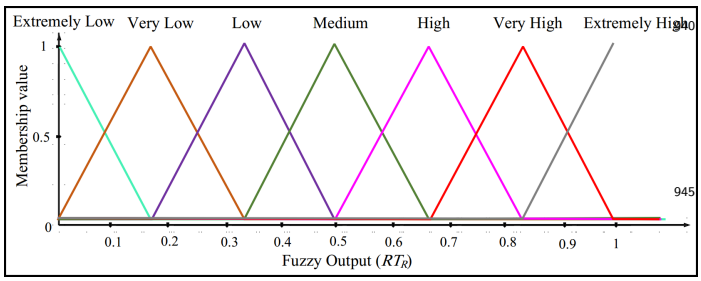

Figure 15: The diagram of MF corresponding to RT_R.

Table 3: Knowledge base in the proposed fuzzy-based trust system.

	Fuzzy system inputs			Fuzzy system output
Fuzzy rules	RT_{BH-SF}	RT_{WH}	RT_F	RT_R
1	Low	Low	Low	Extremely low
2	Low	Low	Medium	Very low
3	Low	Low	High	Low
4	Low	Medium	Low	Very low
5	Low	Medium	Medium	Low
6	Low	Medium	High	Medium
7	Low	High	Low	Low
8	Low	High	Medium	Medium
9	Low	High	High	High
10	Medium	Low	Low	Very low
11	Medium	Low	Medium	Low
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Low
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Very high
16	Medium	High	Low	Medium
17	Medium	High	Medium	High
18	Medium	High	High	Very high
19	High	Low	Low	Low
20	High	Low	Medium	Medium
21	High	Low	High	High
22	High	Medium	Low	Medium
23	High	Medium	Medium	High
24	High	Medium	High	Very high
25	High	High	Low	High
26	High	High	Medium	Very high
27	High	High	High	Extremely high

this algorithm is analyzed. It consists of 8 commands (in lines 1-8) whose time complexities are determined by Equations 34, 35 and 36.

- The time complexity of Equation 34 is $O(N)$.
- The time complexity of Equation 35 is $O(N)$.
- The time complexity of Equation 36 is $O(N)$.

Therefore, the time complexity of Algorithm 4 is $O(N)$.

5.5. Path maintenance phase

At this phase, the path maintenance procedure is carried out to restart the path discovery procedure when occurring a route failure. U_S periodically examines whether the paths available in the routing table are still connected. In order to validate the paths, U_S regularly sends a path validation message through the existing path. If this message is successfully gotten by U_D, it will respond by sending an acknowledgment message (ACK) to confirm that this path is connected. Otherwise, if U_S does not get any ACK message from U_D after a certain time, then U_S will be informed of the disconnected route and will begin the path discovery procedure to get new paths in accordance with Section 5.3.

Algorithm 4 Route selection process based on the fuzzy route trust evaluation

Input: U_i: UAVs in the networks so that $i = 1, 2, ..., N$
U_S: Source UAV
U_D: Destination UAV
Output: Selecting a path between U_S and U_D.
Begin
1: **U_S:** Calculate the route trust based on BH and SF attacks (RT_{BH-SF}) using Equation 34;
2: **U_S:** Convert RT_{BH-SF} to a fuzzy variable using the fuzzy membership function (FMF) presented in Figure 12;
3: **U_S:** Compute the route trust based on WH attack (RT_{WH}) according to Equation 35;
4: **U_S:** Fuzzify RT_{WH} according to the FMF displayed in Figure 13;
5: **U_S:** Obtain the route trust based on flooding attack (RT_F) from Equation 36;
6: **U_S:** Transform RT_F into a fuzzy variable using the FMF represented in Figure 14;
7: **U_S:** Calculate the fuzzy value of the route trust (RT_R) based on the proposed fuzzy system;
8: **U_S:** Convert the fuzzy vale of RT_R to a crisp value using the FMF presented in Figure 15;
End

Algorithm 5 illustrates the pseudo-code related to the process whose time complexity is presented below. Algorithm 5 contains a *while* loop (in lines 2-15) that is repeated at the simulation time (t_s). This *while* loop contains an *IF* condition (in lines 3-13) to measure the time of sending a path validation message. This *IF* condition includes the following commands:

- A command (line 4) that has a fixed run time h_1.
- A *IF* condition (lines 6-8) with a fixed run time h_2.
- A command (line 9) with a fixed run time h_3.
- A *IF* condition (lines 10-12) whose run time depends on Algorithm 3. The time complexity of Algorithm 3 is $O(N)$.

Therefore, the time complexity of Algorithm 5 is calculated below.

$$T(n) = t_s(h_1 + h_2 + h_3 + N) \tag{37}$$

Suppose there is a fixed number h so that $h \geq h_1 + h_2 + h_3$:

$$T(n) = t_s(h_1 + h_2 + h_3 + N) \leq h t_s N \tag{38}$$

Therefore, the time complexity of Algorithm 5 is $O(t_s N)$.

6. Simulation and result evaluation

For performance and efficiency analysis, the simulation operation of FTSR is performed using Network simulator 2 (NS2). This operation considers the simulation parameters introduced in Table 4. According to this table, the network is $800 \times 800 \times 800\ m^3$ for the simulation operation. In this network, the number of nodes and packets varies between 10-70 nodes and 30-300 packets, respectively. The movement of UAVs in the FANET is defined by the random waypoint model. The traffic model is the constant bite rate (CBR), and its rate is 1 Mbps. Finally, the performance of FTSR is examined with regard to the malicious detection rate, PDR, PLR, accuracy, and delay, and the results are compared with those of TOPCM [14], MNRiRIP [14], and MNDA [19]. In the following, these evaluation criteria are introduced.

Algorithm 5 Route maintenance process

Input: U_S: Source UAV
U_D: Destination UAV
$VT = 0$: Timer
$t = 0$: Timer for simulation process
t_s: Simulation time
$RT_{Validation}$: Route validation period
Output: Checking the formed path between U_S and U_D
Begin
1: **U_S:** Check whether the discovered route between U_S and U_D is valid periodically;
2: **while** $t \leq t_s$ **do**
3: **if** $VT = RT_{Validation}$ **then**
4: **U_S:** Unicast a route validation message to U_D;
5: $VT = 0$;
6: **if** U_D receives the route validation message from U_S **then**
7: **U_D:** Send a ACK message to U_S;
8: **end if**
9: **U_S:** Wait for receiving the ACK message from U_D;
10: **if** U_S does not receive the ACK message form U_D **then**
11: **U_S:** Start the route discovery process based on Algorithm 3;
12: **end if**
13: **end if**
14: $VT = VT + 1$;
15: **end while**
End

Table 4: Simulation parameters.

Evaluation criteria	Value
Simulation tool	NS2
Network dimensions	$800 \times 800 \times 800\ m^3$
Mobility model	Random waypoint
Traffic model	Constant Bite Rate (CBR)
CBR rate	1 Mbps
The number of UAVs	10-70
The number of packets	40, 80, 120, 160, 200, 240, 280, 300
Simulation time	100 s
Traffic type	TCP
Communication range	200-300 m
Number of malicious nodes	15 % of the total network nodes

- **Malicious node detection rate:** It expresses the ability of the security mechanism to detect hostile nodes on the network.
- **Accuracy:** This evaluation criterion illustrates how many hostile nodes have been correctly identified by the security mechanism.
- **Packet delivery ratio (PDR):** This criterion is defined as the ratio of the packets gotten by U_D to all packets produced by U_S.
- **Packet loss rate (PLR):** It indicates the percentage of missing packets that have not reached U_D.
- **End-to-end delay (EED):** This criterion is defined as the average time when a data packet is produced by U_S until the moment of this packet arrives to U_D.

6.1. Malicious node detection rate and accuracy

Figure 16 shows a comparison of different schemes according to the detection rate. In this figure, FTSR has the highest detection rate, meaning that this parameter has been improved compared to TOPCM (about 11.85%), MNRiRIP (approximately 50.49%), and MNDA (about three times). Also, Figure 17 displays the accuracy of different schemes for detecting hostile UAVs correctly. According to this figure, FTSR is

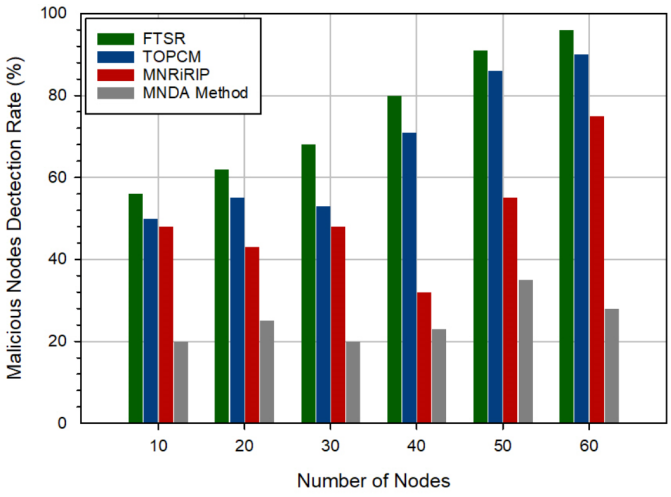

Figure 16: The detection rate of hostile nodes in different schemes.

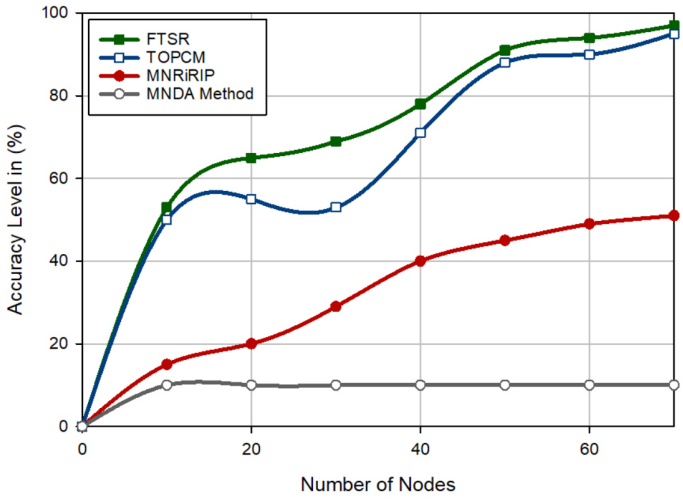

Figure 17: Accuracy evaluation in different schemes.

highly accurate and operates better (about 8.97%) than TOPCM. This proves that FTSR designs a robust security mechanism because this security mechanism uses two techniques, namely local trust and fuzzy-based path trust to detect hostile UAVs accurately. Another point that can be deduced from Figure 16 is that there is a direct relationship between the number of UAVs and the detection rate in FTSR and TOPCM. Therefore, when there are a high number of UAVs in the network, these schemes can better detect hostile UAVs and vice versa. However, there is no such relationship in MNRiRIP and MNDA. According to Figure 17, FTSR, TOPCM, and MNRiRIP have high accuracy for detecting hostile nodes when the density of UAVs is high. However, the accuracy of MNDA is almost constant, it has no change when increasing the number of UAVs. The main reason is that FTSR uses several parameters such as PLR, PRF, PTF, delay, the number of hops, and energy change rate to identify hostile nodes. When the UAV density is low, the probability of path failure is highly increased, and all UAVs experience a high number of missing packets. Thus, the designed security mechanism cannot correctly distinguish hostile UAVs from honest UAVs. While when the density is high, the interaction between UAVs has improved, and the abnormal behavior of hostile nodes is recognizable. As a result, FTSR better identifies these nodes. TOPCM evaluates the trust level of UAVs based on four scales, including broadcast ID, destination address, the next-hop ID, and the current node ID. Thus, when the density of the UAVs is high, more suitable paths are created between UAVs, and their trust level is more accurately evaluated. However, MNDA performs the malicious detection process only based on delay. This has reduced its ability to detect hostile UAVs accurately.

6.2. *Packet delivery rate and packet loss rate*

Figure 18 illustrates the evaluation results of different schemes in terms of PDR. In this figure, FTSR has a better PDR and increased this evaluation criterion by 11.58%, 35.25%, and 78.52% compared to TOPCM, MNiRIP, and MNDA, respectively. Figure 19 also shows the outcomes related to PLR in different schemes. According to this figure, FTSR has an acceptable performance with regard to PLR and reduces this evaluation criterion by 52.15%, 64.64%, and 71.49% compared to TOPCM, MNRiRIP, and MNDA, respectively. The main reason for the better performance of our scheme in terms of PDR and PLR is that FTSR uses a local trust mechanism to diagnose hostile UAVs. These nodes are isolated and cannot participate in the networking processes. On the other hand, after identifying all paths between U_S and U_D, the proposed fuzzy-based path trust system chooses the safest route. This increases the stability of the path and improves PDR in FSTR. Another important point in Figure 18 is that there is a direct relationship between PDR and the density of UAVs in all schemes. Whereas, according to Figure 19, PLR and density have a reverse relationship. These results are quite rational because when the UAV density is high, UAVs can find more paths between themselves, and consequently, the probability of the route stability will improve. Moreover, the results obtained in Section 6.1 show that increasing the density of UAVs improves the detection rate. This increases PDR in all schemes.

6.3. *Delay*

Figure 20 expresses the results of delay in different schemes. As shown in this figure, FTSR has an acceptable delay. It has reduced this evaluation criterion compared to MNRiRIP (about 6.35%) and MNDA (approximately 34.75%). However, TOPCM has approximately 21.34% less delay than FTSR. Note that the fuzzy-based trust mechanism designed by FTSR is a robust security mechanism, which can well detect hostile UAVs but the fuzzy mechanism imposes some computational costs on UAVs and causes a high delay in FTSR compared to that in TOPCM.

7. Conclusion

In this paper, a novel fuzzy trust-based secure routing scheme called FTSR was suggested for FANETs. In FTSR, each UAV

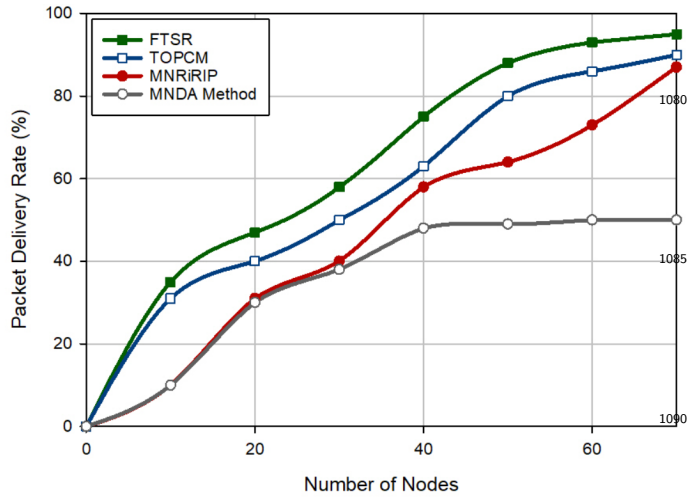

Figure 18: PDR in various schemes.

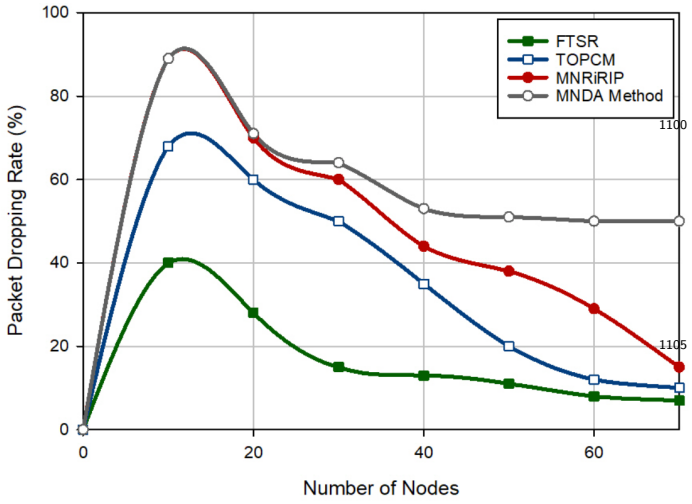

Figure 19: PLR in different schemes.

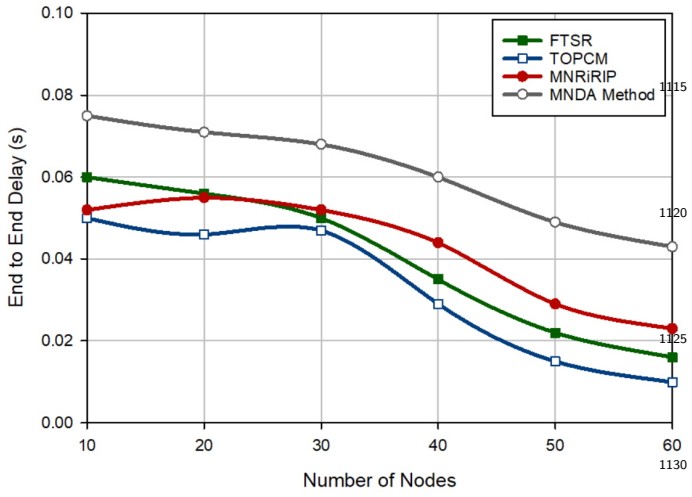

Figure 20: Delay in different schemes.

utilizes a distributed and local trust mechanism to identify reliable neighboring UAVs and separate hostile nodes. Thus, hostile UAVs cannot participate in the routing operation; this reduces the possibility of forming fake paths in the network. However, some attackers, such as Grey hole, may hide, and UAVs cannot identify them using the local trust mechanism. Detection of these hostile nodes is done at the path trust procedure. In this operation, the source UAV uses the fuzzy-based path trust mechanism. It is responsible for evaluating the trust level of paths based on three security scales, namely BH and SF-based path trust, WH-based path trust, and flooding-based path trust to select the safest route. Finally, the simulation process of FTSR is done using NS2, and its performance is evaluated according to the detection rate, PDR, PLR, accuracy, and delay. This evaluation shows that FTSR increased the detection rate compared to TOPCM (about 11.85%), MNRiRIP (approximately 50.49%), and MNDA (about three times), and its detection accuracy is higher (about 8.97%) than TOPCM. Also, FTSR improves PDR by 11.58%, 35.25%, 78.52%, and PLR by 52.15%, 64.64%, and 71.49% compared to TOPCM, MNRiRIP, and MNDA, respectively. However, FTSR increased delay by approximately 21.34% more than TOPCM. It is due to the computational costs of the security mechanism designed in this method. In future research directions, we are trying to design lightweight and robust security mechanisms based on new techniques such as neural networks and reinforcement learning to reduce delay in the routing process.

Acknowledgements

This study is supported via funding from Prince Sattam Bin Abdulaziz University project number (PSAU/2023/R/1444).

References

References

[1] Tsao, K.Y., Girdler, T. and Vassilakis, V.G., 2022. A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks. Ad Hoc Networks, p.102894.

[2] Bekmezci, I., Sahingoz, O.K. and Temel, Ş., 2013. Flying ad-hoc networks (FANETs): A survey. Ad Hoc Networks, 11(3), pp.1254-1270.

[3] M. Satell, 16 Eye-Opening Drone Stats for 2022, 2022, Philly By Air, July. 2022, URL https://www.phillybyair.com/blog/drone-stats/.

[4] Erceg, A., Erceg, B.Ĉ. and Vasilj, A., 2017. Unmanned aircraft systems in logistics–legal regulation and worldwide examples toward use in Croatia. Business Logistics in Modern Management.

[5] Fantin Irudaya Raj, E., 2022. Implementation of Machine Learning Techniques in Unmanned Aerial Vehicle Control and Its Various Applications. In Computational Intelligence for Unmanned Aerial Vehicles Communication Networks (pp. 17-33). Springer, Cham.

[6] L. Schroth, The drone market size 2020–2025: 5 key takeaways, 2020, DRONEII, 22nd, Jun. 2020, URL https://droneii.com/the-drone-market-size-2020-2025-5-key-takeaways.

[7] Skies without limits, 2021, PwC, URL https://www.pwc.co.uk/intelligent-digital/drones/Drones-impact-on-the-UK-economy-FINAL.pdf.

[8] Bombe, M.K., Unmanned Aerial Vehicle (UAV) Market Worth $21.8 billion by 2027-Pre and Post COVID-19 Market Analysis Report by Meticulous Research, 11 June 2020.

[9] Tlili, F., Fourati, L.C., Ayed, S. and Ouni, B., 2022. Investigation on vulnerabilities, threats and attacks prohibiting UAVs charging and depleting UAVs batteries: Assessments & countermeasures. Ad Hoc Networks, 129, p.102805.

[10] Srivastava, A. and Prakash, J., 2021. Future FANET with application and enabling techniques: Anatomization and sustainability issues. Computer Science Review, 39, p.100359.

[11] Lansky, J., Ali, S., Rahmani, A.M., Yousefpoor, M.S., Yousefpoor, E., Khan, F. and Hosseinzadeh, M., 2022. Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review. Mathematics, 10(16), p.3017.

[12] Ahmad, S. and Hassan, M.A., 2022. Secure Communication Routing in FANETs: A Survey. In Computational Intelligence for Unmanned Aerial Vehicles Communication Networks (pp. 97-110). Springer, Cham.

[13] Sangeetha Francelin, V.F., Daniel, J. and Velliangiri, S., 2022. Intelligent agent and optimization-based deep residual network to secure communication in UAV network. International Journal of Intelligent Systems.

[14] Buksh, W., Guo, Y., Iqbal, S., Qureshi, K.N. and Lloret, J., 2022. Trust-oriented peered customized mechanism for malicious nodes isolation for flying ad hoc networks. Transactions on Emerging Telecommunications Technologies, p.e4489.

[15] Fotohi, R., Nazemi, E. and Aliee, F.S., 2020. An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks. Vehicular Communications, 26, p.100267.

[16] Du, X., Li, Y., Zhou, S. and Zhou, Y., 2022. ATS-LIA: A lightweight mutual authentication based on adaptive trust strategy in flying ad-hoc networks. Peer-to-Peer Networking and Applications, pp.1-15.

[17] Agron, D.J.S., Ramli, M.R., Lee, J.M. and Kim, D.S., 2019, October. Secure ground control station-based routing protocol for UAV networks. In 2019 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 794-798). IEEE.

[18] Bhardwaj, V., Kaur, N., Vashisht, S. and Jain, S., 2021. SecRIP: Secure and reliable intercluster routing protocol for efficient data transmission in flying ad hoc networks. Transactions on Emerging Telecommunications Technologies, 32(6), p.e4068.

[19] Muruganandam, D. and Martin Leo Manickam, J., 2019. An efficient technique for mitigating stealthy attacks using MNDA in MANET. Neural Computing and Applications, 31(1), pp.15-22.

[20] Rahmani, A.M., Ali, S., Yousefpoor, E., Yousefpoor, M.S., Javaheri, D., Lalbakhsh, P., Ahmed, O.H., Hosseinzadeh, M. and Lee, S.W., 2022. OLSR+: A new routing method based on fuzzy logic in flying ad-hoc networks (FANETs). Vehicular Communications, p.100489.

[21] Lee, S.W., Ali, S., Yousefpoor, M.S., Yousefpoor, E., Lalbakhsh, P., Javaheri, D., Rahmani, A.M. and Hosseinzadeh, M., 2021. An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets). IEEE Access, 9, pp.129977-130005.

[22] Zadeh, L.A., 1994. Soft computing and fuzzy logic. IEEE software, 11(6), pp.48-56.

[23] Zadeh, L., 1996. Fuzzy Logic= Computing with Words IEEE Transactions on Fuzzy Systems, vol. 2.

[24] Rahmani, A.M., Naqvi, R.A., Yousefpoor, E., Yousefpoor, M.S., Ahmed, O.H., Hosseinzadeh, M. and Siddique, K., 2022. A Q-Learning and Fuzzy Logic-Based Hierarchical Routing Scheme in the Intelligent Transportation System for Smart Cities. Mathematics, 10(22), p.4192.

[25] Kulkarni, R.V. and Forster, A., 2011. GK Venaya gamoorthy,". Computational Intelligence in Wireless Sensor Networks": A Survey, Journal of IEEE Communications Surveys &Tutorials, 13(1), pp.68-96.

[26] Stallings, W., 2004. IEEE 8O2. 11: wireless LANs from a to n. IT professional, 6(5), pp.32-37.

[27] Chen, Y., Zhao, N., Ding, Z. and Alouini, M.S., 2018. Multiple UAVs as relays: Multi-hop single link versus multiple dual-hop links. IEEE Transactions on Wireless Communications, 17(9), pp.6348-6359.

[28] Yousefpoor, M.S., Yousefpoor, E., Barati, H., Barati, A., Movaghar, A. and Hosseinzadeh, M., 2021. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. Journal of Network and Computer Applications, 190, p.103118.

[29] Chriki, A., Touati, H., Snoussi, H. and Kamoun, F., 2019. FANET: Communication, mobility models and security issues. Computer Networks, 163, p.106877.

[30] Bhushan, B., Sahoo, G. and Rai, A.K., 2017, September. Man-in-the-middle attack in wireless and computer networking—A review. In 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall) (pp. 1-6). IEEE.

[31] Khabbazian, M., Mercier, H. and Bhargava, V.K., 2009. Severity analysis and countermeasure for the wormhole attack in wireless ad hoc networks. IEEE Transactions on Wireless Communications, 8(2), pp.736-745.

[32] Messaoudi, K., Oubbati, O.S., Rachedi, A., Lakas, A., Bendouma, T. and Chaib, N., 2023. A survey of UAV-based data collection: Challenges, solutions and future perspectives. Journal of Network and Computer Applications, p.103670.

[33] Lateef, S., Rizwan, M. and Hassan, M.A., 2022. Security Threats in Flying Ad Hoc Network (FANET). Computational Intelligence for Unmanned Aerial Vehicles Communication Networks, pp.73-96.

[34] Zhang, M., Dong, C., Yang, P., Tao, T., Wu, Q. and Quek, T.Q., 2022. Adaptive routing design for flying ad hoc networks. IEEE Communications Letters, 26(6), pp.1438-1442.

[35] Zheng, B., Zhuo, K., Zhang, H. and Wu, H.X., 2022. A novel airborne greedy geographic routing protocol for flying Ad hoc networks. Wireless Networks, pp.1-15.

[36] Kundu, J., Alam, S. and Koner, C., 2022, August. TCSFANET: Trusted Communication Scheme for FANET system. In 2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS) (pp. 353-357). IEEE.

[37] Yu, S., Lee, J., Sutrala, A.K., Das, A.K. and Park, Y., 2023. LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad-hoc networks. Computer Networks, 224, p.109612.

[38] Zhu, R., Boukerche, A., Feng, L. and Yang, Q., 2023. A trust management-based secure routing protocol with AUV-aided path repairing for Underwater Acoustic Sensor Networks. Ad Hoc Networks, p.103212.

[39] Yang, J., Sun, K., He, H., Jiang, X. and Chen, S., 2022. Dynamic virtual topology aided networking and routing for aeronautical ad-hoc networks. IEEE Transactions on Communications, 70(7), pp.4702-4716.

[40] Rahman, K., Aziz, M.A., Kashif, A.U. and Cheema, T.A., 2022. Detection of Security Attacks Using Intrusion Detection System for UAV Networks: A Survey. In Big Data Analytics and Computational Intelligence for Cybersecurity (pp. 109-123). Cham: Springer International Publishing.