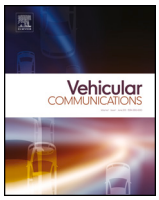
Vehicular Communications 49 (2024) 100826

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom



A novel Q-learning-based secure routing scheme with a robust defensive system against wormhole attacks in flying ad hoc networks

Mehdi Hosseinzadeh [a,b], Saqib Ali [c], Husham Jawad Ahmad [d], Faisal Alanazi [e], Mohammad Sadegh Yousefpoor [f], Efat Yousefpoor [f], Omed Hassan Ahmed [g], Amir Masoud Rahmani [h,*], Sang-Woong Lee [i,*]

[a] Institute of Research and Development, Duy Tan University, Da Nang, Viet Nam
[b] School of Medicine and Pharmacy, Duy Tan University, Da Nang, Viet Nam
[c] Department of Information Systems, College of Economics and Political Science, Sultan Qaboos University, Al Khoudh, Muscat, Oman
[d] Department of Communication and Computer Engineering, Cihan University-Erbil, Kurdistan Region, Iraq
[e] Department of Electrical Engineering, College of Engineering, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[f] Center of Research and Strategic Studies, Lebanese French University, Kurdistan Region, Iraq
[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] Pattern Recognition and Machine Learning Lab, Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam 13120, Republic of Korea

ARTICLE INFO

Keywords:
Flying ad hoc networks (FANETs)
Reinforcement learning (RL)
Security
Routing
Machine learning (ML)

ABSTRACT

Nowadays, unmanned aerial vehicles (UAVs) organized in a flying ad hoc network (FANET) can successfully carry out complex missions. Due to the limitations of these networks, including the lack of infrastructure, wireless communication channels, dynamic topology, and unreliable communication between UAVs, cyberattacks, especially wormholes, weaken the performance of routing schemes. Therefore, maintaining communication security and guaranteeing the quality of service (QoS) are very challenging. In this paper, a novel Q-learning-based secure routing scheme (QSR) is presented for FANETs. QSR seeks to provide a robust defensive system against wormhole attacks, especially wormhole through encapsulation and wormhole through packet relay. QSR includes a secure neighbor discovery process and a Q-learning-based secure routing process. Firstly, each UAV gets information about its neighboring UAVs securely. To secure communication in this process, a local monitoring system is designed to counteract the wormhole attack through packet relay. This system checks data packets exchanged between neighboring UAVs and defines three rules according to the behavior of wormholes. In the second process, UAVs perform a distributed Q-learning-based routing process to counteract the wormhole attack through encapsulation. To reward the safest paths, a reward function is introduced based on five factors, the average one-hop delay, hop count, data loss ratio, packet transmission frequency (PTF), and packet reception frequency (PRF). Finally, the NS2 simulator is applied for implementing QSR and executing different scenarios. The evaluation results show that QSR works better than TOPCM, MNRiRIP, and MNDA in terms of accuracy, malicious node detection rate, data delivery ratio, and data loss ratio. However, it has more delay than TOPCM.

1. Introduction

Flying ad hoc network (FANET) is dynamic and self-organized and includes a number of unmanned aerial vehicles (UAVs), which cooperate with each other to do a specific task [1,2]. UAVs is responsible for carrying out missions such as navigation, industrial inspections, advanced mapping and imaging, search and rescue operations in large areas. In the network, drones connect to each other through the Internet of Drones (IoD). In FANET, guaranteeing security and increasing the quality of service (QoS) are very important because FANETs are often used in insecure and dangerous environments [3–5]. In these networks, a routing scheme is responsible for creating a valid path, which contains several hops (i.e. intermediate nodes) for forwarding data packets in the network. Routing protocols designed in FANET must consider various issues such as QoS requirements, energy limitation, and link stability to create reliable paths between nodes in the network. Today, many routing pro-

* Corresponding authors.
E-mail addresses: rahmania@yuntech.edu.tw (A.M. Rahmani), slee@gachon.ac.kr (S.-W. Lee).

https://doi.org/10.1016/j.vehcom.2024.100826
Received 19 December 2023; Received in revised form 16 May 2024; Accepted 20 June 2024
Available online 3 July 2024

tocols are designed for FANETs. These methods can be categorized into several groups, including topology-based routing, geographic routing, machine learning-based routing, swarm-based routing, and hybrid routing [6,7]. Due to the limitations of these networks, including the lack of infrastructure, wireless communication channels, dynamic topology, and unreliable communication between UAVs, cyberattacks have a bad effect on routing protocols in FANET [8–10]. A common hypothesis in most routing protocols is that UAVs are trustable. These protocols often neglect security issues. As a result, they provide many opportunities for attackers to carry out cyberattacks [11,12]. Therefore, maintaining communication security and guaranteeing services are very important issues. This paper focuses on a very serious attack called "***Wormhole***". This cyberattack is a destructive threat to FANET and affects network communication links [13,14]. FANETs are potentially susceptible to this cyberattack due to their open architecture and dynamic topology. Today, the number of these threats is significantly increasing on FANETs [15]. Therefore, many researchers are investigating facilities to secure these systems and guarantee the security of drones as an important part of the Internet of Things. It is difficult to deal with these threats and ensure network security because common approaches, such as intrusion detection systems or encryption, are not enough to combat such threats. Thus, FANET requires a defensive system to prevent damage to drones because they are an attractive goal for cyberattacks, such as wormholes. The detection of this attack is very challenging because wormholes can be implemented in FANET without accessing secret keys, capturing UAVs, or knowing the routing protocol [16–18]. As a result, most powerful key cryptography methods cannot neutralize this attack and protect communication. A wormhole does not require the content of control packets. It is very difficult to detect wormholes because they make no change to the contents of the packets [19–21].

In a wormhole attack, a malicious UAV constantly checks wireless communication channels and captures the exchanged packets [22,23]. Then, these packets will be tunneled into another malicious UAV. Lastly, it disseminates these packets locally in its surrounding area [24,25]. Hence, UAVs close to these attackers cannot find legal routes in the network because wormholes announce attractive paths with one or two hops, while legal paths have more hops. After performing this subversive operation, network performance will be sharply weak [26,27]. The attack can negatively affect a wide range of routing protocols because it can provide suitable conditions for implementing other attacks such as selective forwarding (SF), traffic analysis, or denial of service (DoS) attacks [28,29]. In general, malicious nodes launch wormholes in two modes, namely hidden and participation. In the hidden mode, malicious UAVs hide their identifiers in FANETs [30,31]. In this case, they play the role of simple transceivers. These hostile UAVs sniff data packets on one side of the wormhole tunnel. Then, they capture and propagate these packets on the other side of this tunnel [32,33]. Clearly, this wormhole attack does not require any secret key. In the hidden mode, there are two types of wormholes, namely encapsulation and packet relay [34,35]. In packet encapsulation, the first attacker performs the encapsulation operation as soon as it captures a packet. This prevents the increase in hop counts in the original packet. Then, the encapsulated packet is sent to the second attacker through the wormhole tunnel [36–38]. In the packet relay, the network can be threatened by only one malicious UAV. This attacker rebroadcasts packets received from a legal UAV to other distant UAVs. As a result, these distant UAVs mistakenly imagine that they are neighbors. In the participation mode, attackers need secret keys in the network to run powerful attacks on FANET [39–41]. These attackers act as legal UAVs and play the role of intermediate nodes in routing paths. Wormholes increase the attractiveness of their illegal routes in terms of low delay and least hops [42,43]. In the participation mode, there are two types of wormholes, namely, high transmission power and out-of-band channel. In the wormhole with high transmission power, the attacker has a high transmission power [44,45]. This feature increases the attractiveness of this fake route. In the wormhole through the out-of-band channel, hostile UAVs use an external channel with high bandwidth to build an attractive path in FANET [46,47].

Due to the limited resources, especially bandwidth and energy, in FANET, it is very challenging to design defensive mechanisms against this dangerous attack. In the research studies such as [48–50], many solutions have been suggested to detect wormhole attacks in ad hoc networks, but special attention to this area in FANETs is very low. On the other hand, most existing security solutions in ad hoc networks require specific tools to provide the location of UAVs or time synchronization [51–53]. These solutions usually detect wormholes, but they cannot isolate hostile nodes on the network [54–56]. Therefore, there is a research gap in the field of detection and isolation of wormholes in FANETs. Additionally, some security solutions have used artificial intelligence (AI) and machine learning (ML) to combat wormhole attacks [57–59]. AI and ML have a lot of potential to solve problems related to the detection and isolation of wormholes in FANETs [60–64]. However, the number of AI-based and ML-based security solutions in FANET is very small. Therefore, the goal of this present research is to fill the existing gaps in this field using ML techniques, especially reinforcement learning (RL), which is a suitable option for solving distributed problems such as routing. Most existing ML-based routing algorithms use RL techniques, especially Q-learning, to build routing paths in the network [65–67]. The key advantage of RL-based algorithms is the online and continuous learning of the FANET environment. These methods learn from experiences. Hence, they do not need topology prediction techniques or channel estimation strategies. Although they take a long time to converge the optimal solution. These schemes are easily implemented and have high flexibility against topological changes. Hence, they can achieve better and more accurate results [68–70].

In this paper, a novel Q-learning-based secure routing approach (QSR) is proposed to neutralize wormhole attacks in FANETs. In general, QSR consists of the secure neighbor discovery process and the Q-learning-based secure routing process. Firstly, each UAV is aware of its neighboring UAVs and establishes a neighbor table. To secure this process, a local monitoring system is presented to detect and isolate wormholes. In the second phase, UAVs decide on the most secure routes using a distributed Q-learning algorithm. In the following, the innovations in this paper are listed:

- In QSR, a secure neighbor discovery phase is provided. To secure this process, each UAV monitors its adjacent UAVs to detect and isolate wormholes. In this phase, the local monitoring system defines three rules based on the behavior of wormholes. If a UAV violates these rules, it is detected and isolated as a wormhole.
- In QSR, a distributed Q-learning-based routing framework is designed. This framework proposes a new reward function by evaluating the behavior of wormholes and their effect on the five important factors, namely the average one-hop delay, hop counts, packet loss ratio (PLR), packet transmission frequency (PTF), and packet reception frequency (PRF). Furthermore, Q-learning parameters, namely learning rate and discount factor, are calculated dynamically to increase the compatibility of QSR with FANET.
- In the simulation process, QSR is compared with TOPCM, MNRiRIP, and MNDA in terms of malicious node detection rate, data delivery ratio, data loss ratio, accuracy, and latency. These evaluations show that the proposed method works better than other schemes. QSR has a high accuracy to detect malicious UAVs in the network. It improves the detection rate of malicious UAVs. However, it has more delay than TOPCM in the routing process.

The structure of this paper is as follows: Section 2 introduces the research works related to reinforcement learning and secure schemes on FANETs. Section 3 presents the basic principles and concepts of reinforcement learning. In Section 4, both network and threat models in QSR are explained. In Section 5, the details of QSR are introduced. Section 6 presents the security analysis of QSR. In Section 7, the simulation

process and its results are described. Finally, Section 8 concludes this paper.

2. Related works

Here, the related works are briefly presented in two sub-sections, namely routing methods and secure routing schemes.

2.1. Routing methods

To increase the efficiency and reliability of the communication in FANET, many researchers have focused on routing protocols in FANETs. In this research, some researchers provide location-based routing protocols focusing on the classical computing field so that these methods use mathematical ideas to obtain optimal paths in the network. However, the classic computing-based routing protocols often are not scalable, and their performance is weakened in large-scale FANETs due to features such as dynamic topology, and high speed of UAVs.

For example, in [78], a greedy perimeter stateless routing scheme based on a location estimation system called GPSR+ is introduced in FANETs. It uses a weighted linear regression model-based location estimation strategy to forecast the next location of each flying node and modify the dissemination time interval of hello packets. This increases the adaptability of GPSR+ to FANET. In addition, GPSR+ chooses a set of candidate nodes using the spherical removal strategy to find the next forwarder in the communication path. It singles out the most stable node toward the destination as the next forwarder. Evaluations illustrate that GPSR+ is powerful and efficient.

In [79], the authors introduced a geographic location-based opportunistic routing approach called GPHLOR for FANETs. This method merges the features of both hopless and position-based routing schemes. GPHLOR considers the hopeless idea to benefit from transfer opportunities of all communication links. It also chooses relay nodes with regard to their location regardless of the network topology because GPHLOR is compatible with topology changes in FANET. In this scheme, UAVs get their sending priorities based on their location information. This technique does not need to exchange data in the network and lowers routing overhead. Experiments indicate that GPHLOR works well in terms of PDR, latency, throughput, and overhead.

In [82], the authors offered a data dissemination approach called SF-GoeR for FANET. Note that SF-GoeR is rooted in greedy perimeter stateless routing (GPSR) and improves healthcare applications. This scheme connects FANET to the wireless body sensor network (WBSN). In this scheme, biosensors produce health data in WBSN, and UAVs forward this health data to the hospital immediately. SF-GoeR is responsible for disseminating essential information between doctors and patients. In SF-GoeR, the authors proposed a stability coefficient to reduce path failures. Two key scales namely the nearness ratio and the remaining energy ratio are considered to calculate the stability coefficient. Lastly, SF-GoeR is compared with GPSR-WG and GPSR with regard to PDR, network longevity, and latency. The results represent the successful performance of SF-GoeR.

In [83], the authors presented a data transmission scheme in FANET. It merges directional and Omnidirectional strategies to adjust flight angles dynamically. Also, it integrates geocast and unicast routing strategies. In this approach, the 3D location of intermediate UAVs is predicted based on their movement direction. This prediction increases the adaptability of this scheme to topology changes. Furthermore, this scheme reduces disconnections in the network, and consequently, improves path reliability, network lifetime, and successful data transmission. In this solution, when the route request (RREQ) reaches the destination, it utilizes directional beamforming with regard to the route information and location to send a route reply (RREP). This technique reduces the transmission distance, shortens the route construction time, and avoids packet collision. The experimental results show that this approach is compatible with the characteristics of FANET.

Recently, routing methods based on modern computing techniques, such as meta-heuristic algorithms and machine learning, have received much attention from the scientific community. These methods are often more scalable than classic routing methods and work well in FANETs. In the following, some of these methods are introduced.

In [80], the authors proposed a Q-learning-based routing method called QFAN for FANETs. QFAN comprises two components, namely path discovery and path maintenance. The first component builds a Q-learning-based routing procedure. Moreover, the authors suggested a filtering scale to decrease the search area. In the second component, QFAN traces and rebuilds broken paths. Finally, the evaluation results of QFAN show that it exceeds other approaches in accordance with latency, PDR, consumed energy, and network longevity. However, the overhead in QFAN is high slightly.

In [81], the authors suggested a gray wolf optimization (GWO)-based routing approach called GW-COOP, which applies a collaborative diversity technique. This method finds routing paths efficiently and is suitable for regular flying missions since GW-COOP mimics the hierarchy of gray wolves. GW-COOP guarantees reliable communication links between UAVs. This approach simulates the hierarchy of leadership and the hunting strategy in the behavior of gray wolves. GW-COOP applies a collaborative strategy to find the best paths in terms of energy, location of UAVs, and distance to the desired node. Furthermore, in this approach, the signal-to-noise ratio (SNR) and the link state affect the routing management strategy. GW-COOP is compared with BAT-COOP and BAT-FANET, and simulation results show that GW-COOP has better performance than the two methods in terms of data loss rate, consumed energy, one-hop latency, and PLR.

In [84], the authors suggested a Q-learning-based topology-aware routing scheme called QTAR. It collects information about one-hop and two-hop neighbors to build reliable routes. For this reason, this technique improves the path discovery procedure, reduces the path construction time, and improves decision-making about the next forwarder. However, this strategy increases the routing overload and the complexity of the system. QTAR uses location, latency, speed, and energy when choosing the next hop. Moreover, in this scheme, learning parameters are calculated according to network conditions. QTAR is compatible with 3D environments, like FANETs.

In [85], the authors presented a Q-learning-based routing approach called Q-FANET in FANET. This approach merges two Q-learning routing schemes, called QMR and Q-Noise+ to decide on the communication paths. Q-FANET employs an improved Q-learning algorithm named Q-learning+ to create the learning model. In addition, Q-FANET takes into account the channel conditions when updating Q-values in the Q-table. In Q-FANET, the signal-to-interference-plus-noise ratio (SINR) is considered to evaluate the quality of transmissions. Moreover, this scheme defines a speed constraint to decrease latency in the routing process. Also, a penalty structure is designed to prevent routing holes. Thus, when a node is trapped in a routing hole or the next forwarder does not respond to its previous-hop node, its reward value will be minimal. In this case, the probability of selecting this UAV in the communication path is reduced in the future. Evaluations approve the performance of this scheme.

2.2. Secure routing schemes

Routing protocols in FANETs are often vulnerable to routing attacks. To solve this problem, many researchers have focused on designing secure routing methods to combat various attacks.

In [71], a security system called TSPO-based DRN is offered to secure communication links between UAVs in FANET. This system has been tested for three cyberattacks, namely selective forwarding (SF), sinkhole (SH), and wormhole (WH). These tests show that TSPO-based DRN successfully counteracts these attacks. In TSPO-based DRN, a new optimization algorithm called TSPO is derived from Political Optimizer (PO) and Tunicate Swarm Algorithm (TSA) to carry out the routing process

between UAVs. TSPO finds the best routes according to two routing metrics, namely connection quality and distance. In TSPO-based DRN, the security system includes three virtual agents: evaluation agent, decision-making agent, and defense agent. The first agent controls and evaluates the behavior of drones and keeps the relevant information in an evaluation table. The decision-making agent employs the information available in this table along with a deep learning technique, deep residual network (DRN), to evaluate routes based on round trip time, signal strength, the size of packet, number of the received packets, and number of incoming packets. Lastly, the defense agent uses test packets to determine whether these routes are legal or illegal.

In [72], a trusted clustering method called TBCS is designed for FANET. In TBCS, a multi-criteria fuzzy classification method is presented to divide UAVs into three classes, namely trustable, hostile, and suspicious. This scheme identifies hostile nodes using two metrics, namely direct trust and recommended trust. In addition, TBCS can detect various attacks, including WH, SH, bad-mouthing, and SF. It considers five scales, namely signal strength, packet delivery ratio, residual energy, past interactions, and transmission delay to obtain the trust of UAVs based on a TSK fuzzy inference method. In this mechanism, a reward and penalty-based technique is used to detect hostile UAVs accurately and quickly. In each cluster, a cluster head is selected based on the trust value. It communicates with different clusters and the ground control station (GCS).

In [73], the authors present a security system to find malicious drones in FANETs. They propose a trust-oriented peered customized mechanism (TOPCM) to compute the trust value related to each UAV. This system improves PDR and maintains network reliability. In this study, malicious UAVs that behave abnormally and threaten the routing process in FANET will be eliminated. Malicious nodes exchange fake and false information with other UAVs to weaken throughput and reduce network reliability. For this reason, TOPCM secures the network environment and counteracts these hostile nodes. The experimental results indicate that TOPCM obtains the trust of UAVs accurately and rapidly, identifies hostile nodes, and increases PDR in the network.

In [74], the authors offer a new security strategy for detecting fake routes in MANET. To find malicious nodes due to stealthy attacks, the dynamic malicious node detection algorithm (MNDA) is proposed. It gets information about intermediate nodes and stores them in a special table. According to this table, hostile nodes are identified, and alternative paths are selected. The evaluation outcomes indicate the efficiency of MNDA compared to the Bayesian linear model-based detection strategy.

In [75], a fuzzy secure routing method called FTSR is proposed for FANETs. It focuses on four cyberattacks, namely SF, black hole (BH), WH, and flooding attack (FA), and tries to detect and separate these adversary nodes. FTSR uses two security mechanisms, local trust and route trust. In the first security mechanism, local trust, each UAV attempts to find and separate hostile nodes from its neighboring nodes locally. Local trust is dependent on BH and SF-based, WH-based, and FA-based local trust values. Hence, reliable nodes participate in the routing phase. However, some hostile nodes may be hidden and, consequently, the local trust mechanism cannot detect them. In the routing phase, a new criterion called route trust is calculated to find trustable paths. Route trust is derived from a fuzzy system with three inputs, namely BH and SF-based, WH-based, and FA-based path trust values.

In [76], a human immune system-based security mechanism called SUAS-HIS is suggested for FANETs. This scheme neutralizes four cyberattacks, namely WH, BH, SF, and fake information dissemination. In SUAS-HIS, each antibody uses three scales, namely round trip time, signal strength intensity (SSI), and the number of backward packets from the destination to limit these attacks, and antigens represent a set of all formed paths in the network. In this step, unsafe paths are identified and eliminated through SUAS-HIS, and the safest routes are applied for transferring data packets. Extensive tests confirm the speed and accuracy of SUAS-HIS to detect hostile nodes on the network. Additionally, SUAS-HIS not only detects the aforementioned attacks well but also isolates these hostile UAVs correctly.

In [77], an self-protective routing scheme called ASP-UAVN in FANETs. Initially, AODV is used to find routes between UAVs in the network. Then, ASP-UAVN neutralizes SF, WH, and SH attacks in the routing process. In this approach, a HIS-based security mechanism is used to ensure network security. This mechanism consists of three agents, namely evaluation agent, decision-making agent, and defense agent. The evaluation agent inspects drones in the route discovery process and records the relevant information in an evaluation table. This agent determines the likelihood of the hostility of UAVs in the relevant path. If this path includes hostile UAVs, it should not be used for transferring data packets. Then, the decision-making agent employs a knowledge base to find suspicious paths based on delay, PDR, PLR, and packet sending rate (PSR). Then, the defense agent applies test packets to analyze these suspicious paths and isolate hostile paths. Lastly, the rest of these paths calculate a threshold value based on round trip time and signal strength to find the safest routes in the network.

Based on the related works, it can be said that the number of solutions presented to detect wormhole attacks in FANETs is very low. Some security approaches such as [71], [72], [75], [76], and [77] have used artificial intelligence and machine learning techniques to combat wormhole attacks because these techniques have a great ability to solve problems related to the identification and isolation of wormholes in FANETs. Therefore, the purpose of this paper is to fill in the existing research gaps in this field by using ML techniques, especially reinforcement learning. QSR is a novel Q-learning secure routing method for FANETs. It focuses on wormhole attacks. The neighbor discovery process in QSR is to find local network topology. A local monitoring system guarantees the security of this process. Moreover, QSR employs a Q-learning-based secure routing phase, which analyzes the behavior of wormholes and its effect on important scales such as the average one-hop delay, hop count, packet loss ratio (PLR), packet transmission frequency (PTF), and packet reception frequency (PRF). Table 1 lists the related works and highlights their most important features.

3. Basic concepts

In this section, the principles and concepts related to reinforcement learning (RL) are explained briefly. RL checks the behavior of a virtual agent (such as software or a robot) and rewards or penalizes its behaviors constantly. Accordingly, the agent acquires knowledge of the environment. The agent chooses one of the possible actions and gets a reward for this specific action. Note that the agent tests various actions in different patterns to get experiences. Usually, the framework of reinforcement learning is described as a Markov decision process (MDP) with tuple $(S, A, P, R, \alpha, \gamma)$, where S includes the possible states, $A\left(s_t\right)$ contains the actions related to each state s_t, P denotes the transmission probability, and $R\left(s_t, a_t\right)$ indicates a reward function, which determines the reward value related to the chosen action. α and γ are limited to $[0, 1]$ and show the learning rate and the discount factor, respectively. In addition, π_t denotes the policy, which shows the behavior of the learning agent at different times. The purpose of the agent is to achieve the optimal policy π^*. $V\left(s_t, a_t\right)$ is the value function, which calculates the sum of the expected reward when choosing the action a_t in the state s_t. This function trains the agent to learn the best policy. RL is suitable for solving distributed issues, especially routing in FANETs. This technique consumes high storage space and needs high computing power, which is dependent on the volume of the state space. The RL-based routing algorithm requires a suitable time to converge an optimal solution (i.e. convergence rate is relatively slow), but its implementation is easy. These methods are very flexible for dynamic topologies and can achieve optimal results [86,87].

A well-known RL strategy is Q-learning, which operates without any environment model and learns the value function (also called Q-value) online. Q-values are refreshed based on Equation (1):

Table 1
Comparison of the related works.

Scheme	Security mechanism	Desired routing attacks	Detection or isolation of attackers	Network	Routing technique	Routing or security scales	Strengths	Weaknesses
TSPO-based DRN [71]	A DRN-based security system	Selective forwarding, sinkhole, wormhole	Detection and isolation	FANET	TSPO-based routing scheme	Round trip time, signal strength, packet size, number of received packets, number of sent packets distance and link quality	High convergence speed, ability to detect various attacks	High routing overhead, not considering energy scale
TBCS [72]	A fuzzy security system	Wormhole, sinkhole, bad mouthing, selective forwarding	Detection and isolation	FANET	A cluster based routing protocol	Signal strength, energy, packet delivery rate, past interactions, transmission delay	Ability to detect several attacks, using a hierarchical topology, high scalability	High energy consumption
TOPCM [73]	Trust-based security mechanism	Not mentioned	Detection and isolation	FANET	AODV	Monitoring data packets and extracting broadcast identifier, destination address, next-hop identifier, current hop identifier	High malicious detection accuracy, high data delivery ratio	High routing overhead, high delay in the routing process
MNDA [74]	A dynamic malicious node detection mechanism	Not mentioned	Detection	MANET	DSR	Waited time to reach next packet	Avoiding packet collision, ability to detect routing attacks	Low detection accuracy, high latency, not suitable for FANET
FTSR [75]	A local trust strategy and a fuzzy-based path trust system	Selective forwarding, blackhole, wormhole, flooding	Detection and isolation	FANET	AODV	Introducing three new local scales, namely WH-based local trust, FA-local trust, BH and SF local trust and defining three new path scales, namely WH-based path trust, FA-path trust, BH and SF path trust	Ability to detect and isolate malicious nodes, considering energy efficiency, high malicious node detection rate, high detection accuracy	High latency
SUAS-HIS [76]	A HIS-based security system	Wormhole, blackhole, grayhole, fake information dissemination	Detection and isolation	FANET	AODV	Round trip time, signal strength intensity (SSI), number of backward packets	Fast malicious detection ability, detecting several attacks	Not regarding energy in the routing procedure
ASP-UAVN [77]	A HIS-based security system	Wormhole, sinkhole, selective forwarding	Detection and isolation	FANET	AODV	Delay, packet delivery rate, packet loss rate, round trip time, signal strength	High detection accuracy, ability to detect and isolate various malicious nodes	Not attention to energy, long delay in the routing process
Scheme	Security mechanism	Desired routing attacks	Detection or isolation of attackers	Network	Routing technique	Routing or security scales	Strengths	Weaknesses
GPSR+ [78]	×	×	×	FANET	An improved GPSR scheme	Distance to the destination node and remaining energy	Presenting a position forecast strategy without additional overhead, considering energy efficiency in the routing process, considering a dynamic hello broadcast time	Long delay in the routing process
GPHLOR [79]	×	×	×	FANET	A geographic location-based opportunistic routing protocol	Location information	Combining two hopless and position-based methods, compatible with topology changes in FANET, lowering routing overhead	Not regarding energy consumption in the routing procedure

(*continued on next page*)

Table 1 (*continued*)

Scheme	Security mechanism	Desired routing attacks	Detection or isolation of attackers	Network	Routing technique	Routing or security scales	Strengths	Weaknesses
QFAN [80]	×	×	×	FANET	A Q-learning-based routing method	Movement information, remaining energy, received signal strength indication (RSSI) information, distance, delay, hop count	Limiting the search space, high convergence rate, low latency, enhancing the data transfer rate, high scalability	High routing overhead, constant learning parameters
GW-COOP [81]	×	×	×	FANET	A GWO-based routing approach	Remaining energy, location information, distance to the desired node	Using a cooperative diversity technique, insuring reliable communication between flying nodes, considering energy efficiency in the network	High time complexity
SF-GoeR [82]	×	×	×	FANET	An improved GPSR scheme	Remaining energy and closeness ratio	Considering energy consumption of UAVs	Non-reliability of communication links
RARP [83]	×	×	×	FANET	An improved AODV scheme	Location information, velocity, movement angle, connection time	Combining directional and Omni-directional strategies, improving path reliability, reducing packet collision	Not attention to energy consumption in the routing process
QTAR [84]	×	×	×	FANET	A Q-learning-based routing protocol	Using information of two-hop neighbors such as residual energy, location, speed, and delay	Adaptability to FANET, low latency, adjusting learning parameters dynamically	High time complexity and slow convergence speed
Q-FANET [85]	×	×	×	FANET	A Q-learning-based routing method	Signal-to-interference-plus-noise ratio (SINR) and the speed of nodes	Presenting a dis-centralized routing scheme, improving delay and PDR in the routing procedure, preventing routing holes using the penalty mechanism	High routing overhead and using fixed learning parameters
QSR	Local monitoring system and a Q-learning-based secure routing process	Wormholes	Detection and isolation	FANET	A Q-learning-based routing protocol	Average one-hop delay, hop count, packet loss ratio, packet transmission frequency (PTF), packet reception frequency (PRF)	Adaptability to FANET, detecting malicious nodes using two local and Q-leaning security systems, detecting and isolating malicious nodes, adjusting learning parameters dynamically	Long delay in the routing process

$$Q\left(s_{t+1},a_t\right)=Q\left(s_t,a_t\right)+\alpha\left[r_{t+1}+\gamma\max_a Q\left(s_{t+1},a_t\right)-Q\left(s_t,a_t\right)\right] \quad (1)$$

where $Q\left(s_t,a_t\right)$ represents the value of the current state s_t when choosing the action a_t. In Q-learning, when the agent's state is s_t, it considers an action such as a_t. Then, the agent repetitively tunes its action election mechanism through the award calculated by the environment. The agent tries to maximize the sum of the expected reward in these iterations and find the highest Q-value in the next state s_{t+1}. After taking the action a_t in each iteration, Q-value must be updated. α (i.e. the learning rate) tunes the learning speed of Q-learning (so that $0<\alpha<1$) to achieve better convergence and more stability. It determines the effect of new and old information on the learning process. γ (i.e. the discount factor) controls the importance of future rewards. If $\gamma \ll 1$, the discount factor prioritizes the immediate rewards. However, if $\gamma \gg 0$, the discount factor prioritizes the rewards obtained from previous experiences. Q-learning is widely used for designing networking protocols, such as routing in FANET, in wireless ad hoc networks because it is easily implemented and makes a good balance between memory and energy consumption. Therefore, the proposed method uses Q-learning to find the safest routes in the network [88,89].

4. System model

Here illustrates the presuppositions of the network and threat models in QSR.

4.1. Network model

In the network, there are a number of UAVs, which are responsible for monitoring and sensing the desired area. The set of UAVs is $USET=\left\{U_1,U_2,...,U_i,...,U_N\right\}$, where N indicates the number of UAVs. U_i (i.e. i-th UAV), is equipped with sensors, cameras, a communication interface, and GPS. U_i covers an area with a constant communication radius r. Moreover, the position of U_i is expressed as $\left(x_i,y_i,z_i\right)$ obtained from GPS. In QSR, several UAVs are directly connected to the ground control station (GCS), and communications between other UAVs

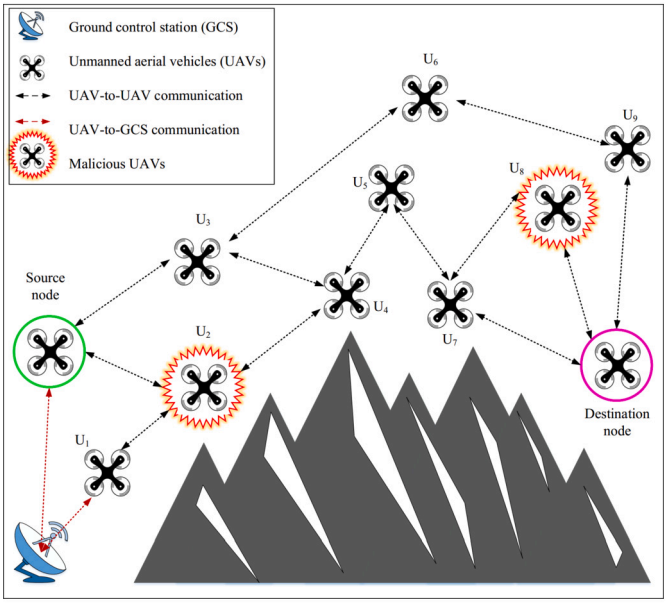


Fig. 1. Network model in QSR.

and GCS are provided using a multi-hop strategy. In this network, UAVs may be honest or hostile.

- **Honest UAVs:** These nodes behave normally and send the right data to other UAVs.
- **Hostile UAVs:** These nodes behave abnormally and send incorrect data to other UAVs. Their purpose is to reduce network efficiency and disable it.

See this network model in Fig. 1. In QSR, all UAVs support IEEE.802.11a as a communication standard. This communication standard guarantees a suitable bandwidth and provides a good performance in dynamic topologies [90]. In QSR, air-to-air (A2A) and air-to-ground (A2G) channels are used in UAV-to-UAV communication and UAV-to-GCS communication, respectively [91]. Air-to-air channels usually have better quality than air-to-ground channels because the path loss in the A2A channel is achieved according to the free-space propagation model, which is less faded, while A2G channels are greatly faded because objects are close to GCS. Also, the path loss in the A2G channel is more severe [92]. However, it is assumed that all channels are associated with path loss and fading. In QSR, the path loss in A2A channels is calculated based on Equation (2):

$$L_{AA}\left(d_{i,j}\right) = \alpha_1 10\log_{10} d_{i,j} + \eta_1 \tag{2}$$

here α_1 and η_1 are the path loss exponent and the path loss at the reference point. Furthermore, $d_{i,j}$ is the distance between U_i and U_j.

$$d_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \tag{3}$$

where $\left(x_i, y_i, z_i\right)$ and $\left(x_j, y_j, z_j\right)$ are the locations of U_i and U_j, respectively.

Furthermore, according to the Friis equation, in the free-space model, $\alpha_1 = 2$ and $\eta_1 = 10\log_{10}\left(\frac{4\pi f}{c}\right)^2$ so that f is the carrier frequency and $C = 3 \times 10^8 m/s$ is the speed of light. Likewise, the path loss in the A2G channel is achieved based on Equation (4).

$$L_{AG}\left(d_{i,j}\right) = \alpha_2 10\log_{10} d_{i,j} + \eta_2 \tag{4}$$

so α_2, $d_{i,j}$, and η_2 represent the path loss exponent, distance, and the path loss at the reference point of the A2G channel, respectively.

4.2. Energy model

In QSR, the first-order radio model [93] is applied to compute the energy consumption of UAVs in the signal transmission procedure. It contains two propagation models, free-space and multi-path fading. In this regard, assume that U_i transfers a data packet with a size of l bits to U_j. Hence, the energy usage of the sender and that of the receiver are computed according to Equations (5) and (6), respectively.

$$E_{tx}\left(l, d_{i,j}\right) = \begin{cases} l \times E_{elec} + l \times \varepsilon_{fs} \times {d_{i,j}}^2 & d_{i,j} \le d_t \\ l \times E_{elec} + l \times \varepsilon_{mp} \times {d_{i,j}}^4 & d_{i,j} > d_t \end{cases} \tag{5}$$

$$E_{rx}(l) = l \times E_{elec} \tag{6}$$

so E_{elec} represents the energy used in the electrical equipment of the receiver/transmitter to receive/send a bit. $\varepsilon_{fs}\left(d_{i,j}\right)^2$ represents the energy used in the free space amplifier and $\varepsilon_{mp}\left(d_{i,j}\right)^4$ denote the energy used in the multi-path fading amplifier. d_t is the distance threshold so that $d_t = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$. Furthermore, $d_{i,j}$ indicates the distance between U_i and U_j, which is obtained from Equation (3).

Furthermore, in QSR, the technique proposed in [94] is considered to obtain the energy used in UAVs when flying and hovering in the sky. Accordingly, each UAV calculates its hovering power based on Equation (7):

$$P_H = \sqrt{\frac{\left(m_u g\right)^3}{2\pi r_\omega n_\omega \rho_a}} \tag{7}$$

where m_u, g, r_ω, and ρ_a denote the mass of UAVs, the gravitational acceleration, wing radius, the number of wings, and the air density, respectively.

Additionally, each UAV calculates its flying power using Equation (8).

$$P_F = \left(P_{\max} - P_H\right)\frac{V_u(t)}{V_{\max}} \tag{8}$$

so that $V_{\max}$, $V_u(t)$, and $P_{\max}$ are the maximum speed, the current speed, and the maximum power of the UAV.

Then, the UAV calculates its energy consumption when hovering and flying in the sky based on Equations (9) and (10), respectively.

$$E_H = P_H T_H \tag{9}$$

$$E_F = \int_0^{T_F} P_F dt \tag{10}$$

here T_H and T_F are two hovering and flight times, respectively.

4.3. Threat model

Due to the features of FANETs, including the lack of infrastructure, wireless communication channels, dynamic topology, and unreliable communications between UAVs, routing protocols in these networks are vulnerable to cybersecurity attacks. In FANET, hostile UAVs penetrate the network and design various attacks to disrupt the data transfer procedure. Hence, it is essential to provide defensive strategies to counteract these attacks in FANET. This paper focuses on wormhole attacks. ***"Wormhole"*** is an important attack on the routing protocols. Usually, a wormhole attack includes two hostile UAVs and a wormhole tunnel. To carry out the attack, the adversaries form a direct link called the wormhole tunnel between themselves. The tunnel can be formed through a wired link, an out-of-band link, legal links, and packet encapsulation [95,96]. After building the wormhole tunnel, the malicious UAV receives the packets from its neighbors, copies them, and sends the duplicated packets through the tunnel to the second malicious UAV. The second hostile UAV broadcasts these duplicated packets to its neighbors.

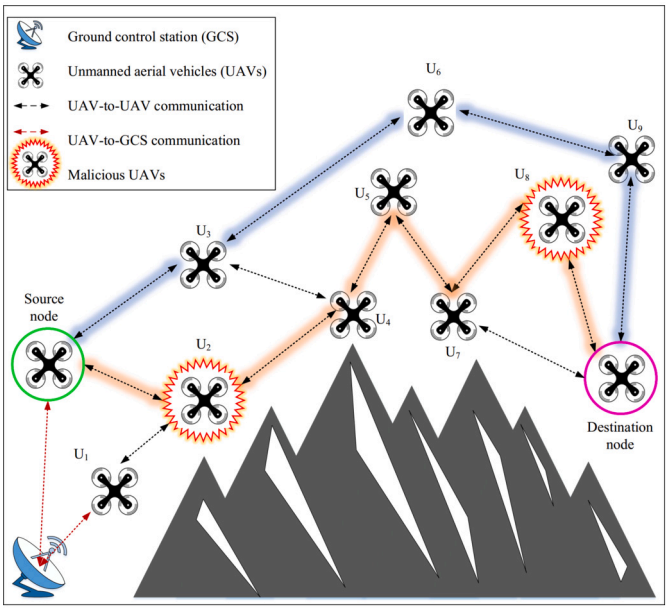


Fig. 2. Wormhole attack through encapsulation.

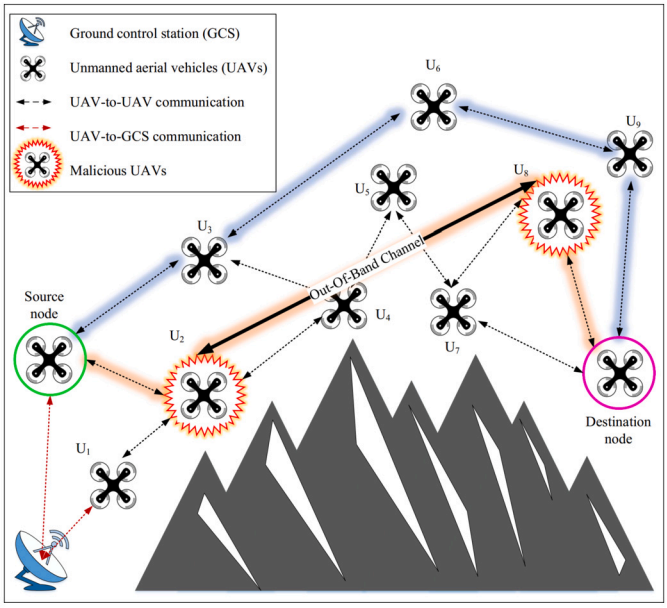


Fig. 3. Wormhole attack through the out-of-band channel.

The attack is divided into four categories with regard to the technique used to launch:

- **Wormhole through encapsulation:** In Fig. 2, source and destination UAVs are trying to get the shortest route between themselves, and there are two hostile UAVs (i.e. U_2 and U_8) in the network. The source UAV disseminates a route request in the network. Then, the malicious node U_2 receives and encapsulates the message and sends the encapsulated packet to U_8 through the wormhole tunnel between the hostile nodes (i.e. $U_4 - U_5 - U_7$). Then, U_8 extracts RREQ from the encapsulated packet and disseminates it until RREQ reaches the destination UAV. Note that if RREQ is encapsulated, hops in the path do not increase. At the same time, the source UAV sends RREQ to the destination UAV through a legal path (i.e. $U_3 - U_6 - U_9$). Now, there are two paths between the source and destination UAVs:
  - ***Route 1 (4 hops):*** $Source - U_3 - U_6 - U_9 - Destination$
  - ***Route 2 (apparently 3 hops):*** $Source - U_2 - U_8 - Destination$

  The destination UAV selects Route 2 because it is the shortest route apparently. However, in fact, this illegal path has six hops. If a routing approach considers the number of hops to select the best path, it is vulnerable to the attack. This type of attack is easily launched because it does not need specific equipment. However, in a wormhole with packet encapsulation, data transmission is almost slow.

- **Wormhole through the out-of-band channel:** In this attack, an out-of-band channel with high bandwidth is formed between hostile UAVs (i.e. U_2 and U_8). In this wormhole attack, the attackers are directly connected to each other. This fastens data transmission, reduces the number of hops, and consequently, creates the view that the two hostile UAVs are very close to each other. The implementation of this attack is more difficult than the wormhole attack through encapsulation because malicious UAVs need specific hardware. See Fig. 3. In this figure, if the source UAV disseminates a RREQ toward the destination UAV, and hostile UAVs connect to each other via the out-of-band channel. U_2 tunnels RREQ to U_8 through this out-of-band channel. U_8 is near the destination and consequently, sends RREQ to the destination directly. In this case, there are two routes in the network:
  - ***Route 1 (4 hops):*** $Source - U_3 - U_6 - U_9 - Destination$
  - ***Route 2 (3 hops):*** $Source - U_2 - U_8 - Destination$

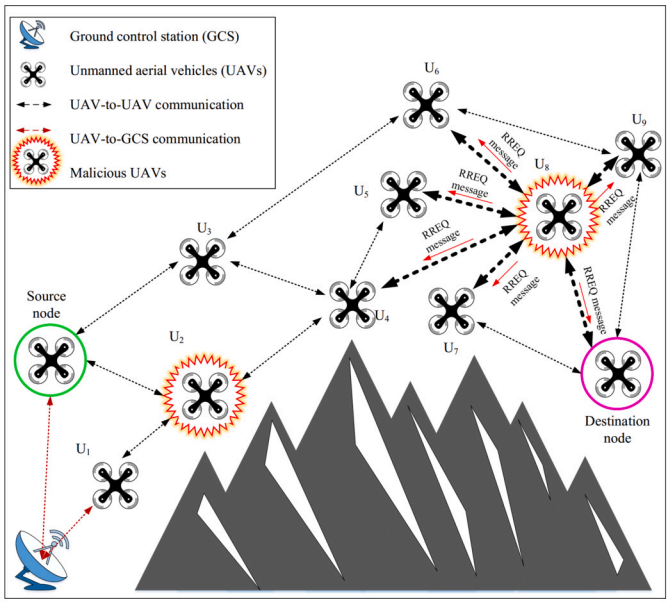


Fig. 4. Wormhole attack with high transmission power.

Route 2 is shorter and faster, so the destination UAV selects it as the best route.

- **Wormhole with high transmission power:** This attack can be carried out by a hostile UAV. This attack is shown in Fig. 4. After getting RREQ from the previous-hop node, the hostile node U_8 uses a high transmission power to broadcast this RREQ because this UAV is equipped with a more powerful energy source. This ability is beyond the capacity of normal UAVs in FANET. Thus, the hostile UAV increases its chances to create paths between the source and destination UAVs to carry out its sabotage operations on the network.
- **Wormhole through packet relay:** In this case, hostile UAVs (i.e. U_2 and U_8) act as a simple receiver/transmitter and relay packets between the two distant UAVs to create a virtual link between them. In this case, the distant UAVs mistakenly imagine that they are neighbors. Only one hostile node can perform this attack. However, when there are many malicious UAVs, this attack increases the neighbor list of the desired UAV. For example, suppose that U_1

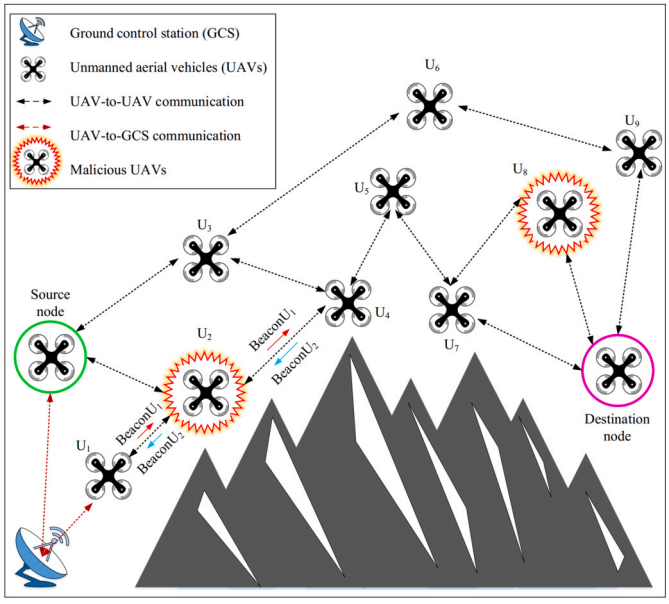


Fig. 5. Wormhole attack through packet relay.

Table 2
Abbreviations used in this paper.

Abbreviations	Description
FANET	Flying ad hoc network
UAV	Unmanned aerial vehicles
QoS	Quality of service
PTF	Packet transmission frequency
PRF	Packet reception frequency
PLR	Packet loss ratio
QSR	Q-learning-based secure routing scheme
MANET	Mobile ad hoc network
VANET	Vehicular ad hoc network
RL	Reinforcement learning
AI	Artificial intelligence
ML	Machine learning
IoD	Internet of Drones
SF	Selective forwarding attack
DoS	Denial of service attack
SH	Sinkhole attack
WH	wormhole attack
BH	Black hole attack
FA	Flooding attack
PO	Political Optimizer
TSA	Tunicate Swarm Algorithm
DRN	Deep residual network
GCS	Ground control station
SSI	Signal strength intensity
GWO	Grey wolf optimization algorithm
SNR	Signal-to-noise ratio
WBSN	Wireless body sensor network
RREQ	Route request message
RREP	Route reply message
MDP	Markov decision process
GCS	Ground control station
GPS	Global positioning system
RREQ	Route request message
RREP	Route reply message

and U_4 are not neighbors, and a malicious UAV (i.e. U_2) is located between them. In this attack, U_2 can tunnel all beacon messages from U_1 to U_4 and vice versa. In this case, U_1 and U_4 conclude that they are neighbors, and they may choose each other as the next-hop in the routing process. See Fig. 5.

The proposed method analyzes the effect of wormhole attacks on network performance. This helps researchers understand the attack and how to deal with it. In this paper, it is assumed that hostile UAVs use similar hardware equipment to other UAVs in the network. Therefore, QSR focuses only on two wormhole attacks, namely wormhole through encapsulation and wormhole through packet relay. Hence, wormholes through out-of-band channels and wormholes with high transmission power are outside the scope of this paper.

Table 3
Symbols used in QSR.

Symbols	Description
$USET$	A set of UAVs in the network
N	Number of UAVs in the network
U_i	i-th UAV in the network
(x_i, y_i, z_i)	Position of U_i
E_{tx}	Energy consumed in the transmitter
E_{rx}	Energy consumed in the receiver
E_{elec}	Energy used by the electrical equipment of the receiver/transmitter
ε_{fs}	Signal amplifier coefficient in the free space
$d_{i,j}$	Distance between U_i and U_j
$Table_i$	Neighbor table of U_i
N_i	Number of neighboring nodes of U_i in $Table_i$
E_i	Remaining energy of U_i
V_i	Speed of U_i
ID_i	Identifier of U_i
α_{ij}	Learning rate related to the link between U_i and U_j
γ_j	Discount factor related to U_j
Q_j	Q-value corresponding to U_j
$VTime_j$	Validity time of the entry related to U_j in $Table_i$
$AvgDelay_{OneHop}$	Average single-hop delay
$Delay^{ij}_{OneHop}$	Single-hop delay between U_i and U_j
$PDelay^{ij}_{OneHop}$	Propagation delay between U_i and U_j
$MDelay^{ij}_{OneHop}$	Media access delay between U_i and U_j
$QDelay^{j}_{OneHop}$	Queuing delay of U_j
HC_j	Number of hops from source node to U_j
$D_{Beacon}(i,j)$	Number of missing beacon messages in U_j
$S_{Beacon}(i,j)$	Number of beacon messages sent from U_i to U_j
R_t	Reward function
r	Communication range of UAVs

5. Proposed scheme

Here, the Q-learning-based secure routing (QSR) scheme is described in flying ad hoc networks. QSR deals with the wormhole attack because it causes serious damage to the network and provides suitable conditions for other attacks like selective forwarding, traffic analysis, and denial of service (DoS) attacks. In QSR, UAVs use the Q-learning algorithm to decide on the best path. In general, QSR consists of two phases:

- Secure neighbor discovery
- Q-learning-based secure routing

Tables 2 and 3 present the most important abbreviations and symbols used in this paper, respectively. The diagram of the proposed method is also shown in Fig. 6.

5.1. Secure neighbor discovery phase

In the secure neighbor discovery process, each UAV is aware of its neighboring UAVs in the network and establishes a neighbor table. In addition, UAVs monitor each other to locally identify and isolate wormholes. Therefore, this phase consists of two steps: neighbor table establishment and local monitoring system.

5.1.1. Neighbor table establishment

The purpose of this step is that each UAV, like U_i (where $i = 1, 2, ..., N$) knows its nearby UAVs and forms a neighbor table (i.e. $Table_i$). Table 4 presents the format of $Table_i$. Note that the information

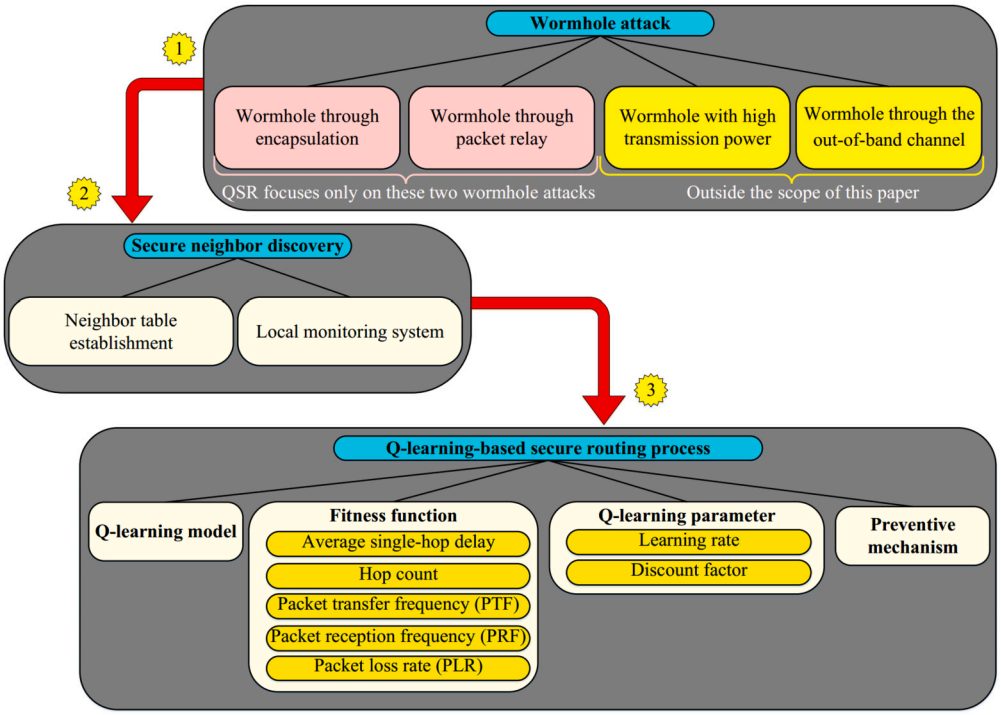


Fig. 6. Diagram of the proposed method.

Table 4
The format of $Table_i$.

Identifier	Information about neighboring nodes				Information about the Q-learning algorithm			Valid time
	Position	Speed	Remaining energy	Queuing delay	Discount factor	Q-value	Learning rate	
ID_j	(x_j, y_j, z_j)	V_j	E_j	$QDelay^j_{OneHop}$	γ_j	Q_j	α_{ij}	$VTime_j$

stored in $Table_i$ is applied in the Q-learning-based secure routing phase. To establish $Table_i$, U_i propagates a beacon message for its surrounding UAVs (for example, U_j) at regular times. It contains an identifier (ID_i), geographic location (x_i, y_i, z_i), remaining energy (E_i), speed (V_i), and queuing delay ($QDelay^j_{OneHop}$). When U_i obtains a beacon packet from U_j for the first time, it adds an entry to $Table_i$ and saves its information. Note that the learning parameters in QSR are refreshed in accordance with network situations. Hence, information related to the learning rate (i.e. α_{ij}) related to the link between U_i and U_j, the discount factor of U_j (i.e. γ_j), and the Q-value of U_j (i.e. Q_j) is recorded in $Table_i$. In Table 4, $VTime_j$ is a time interval, which determines whether the entry related to U_j is valid or not. It is updated after receiving each beacon message from U_j, and its value is directly proportional to the beacon message period. Now, if $VTime_j$ is ended, and U_i does not get any beacon message from U_j, the entry related to U_j will be removed from $Table_i$.

In QSR, the beacon time is adjusted according to the speed of UAVs. They have a reverse relationship with each other so that if UAVs move at high speed, the beacon period will be shorter because network topology experiences high changes in this case. Nevertheless, when UAVs move slowly, the beacon time will be longer because network topology experiences low changes in this case.

5.1.2. Local monitoring system

In the local monitoring system proposed in QSR, each UAV is responsible for monitoring its neighboring UAVs and identifying and isolating hostile nodes locally. This operation is dependent on the relationships between the two neighboring UAVs. According to the definition of the wormhole attack through packet relay in Section 4.3, hostile UAVs can act as simple receivers/transmitters. In this case, the wormhole node is located between the two honest non-neighboring UAVs, and relays the beacon messages received from the two UAVs. Two non-neighboring UAVs feel falsely that they are connected directly to each other. Thus, the state space will be increased. This will greatly reduce the convergence rate of the Q-learning. Therefore, the detection and isolation of wormholes are very important in the secure neighbor discovery phase. Three rules are defined to achieve this goal.

- **Rule 1:** A UAV should not receive its beacon messages from neighboring UAVs. If there is a wormhole UAV in the network, it can relay the packets related to other nodes. In Fig. 7, U_1 disseminates its beacon message to neighboring nodes. The hostile node U_4 receives this message and relays it on the network. U_1 gets this message and checks the identifier inserted in this message (i.e. ID_1). As a result, U_1 finds out that Rule 1 has been violated.
- **Rule 2:** A UAV should not receive the beacon message related to a neighboring UAV from other UAVs. In Fig. 8, U_1 broadcasts a beacon message to its neighbors, including U_2, U_3, U_4, and U_5. U_4 is a hostile node and rebroadcasts the message in the network. U_3 and U_5 check the identifier inserted into this message (i.e. ID_1), they realize that this beacon message is duplicated. As a result, and Rule 2 has been violated.
- **Rule 3:** A UAV receives a beacon message only from its single-hop neighboring UAVs. In Fig. 9, U_1 and U_6 are not neighbors, and the hostile node U_4 is located between the two UAVs. As soon as U_4 receives a beacon message from U_1, it relays the message to U_6, and vice versa. U_4 wants to deceive U_1 and U_6 that they are neighbors. U_1 and U_6 analyze the location information contained in these beacon messages and calculate the distance between themselves. Hence, U_1 and U_6 find that their distance is greater than their com-

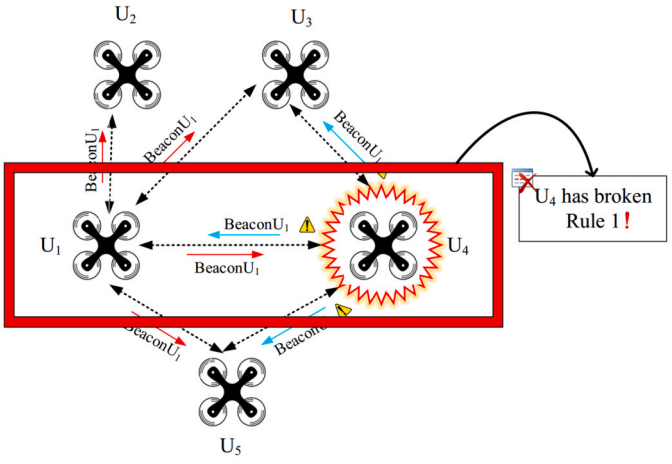


Fig. 7. Violation of Rule 1.

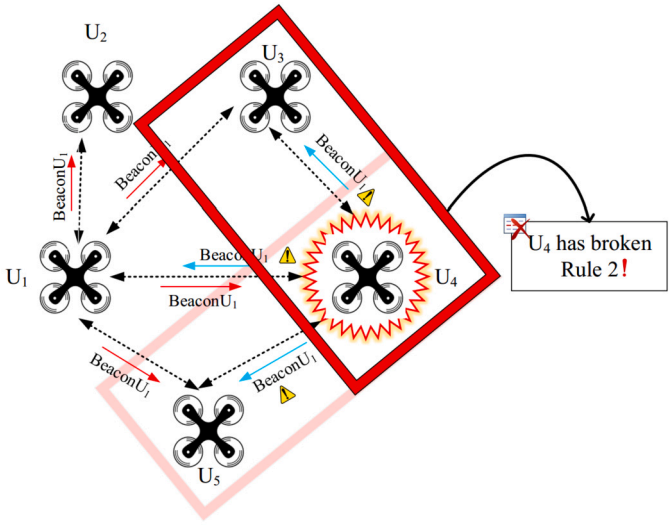


Fig. 8. Violation of Rule 2.

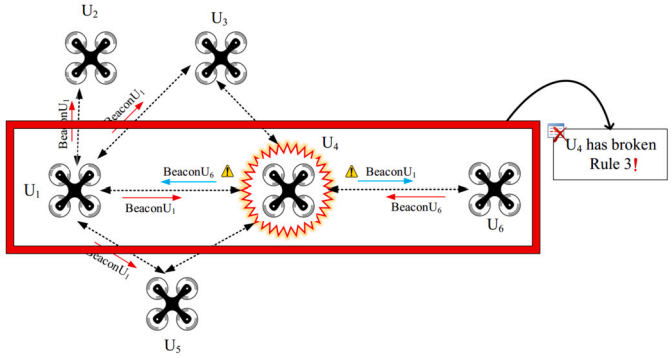


Fig. 9. Violation of Rule 3.

munication radius (see Equation (11)), and the two UAVs cannot be neighbors. In this case, Rule 3 has been violated.

$$\sqrt{(x_1 - x_6)^2 + (y_1 - y_6)^2 + (z_1 - z_6)^2} > r \tag{11}$$

where (x_1, y_1, z_1) and (x_6, y_6, z_6) are the geographic coordinates of U_1 and U_6, r shows the transmission radius of UAVs.

Algorithm 1 Secure neighbor finding phase.

Input: U_i: UAVs in the network so that $i = 1, 2, ..., N$.
T_{Beacon}: Beacon broadcast time.
$T_{network}$: A timer for holding the time spent in the network.
Output: $Table_i$: Neighbor table corresponding to U_i.

```
Begin
1:  Initialize T_network = 0;
2:  repeat
3:    U_i: Set T_Beacon according to UAVs' velocity;
4:    if T_network mod T_Beacon = 0 then
5:      U_i: Broadcast a beacon message for all single-hop neighbors;
6:      if U_i receipts a beacon packet from a new nearby UAV such as U_j and U_j
        violates Rule 1 or Rule 2 or Rule 3 then
7:        U_i: Discard this beacon message;
8:      else if U_i receipts a beacon packet from a new nearby UAV such as U_j and U_j
        does not violate Rule 1 or Rule 2 or Rule 3 then
9:        U_i: Add a new entry to Table_i;
10:       U_i: Inserts the information extracted from this beacon message into Table_i;
11:     end if
12:     if U_i receipts a beacon packet from an old neighbor and U_j violates Rule 1 or
        Rule 2 or Rule 3 then
13:       U_i: Remove U_j from Table_i;
14:     else if U_i receives a beacon message from an old neighbor and U_j does not
        violate Rule 1 or Rule 2 or Rule 3 then
15:       U_i: Refresh the entry related to this neighbor in Table_i;
16:     end if
17:   end if
18:   T_network = T_network + 1;
19: until T_network ≤ Simulation time
    End
```

5.1.3. Description of Algorithm 1

Algorithm 1 indicates the pseudo-code of the secure neighbor discovery phase in QSR. The various steps of this algorithm are explained below.

- In Line 1, a timer is set to zero to measure the simulation duration.
- Line 2 includes a loop *Repeat-Until*. This loop is repeated throughout the simulation period to update $Table_i$ related to U_i continuously.
- In Line 3, U_i tunes its beacon message broadcast time according to the velocity of UAVs.
- Line 4 contains a condition *IF-THEN*. This condition checks whether the beacon broadcast period is or not.
- In Line 5, when this period is reached, U_i broadcasts a beacon message to its neighbors in the network.
- In Line 6, if U_i receipts a fresh beacon packet from a new neighbor, like U_j, and the local monitoring system confirms that U_j violates Rules 1, 2 or 3, then U_i discards the beacon packet received from U_j (see Line 7).
- In Line 8, if U_i receipts a fresh beacon message from a new neighbor, like U_j, and the local monitoring system confirms that U_j does not violate Rules 1, 2 or 3, then U_i adds this new neighbor to $Table_i$ (see Lines 9 and 10).
- If a beacon message is received from an old neighbor such as U_j whose information is available in $Table_i$, and the local monitoring system concludes that U_j violates Rules 1, 2 or 3 (see Line 12), then U_i deletes information about U_j from $Table_i$ (see Line 13).
- If U_i gets a beacon message from an old neighbor such as U_j whose information is available in $Table_i$, and the local monitoring system confirms that U_j does not violate Rules 1, 2 or 3 (see Line 14). The information about U_j in $Table_i$ is updated based on this new beacon message (see Line 15).

Lastly, the time complexity of Algorithm 1 is equal to $O(1)$.

5.2. Q-learning-based secure routing phase

QSR defines the routing process based on a distributed Q-learning algorithm. This framework includes four parts: learning model, reward function, learning factors, and preventative mechanism.

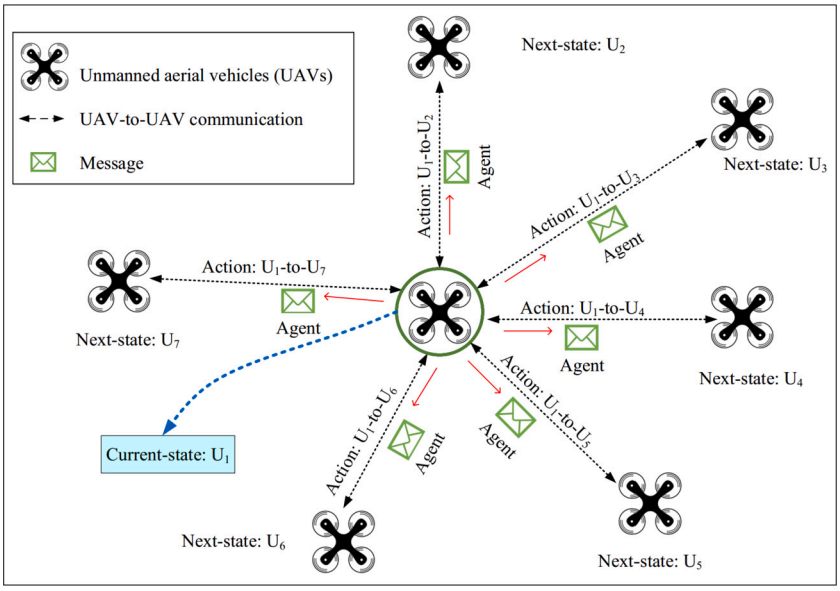


Fig. 10. The routing procedure in QSR.

Table 5
Elements related to the learning model.

Learning elements	Description
Learning issue	Learning the best routing paths in the network
Learning environment	Flying ad hoc network
Learning agent	Each data packet
State space	Set of all UAVs in the network
Action	Sending a data packet from current node to the neighboring UAV

5.2.1. Learning model

In QSR, routing is the learning issue, in which a packet is delivered from the source node to the destination node though multi-hop communications. In this routing procedure, the learning environment is related to the network, and the agent corresponds to each message, respectively. The state space represents UAVs available in the network (i.e. $USET = \{U_1, U_2, ..., U_i, ..., U_N\}$ so N represents the number of UAVs). At each moment, the state of a packet (i.e. agent) demonstrates the current node, which keeps the packet. For example, if U_i is holding a packet, the current state of the agent is U_i. In this routing procedure, the action set presents the transmission operation from the current node to the next node. For example, if the packet in the current state (U_i) is sent to the next-hop (i.e. U_j), the agent carries out the action $U_i - to - U_j$, and its new state is U_j. See this routing process in Fig. 10. Table 5 introduces the elements of this learning model. After selecting the action $U_i - to - U_j$, the packet moves from U_i to U_j. After doing this action, the environment calculates the reward value in accordance with the reward function presented in Section 5.2.2 and gives its return to the agent.

5.2.2. Reward function

In the route learning process, QSR analyzes the behavior of wormholes to propose a new reward function to reward safe routes. Wormholes affect five important factors, namely the average one-hop delay, hop count, packet loss ratio, packet transmission frequency (PTF), and packet reception frequency (PRF). In the following, the effect of wormholes on these factors is explained.

- **Average single-hop delay ($\mathbf{AvgDelay_{OneHop}}$):** In the wormhole attack through encapsulation stated in Section 4.3, if the hostile UAV gets a packet from the source UAV, it encapsulates the packet and sends the encapsulated packet to the next hostile UAV via the wormhole tunnel. This tunnel may include a large number of intermediate UAVs that are hidden in the encapsulation process. As a result, it seems that the wormhole path has fewer hops, but it includes high delay when transferring data packets. This is shown in Fig. 11. According to Fig. 11 (a), the single-hop delay between the two wormhole nodes (i.e. U_2 and U_5) equalizes the sum of the single-hop delays between all the hidden nodes in the wormhole tunnel. It is equal to $Delay^{25}_{OneHop} = Delay^{23}_{OneHop} + Delay^{34}_{OneHop} + Delay^{45}_{OneHop}$. Now, the delay related to the unsafe path (i.e. $Delay^{Unsafe}_{Route}$) is expressed according to Equation (12):

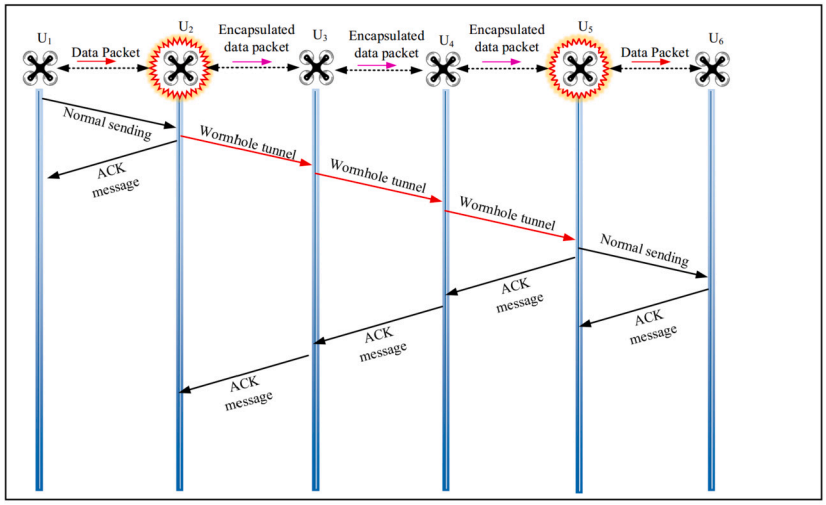
$$Delay^{Unsafe}_{Route} = Delay^{12}_{OneHop} + \underbrace{Delay^{25}_{OneHop}}_{Delay^{23}_{OneHop}+Delay^{34}_{OneHop}+Delay^{45}_{OneHop}} + Delay^{56}_{OneHop} \tag{12}$$

where $Delay^{12}_{OneHop}$ indicates the single-hop delay between U_1 and U_2.

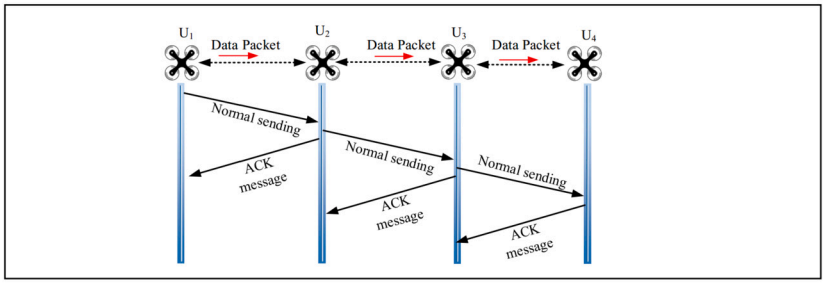
In contrast, a safe path that has similar hops to the unsafe path, experiences less delay. According to Fig. 11 (b), the single-hop delay between the two honest UAVs is much less than the delay between the two wormhole nodes. Hence, the delay related to the safe path (i.e. $Delay^{Safe}_{Route}$) is obtained from Equation (13).

$$Delay^{Safe}_{Route} = Delay^{12}_{OneHop} + Delay^{23}_{OneHop} + Delay^{34}_{OneHop} \tag{13}$$

Therefore, delay is a very good metric for finding unsafe paths. According to the mentioned points, the single-hop delay between U_i and U_j (i.e. $Delay^{ij}_{OneHop}$) is calculated based on Equation (14). It is dependent on four criteria, including propagation delay ($PDelay^{ij}_{OneHop}$), media access delay ($MDelay^{ij}_{OneHop}$), and queuing delay ($QDelay^{j}_{OneHop}$).



(a) An unsafe route



(b) A safe route

Fig. 11. Comparison of unsafe and safe paths in terms of delay.

$$Delay^{ij}_{OneHop} = PDelay^{ij}_{OneHop} + MDelay^{ij}_{OneHop} + QDelay^{j}_{OneHop} \tag{14}$$

where $PDelay^{ij}_{OneHop}$ is obtained from Equation (15). It is directly associated with the distance between U_i and U_j, and has a reverse relationship with the media velocity (v_{media}), which is equal to the light speed.

$$PDelay^{ij}_{OneHop} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{v_{media}} \tag{15}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) represent the location coordinates of U_i and U_j, $v_{media} = 3 \times 10^8\ m/s$.

$MDelay^{ij}_{OneHop}$ is the difference between the sending time of the packet sent by U_i (i.e. $Time^{i}_{Sending}$) and the receiving time of ACK received from U_j (i.e. $Time^{j}_{ACK}$).

$$MDelay^{ij}_{OneHop} = Time^{j}_{ACK} - Time^{i}_{Sending} \tag{16}$$

Finally, $QDelay^{j}_{OneHop}$ is extracted from the neighbor table. It is calculated based on the occupied buffer space of UAVs and indicates the time required to reach the beginning of the buffer queue.

Now, U_i normalizes $Delay^{ij}_{OneHop}$ based on Equation (17).

$$Delay^{ij}_{norm} = \frac{Delay^{ij}_{OneHop} - MinDelay_{OneHop}}{MaxDelay_{OneHop} - MinDelay_{OneHop}} \tag{17}$$

$MaxDelay_{OneHop}$ and $MinDelay_{OneHop}$ indicate the upper and lower boundaries of the single-hop delay. So that $MinDelay_{OneHop}$ is equal to the least single-hop delay between U_i and its neighbors i.e. $MinDelay_{OneHop} = \min_{U_k \in Table_i} \left\{ Delay^{ik}_{OneHop} \right\}$, and $MaxDelay_{OneHop}$ indicates the maximum single-hop delay between two neighboring UAVs. If $Delay^{ij}_{OneHop} > MaxDelay_{OneHop}$, then the relevant path includes wormholes probably. In this case, $MaxDelay_{OneHop}$ is obtained from Equation (18).

$$MaxDelay_{OneHop} = MaxPDelay_{OneHop} + MaxQDelay_{OneHop} + AvgMDelay_{OneHop} \tag{18}$$

so that $MaxPDelay_{OneHop}$ indicates the maximum propagation delay. To calculate this parameter, it is assumed that the distance between U_i and U_j is equal to their communication radius (r).

$$MaxPDelay_{OneHop} = \frac{r}{v_{media}} \tag{19}$$

Also, $MaxQDelay_{OneHop}$ is the maximum queuing delay extracted from the neighbor table. Lastly, $AvgMDelay_{OneHop}$ represents the average media access delay, which is calculated based on Equation (20).

$$AvgMDelay_{OneHop} = \frac{\sum_{j \in Table_i}^{N_i} MDelay^{ij}_{OneHop}}{N_i} \tag{20}$$

so, N_i represents the number of neighbors related to U_i.

- **Hop count (HC_j):** In the wormhole attack through encapsulation, when the hostile UAV receives the packet, it encapsulates the packet and sends it to the next attacker through the wormhole tunnel. In the encapsulation process, the number of hops is not counted in the wormhole tunnel, and the two attackers claim to have a very short route to the destination. In [49], the authors have proven that the number of hops in a safe path meets the condition $\left\lfloor \frac{d_{S,D}}{r} \right\rfloor \leq HC \leq \left\lceil 2\frac{d_{S,D}}{r} \right\rceil$ so that $d_{S,D} = \sqrt{(x_S - x_D)^2 + (y_S - y_D)^2 + (z_S - z_D)^2}$

is the distance between the source and destination nodes, and r expresses the communication radius. Also, (x_S, y_S, z_S) and (x_D, y_D, z_D) describe the locations of the source and destination nodes, respectively. Now, if the hop count is less than $\left\lfloor \frac{d_{S,D}}{r} \right\rfloor$, the path includes wormhole probably, and if the hop count is more than $\left\lceil 2\frac{d_{S,D}}{r} \right\rceil$, the path may include routing loops. HC_j indicates the number of hops from the source node to U_j. This parameter is recorded in data packets. According to the mentioned points, the distance between the source UAV and U_j is equal to $d_{s,j} = \sqrt{(x_s - x_j)^2 + (y_s - y_j)^2 + (z_s - z_j)^2}$. As a result, the number of hops should be limited to $\left\lfloor \frac{d_{S,j}}{r} \right\rfloor \le HC_j \le \left\lceil 2\frac{d_{S,j}}{r} \right\rceil$. Therefore, HC_j is normalized through Equation (21).

$$HC^j_{norm} = \begin{cases} \frac{HC_j}{\left\lfloor \frac{d_{S,j}}{r} \right\rfloor}, & HC_j < \left\lfloor \frac{d_{S,j}}{r} \right\rfloor \\ 1, & \left\lfloor \frac{d_{S,j}}{r} \right\rfloor \le HC_j \le \left\lceil 2\frac{d_{S,j}}{r} \right\rceil \\ 1 - \left(\frac{HC_j - \left\lfloor 2\frac{d_{S,j}}{r} \right\rfloor}{(N-1) - \left\lfloor 2\frac{d_{S,j}}{r} \right\rfloor} \right), & \left\lceil 2\frac{d_{S,j}}{r} \right\rceil < HC_j < N-1 \end{cases} \tag{21}$$

where N is the number of all UAVs in the network.
According to the above equation, if HC_j is limited to $\left\lfloor \frac{d_{S,j}}{r} \right\rfloor \le HC_j \le \left\lceil 2\frac{d_{S,j}}{r} \right\rceil$, HC^j_{norm} is equal to one. However, if $HC_j < \left\lfloor \frac{d_{S,D}}{r} \right\rfloor$ or $HC_j > \left\lceil 2\frac{d_{S,D}}{r} \right\rceil$, then HC^j_{norm} approaches zero.

- **Packet loss rate (PLR):** Usually, hostile UAVs send high traffic to the wormhole tunnel, and consequently, high congestion may cause high packet loss. Hence, packet loss rate is a good metric for finding hostile UAVs. PLR_j is the packet loss ratio related to U_j. It shows the ratio of the packets lost in U_j to all data packets sent from U_i to U_j. UAVs use a counter to enumerate the number of beacon packets receipted from their neighbors. Note that the beacon period is prespecified, and U_j knows the number of beacon messages sent from U_i at a specific timeframe such as $\Phi = [t, t+1]$. PLR_j is calculated through Equation (22).

$$PLR_j = \frac{\sum_{t \in \Phi} Beacon_j^{Dropped}}{\sum_{t \in \Phi} Beacon_j^{Sent}} \tag{22}$$

so that $Beacon_j^{Dropped}$ indicates lost beacon messages in U_j, and $Beacon_j^{Sent}$ is the number of beacon messages sent from U_i to U_j at $\Phi = [t, t+1]$. Now, PLR_j is normalized using Equation (23):

$$PLR^j_{norm} = \frac{PLR_j}{\max\limits_{U_k \in Table_i} \{PLR_k\}} \tag{23}$$

so that $\max\limits_{U_k \in Table_i} \{PLR_k\}$ indicates the maximum PLR related to neighbors of U_i.

- **Packet transfer frequency (PTF):** Since the wormhole nodes usually have high PLR. This increases the retransmission of the packets in the hostile UAV. Therefore, QSR uses PTF for evaluating network nodes and finding wormholes. Equation (24) evaluates the packet transfer frequency related to U_j (i.e. PTF_j) at the time interval Φ.

$$PTF_j = \frac{PK_j^{Transferred}}{\Phi} \tag{24}$$

where $PK_j^{Transferred}$ indicates the total sent packets in the interval Φ. Now, PTF_j is normalized based on Equation (25):

$$PTF^j_{norm} = \frac{PTF_j}{\max\limits_{U_k \in Table_i} \{PTF_k\}} \tag{25}$$

where $\max\limits_{U_k \in Table_i} \{PTF_k\}$ is the maximum PTF related to neighbors of U_i.

- **Packet reception frequency (PRF):** Hostile UAVs try to persuade other network nodes to transmit their packets via the wormhole tunnel. Therefore, the wormholes experience high traffic that will cause data loss and decrease PRF. This metric is used in the reward function to detect and isolate hostile UAVs. PRF related to U_j at the interval Φ is calculated based on Equation (26):

$$PRF_j = \frac{PK_j^{Received}}{\Phi} \tag{26}$$

where $PK_j^{Received}$ shows the total number of the received packets in the interval Φ. In addition, PRF_j uses Equation (27) for the normalization process.

$$PRF^j_{norm} = \frac{PRF_j}{\max\limits_{U_k \in Table_i} \{PRF_k\}} \tag{27}$$

where $\max\limits_{U_k \in Table_i} \{PRF_k\}$ is the maximum PRF related to neighbors of U_i.

Therefore, the reward function is achieved based on Equation (28).

$$R_t = \begin{cases} R_{\max}, & U^{t+1} \; is \; destination \\ R_{\min}, & U^{t+1} \; is \; local \; minimum \\ \omega_1 e^{-\left(Delay^{ij}_{norm}\right)} + \omega_2 \left(1 - e^{-HC^j_{norm}}\right) & \\ \quad + \omega_3 e^{\left(\frac{PRF^j_{norm}}{PLR^j_{norm} + PTF^j_{norm}}\right)}, & Otherwise \end{cases} \tag{28}$$

here, ω_i $(i = 1, 2, 3)$ denotes the weight coefficient and $\sum\limits_{i=1}^{3} \omega_i = 1$. As shown in the reward function, if the selected UAV (U_j) is the destination node, meaning that U_i and the destination are neighbors, then the link between U_i and U_j gets the most reward ($R_{\max}$). Furthermore, when a local optimization happens, there is no U_j that has a shorter distance to the destination in comparison with U_i, thus, the link gets the least reward ($R_{\min}$). Otherwise, the reward function will be specified in accordance with single-hop delay, hop count, PLR, PTF, and PRF.

5.2.3. Q-learning parameters

In QSR, the learning rate and the discount factor are dynamically calculated in accordance with network conditions. According to [89], the single-hop delay ($Delay^{ij}_{OneHop}$) is used to determine the learning rate of the link between U_i and U_j. Based on this definition, if the link is safe and stable, it has low delay. In the first step, $Delay^{ij}_{OneHop}$ is standardized based on Equation (29).

$$Delay^{ij}_{Standard} = \frac{Delay^{ij}_{OneHop} - \mu^{ij}_{Delay}}{\sigma^{ij}_{Delay}} \tag{29}$$

where, μ^{ij}_{Delay} (Equation (30)) and σ^{ij}_{Delay} (Equations (31)) describes the mean and the standard deviation of the single-hop delay of U_i and the neighboring nodes, respectively.

$$\mu^{ij}_{Delay} = \frac{1}{N_i} \sum_{j \in Table_i}^{N_i} Delay^{ij}_{OneHop} \tag{30}$$

$$\sigma^{ij}_{Delay} = \sqrt{\frac{1}{N_i} \sum_{j \in Table_i}^{N_i} \left(Delay^{ij}_{OneHop} - \mu^{ij}_{Delay}\right)^2} \tag{31}$$

In Equations (30) and (31), N_i represents the number of neighbors of U_i, and $Table_i$ is the neighbor set of U_i.

Now, the exponential function presented in Equation (32) adjusts the learning rate between U_i and U_j (α_{ij}).

$$\alpha_{ij} = \begin{cases} 1 - e^{-Delay^{ij}_{Standard}}, & \sigma^{ij}_{Delay} \neq 0 \\ 0.3 & \sigma^{ij}_{Delay} = 0 \end{cases} \tag{32}$$

In this Q-learning algorithm, the discount factor (γ) indicates the importance of the reward gotten from the environment. The high value of γ means that the agent can use previous experiences because of the stability of Q-values. However, when γ is low, the agent should pay more attention to the latest reward gotten from the environment because of the instability of Q-values. In QSR, the purpose of the process is to select a safe and reliable neighboring node. Thus, γ considers the movement of neighbors in the two consecutive periods and the energy change rate of U_i. If the movement of neighbors is fast in two consecutive periods and the energy change rate of U_i is high, it indicates the instability of Q-values and reduces γ. In Equation (33), the energy metric leads to the balance of energy consumed in the network when choosing the next-hop node.

$$\gamma_j = \lambda \left(1 - \frac{|N_i(t-1) \cup N_i(t)| - |N_i(t-1) \cap N_i(t)|}{|N_i(t-1) \cup N_i(t)|}\right) + (1-\lambda)\left(1 - \frac{E_i(t-1) - E_i(t)}{\Delta t}\right) \tag{33}$$

so that $N_i(t-1)$ and $N_i(t)$ indicate the state of neighbors of U_i in $t-1$ and t, respectively. Also, $E_i(t-1)$ and $E_i(t)$ reflects the remaining energy of U_i in $t-1$ and t, respectively.

5.2.4. Preventive mechanism

Events such as wormhole attacks, routing holes, and link failures, disrupt the routing procedure due to the increased latency.

- **Routing hole:** QSR tries to penalize the nodes, which are trapped in routing holes. A routing hole, also called the local optimum issue, occurs when the packet cannot be transmitted to the next-hop UAV. Thus, the current UAV is the closest one to the destination. Fig. 12, shows the local optimum issue. As shown in this figure, U_5 receives the packet from U_2 but it cannot find any adjacent UAV with shorter distance than itself from the destination. Here, U_5 is trapped in a routing hole. In QSR, the preventive mechanism is responsible for preventing routing holes in FANET. Hence, to prevent routing holes, when a flying node, such as U_j, is trapped in a local optimization, it sends a feedback to its previous hop (i.e. U_i). In this case, the reward corresponding to the action $U_i - to - U_j$ is $R_{\min}$. This prevents the selection of U_j as the intermediate node in the future routing path.
- **Link failure:** This issue increases packet loss in the routing process. Thus, QSR tries to penalize the nodes, which may be failed. This issue is detected when a node such as U_i does not receive any ACK message from the next-hop node U_j, U_j may be failed. To prevent the link failures, U_i adjusts the reward corresponding to the action $U_i - to - U_j$ to $R_{\min}$, and the Q-value corresponding to U_j is updated in Q-table.

5.2.5. Description of Algorithm 2

Algorithm 2 illustrates the pseudo-code of this process. The various steps of this algorithm are detailed line-by-line below.

- Line 1 initializes ε in the ε-greedy strategy randomly. Note that this strategy is used to counterbalance the exploration and exploitation phases.
- In Line 2, α and γ are determined in accordance with Equations (32) and (33), respectively.
- In Line 3, the initial values of Q-values are set to zero.
- In Line 4, a loop *Repeat-Until* is intended to count the number of episodes because Q-learning is an episodic algorithm.

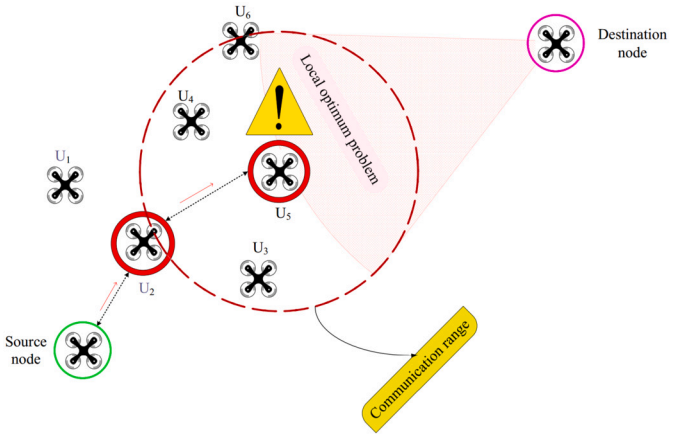


Fig. 12. Local optimum issue.

Algorithm 2 Q-learning-based secure routing structure.

Input: U_i: UAVs in the network so that $i = 1, 2, ..., N$.
ε, α, γ: Learning factors
Output: Routing table: it is corresponding to Q-table stored in each UAV.
Begin
1: Initialize ε randomly;
2: Calculates α and γ based on Equations (32) and (33), respectively;
3: Set Q-values on zero;
4: **repeat**
5: **Agent:** Choose an initial state from $USET = \{U_1, U_2, ..., U_i, ..., U_N\}$ randomly;
6: **for** $t = 1$ to N **do**
7: **Agent:** Get a random number, so that $n_{rand} \in [0, 1]$;
8: **switch** (n_{rand})
9: **case** $n_{rand} \leq \varepsilon$: Choose an action from its action set according to ε-greedy technique;
10: **break**;
11: **case** $n_{rand} > \varepsilon$: Choose an action with maximum Q-value from Q-table;
12: **break**;
13: **end switch**
14: **switch** (U_i^{t+1})
15: **case** destination: $R_t = R_{\max}$;
16: **break**;
17: **case** a local minimum: $R_t = R_{\min}$;
18: **break**;
19: **case** other: $R_t = \omega_1 e^{-\left(Delay^{ij}_{norm}\right)} + \omega_2\left(1 - e^{-HC^j_{norm}}\right) + \omega_3 e^{\left(\frac{PRF^j_{norm}}{PLR^j_{norm} + PTF^j_{norm}}\right)}$;
20: **break**;
21: **end switch**
22: **Agent:** Get its new state from the environment;
23: **Agent:** Set its state on the new state;
24: **Agent:** Refresh Q-value in Q-table according to the reward value;
25: **end for**
26: $episode = episode + 1$;
27: **until** $episode \leq M$
End

- In Line 5, the agent randomly selects its initial state from the possible state set i.e. $USET = \{U_1, U_2, ..., U_i, ..., U_N\}$.
- In Line 6, a loop *For* is repeated to learn the best action in the learning process and update Q-table.
- Inside this loop, the ε-greedy strategy is used to counterbalance between exploration and exploitation. Accordingly, the agent chooses a random number n_{rand} (see Line 7). If $n_{rand} \leq \varepsilon$, then the agent picks out its action from the possible action set randomly (see Line 9) but if $n_{rand} > \varepsilon$, the agent singles out an action with the highest Q-value from actions that have been experienced previously (see Line 11).
- In Line 14, the agent's state is evaluated after taking the selected action.
- In Line 15, if the agent's state shows the position of the destination, the agent receives the most reward, i.e. $R_t = R_{\max}$.
- In Line 17, if the new state is a local optimum, the agent obtains the least award i.e. $R_t = R_{\min}$.

- In other cases, the reward value is calculated based on $R_t = \omega_1 e^{-\left(Delay^{ij}_{norm}\right)} + \omega_2\left(1 - e^{-HC^j_{norm}}\right) + \omega_3 e^{\left(\frac{PRF^j_{norm}}{PLR^j_{norm}+PTF^j_{norm}}\right)}$ (see Line 19).
- In Lines 22 and 23, the agent's state is obtained from the environment.
- In Line 24, the Q-value in Q-table is refreshed based on the obtained reward.

In the end, the time complexity of Algorithm 2 is $O(MN^2)$. so that M expresses the number of episodes, N is the number of UAVs in the network.

6. Security analysis

This section analyzes the proposed solution to tackle and prevent wormhole attacks in FANET. QSR assumes that malicious nodes use hardware similar to other UAVs in the network. Hence, QSR focuses only on two types of wormhole attacks, namely wormhole through packet relay and wormhole through encapsulation.

- **Defensive solution against wormhole through packet relay:** As mentioned in Section 4.3, malicious nodes play the role of simple receivers/transmitters and relay the packets received from other legal UAVs in FANETs. Cooperation and collusion of malicious nodes increase the size of neighboring lists of legal nodes. QSR detects the attack in the secure neighbor discovery phase because UAVs monitor the behavior of each other, and identify and isolate the surrounding wormholes locally. In the secure neighbor discovery phase, nodes exchange beacon messages to identify their neighbors in FANET. An important point is that beacon messages are only broadcast by nodes once, and UAVs can process these messages and do not rebroadcast them again in the network. However, hostile nodes ignore this. Thus, three modes can be considered.
  - **Mode 1:** The malicious UAVs are around a legal node. When a malicious node rebroadcasts the beacon message of this legal node in the network, this legal node receives its beacon message from the malicious node again. In the local monitoring system, Rule 1 can prevent such a condition. As a result, the legal node removes the node that violates Rule 1 from its neighbor table. Hence, this node will be isolated and cannot collaborate in the routing process.
  - **Mode 2:** The malicious node is around two legal neighboring UAVs. When a malicious node rebroadcasts the beacon message received from the second legal node in the network, the first legal node receives this beacon message twice (by its legal neighboring node and the malicious node). In the local monitoring system, Rule 2 prevents such a condition. Thus, the legal node deletes the node that violates Rule 2 from the set of its neighbors. Hence, this node will be isolated and cannot collaborate in the routing process.
  - **Mode 3:** The malicious node is around two legal non-neighboring UAVs. When a malicious node rebroadcasts the beacon message received from the second legal node in the network, the first legal node receives this beacon message from the malicious node. Here, the malicious node deceives the two nodes, and they mistake a neighboring relationship. However, Rule 3 in the local monitoring prevents such a condition. Therefore, the legal node eliminates the node that violates Rule 3 from its neighbor list. Hence, this node will be isolated and cannot collaborate in the routing process.
- **Defensive solution against wormhole through encapsulation:** According to Section 4.3, suppose a legal node wants to identify a suitable path in the presence of hostile nodes. When an adversary captures a packet, it encapsulates this packet and forwards the encapsulated packet to the second adversary through the wormhole tunnel. As a result, the number of hops in this route will be fixed due to the encapsulation process, and the fake route will have fewer hops. Now, if the routing protocol tries to identify the shortest path, it will be vulnerable to the attack. Whereas, QSR detects this attack in the Q-learning-based routing process because the reward function pays attention to the behavior of encapsulation-based wormholes. In general, this function evaluates the negative effect of encapsulation-based wormholes on five important factors, namely the average one-hop delay, hop count, packet loss ratio, packet transmission frequency, and packet reception frequency. The data transfer process between the two legal nodes has less delay than the data transmission process between wormholes because the wormhole tunnel includes a high number of intermediate nodes that are hidden in the encapsulation process. Therefore, evaluating the single-hop delay between the two consecutive UAVs in the route is effective in preventing this attack. On the other hand, in this wormhole attack, the number of hops in the wormhole tunnel is unchanged in the encapsulation process. As a result, the fake route is very short. Hence, to tackle this hostile behavior, QSR accurately determines upper and lower boundaries related to allowed hop counts in one path. Now, if hops in one route is fewer than the lower boundary, this path may include wormholes. Another hostile behavior of malicious UAVs is that wormholes direct high traffic toward the wormhole tunnel. This high congestion can cause some lost packets. Therefore, PLR is a useful scale for detecting and punishing malicious UAVs. Additionally, wormholes usually have a high packet loss rate, which increases data retransmission in FANET. Therefore, QSR uses PTF as a criterion for evaluating nodes and discovering wormholes. On the other hand, a wormhole has high traffic that increases packet loss and reduces PRF. This criterion is used in the reward function to detect and isolate malicious nodes. Therefore, these five factors, namely the average one-hop delay, hop count, PLR, PTF, and PRF in the reward function punish suspicious paths. As a result, UAVs can identify and use secure routes in the data transmission process. Consequently, wormholes cannot negatively affect the routing process.

7. Simulation and result evaluation

The NS2 simulator is used to implement and evaluate QSR. Accordingly, the network has dimensions of $800 \times 800 \times 800\ m^3$. The network environment includes 10-70 nodes that are randomly deployed. The initial energy of UAVs is 100 joules. They move according to the Random Waypoint (RWP) model at a speed of 50 m/s. The communication radius of UAVs is 200 to 300 meters. Each node exchanges 40-300 packets with other nodes in this process, and the runtime is 100 seconds. The constant bit rate (CBR) traffic model is used in the network, and the CBR rate is 1 Mbps. The simulation process assumes that 15% of UAVs are hostile, and the simulation results are obtained in the presence of these hostile UAVs. These parameters are presented in Table 6. Then, the results obtained from this process are presented in the comparison with TOPCM [73], MNRiRIP [73], and MNDA [74] in terms of detection rate, PDR, PLR, accuracy, and delay. The reasons behind selecting these methods are stated below:

- TOPCM is a new secure routing method in FANETs and can well detect hostile nodes.
- QSR, TOPCM, and MNRiRIP can detect and isolate hostile nodes. Hence, they provide a secure network for transferring data packets in the network.
- QSR and MNDA pay attention to the delay parameter in their security systems. As a result, a strong security system is designed to detect wormholes because wormholes usually have a long delay in the data transfer procedure. Additionally, these two methods lo-

Table 6
Simulation scale.

Scale	Value
Simulation tool	NS2
Environment size	$800 \times 800 \times 800$ m^3
Mobility model	Random Waypoint (RWP)
Traffic model	Constant Bite Rate (CBR)
CBR rate	1 Mbps
Mac layer standard	IEEE 802.11a
Antenna	Omni Antenna
Number of UAVs	10-70
Maximum energy of UAVs	100 J
Velocity of UAVs	50 m/s
Number of hostile nodes	15% of total number of UAVs
Communication radius	200 – 300 m
Number of packets	40, 80, 120, 160, 200, 240, 280, 300
Runtime	100 s
Traffic type	TCP
Compared schemes	QSR, TOPCM, MNRiRIP, and MNDA
Evaluation criteria	Detection rate, accuracy, packet delivery rate, packet loss rate, and end-to-end delay

cally monitor the behavior of nodes to identify the hostile nodes in the network.
- QSR, TOPCM, MNRiRIP, and MNDA offer powerful and distributed security systems to identify hostile UAVs.

In the following, the assessment scales are illustrated in summary:

- **Detection rate:** This criterion, also called sensitivity, represents the ability of different schemes to identify and isolate malicious nodes. It determines whether a security method can correctly identify hostile nodes. It is calculated through Equation (34):

$$Sensitivity = \frac{TP}{TP + FN} \tag{34}$$

so that TP represents the true positive and indicates the number of malicious nodes that are correctly detected by the relevant scheme. TN is the true negative and means the number of legal nodes that are properly identified by the relevant method. FP is a false positive and indicates the number of legal nodes that are mistakenly labeled as hostile nodes. Finally, FN denotes the false negative and means the number of hostile nodes that are mistakenly labeled as legal nodes.
- **Accuracy:** This criterion indicates the ratio of the correct detections to all detections. It is obtained through Equation (35):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

- **Packet delivery rate (PDR):** Equation (36) calculates the packet delivery rate and represents the percentage of packets that have been successfully delivered to the destination.

$$PDR = \frac{\sum_{d=1}^{n_d} PK_d}{\sum_{s=1}^{n_s} PK_s} \times 100 \tag{36}$$

where PK_d is d-th delivered packet, and n_d represents the number of delivered packets. In addition, PK_s means s-th sent packet, and n_s indicates the number of sent packets.
- **Packet loss rate (PLR):** Equation (37) is used to calculate data loss ratio. It means the percentage of packets that have not reached the destination node.

$$PLR = \frac{\sum_{l=1}^{n_l} PK_l}{\sum_{s=1}^{n_s} PK_s} \times 100 \tag{37}$$

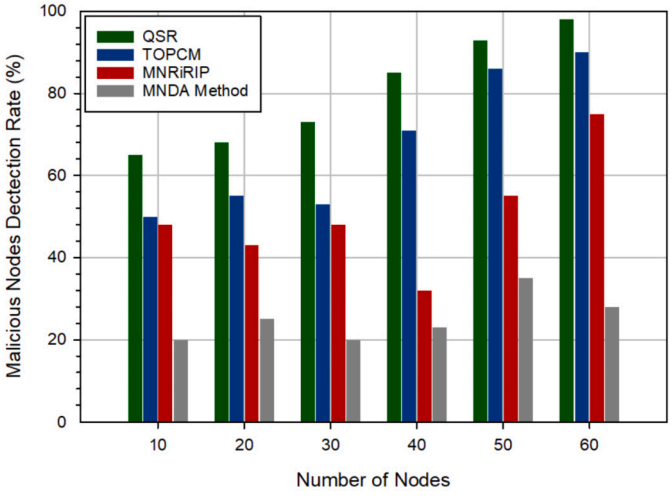


Fig. 13. Results related to the detection rate.

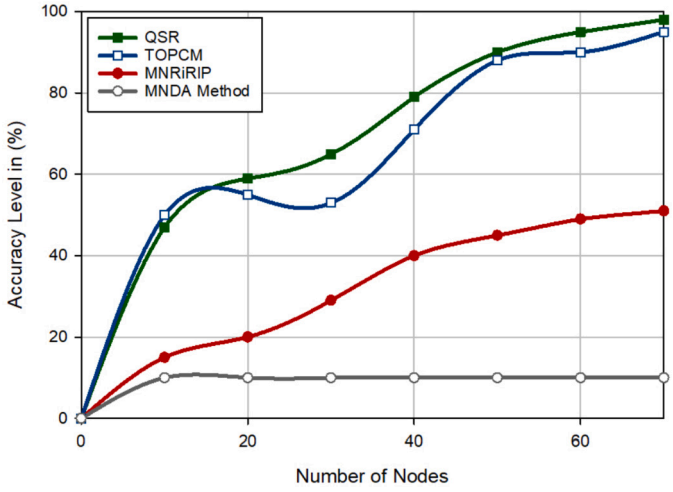


Fig. 14. Results related to the accuracy.

where PK_l indicates l-th lost packet and n_l is the number of these packets. Additionally, PK_s is s-th sent packet, and n_s introduces the number of sent packets.
- **End-to-end (EED):** Equation (38) is used to calculate this scale and represents the time spent to forward a data packet.

$$EED = \frac{\sum_{PK_i \in P = \{PK_1, \ldots, PK_p\}} \left(t_D\left(PK_i\right) - t_S\left(PK_i\right)\right)}{\sum_{i=1}^{p} PK_i} \tag{38}$$

so that PK_i means i-th delivered packet, p is the number of these packets, $t_D\left(PK_i\right)$ means the moment of delivering PK_i, and $t_S\left(PK_i\right)$ indicates the forwarding time of PK_i.

7.1. Detection rate and accuracy

The purpose of the malicious node detection rate is to test the detection power of different schemes, and their ability to isolate hostile nodes. See the results of this experiment in Fig. 13. This experiment proves that QSR detects hostile nodes well and enhances this rate by 19.01% and 60.12%, and more than three times in comparison with TOPCM, MNRiRIP, and MNDA, respectively. On the other hand, accuracy is a valuable metric to evaluate different schemes. This metric is the ratio of the correct detections to all detections. The results of this experiment are shown in Fig. 14. This figure proves that QSR is very

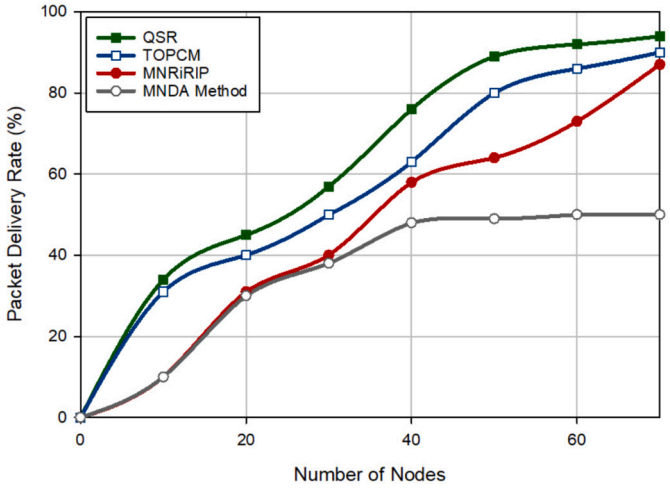


Fig. 15. Results related to the packet delivery rate.

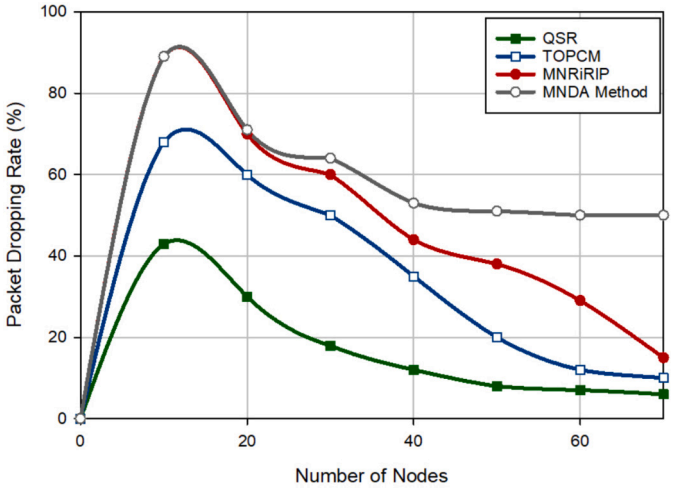


Fig. 16. Results related to the packet loss rate.

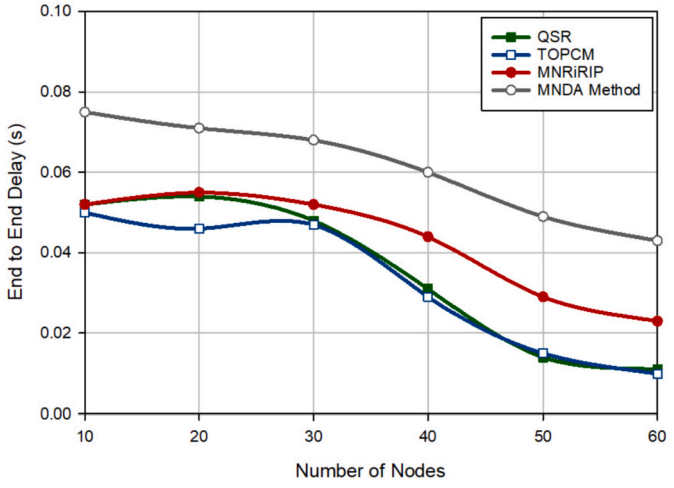


Fig. 17. Delay results.

accurate and 6.71% better than TOPCM. The two experiments confirm that QSR provides a powerful and efficient defensive system to deal with wormhole attacks. The defensive system includes two security mechanisms, namely local monitoring system and Q-learning-based secure routing. In the local monitoring system, drones monitor the behavior of each other to identify and isolate wormholes locally. As a result, these nodes cannot disrupt the routing process. Furthermore, the secure routing procedure has a very good performance in detecting wormholes because this process proposes a reward function according to the behavior of encapsulation-based wormholes. This function consists of five factors, namely delay, hop counts, PLR, PTF, and PRF. Hence, secure routes are identified for transmitting data packets to the destination node. Hence, wormholes cannot negatively affect the routing process. This has successfully improved the accuracy of QSR. However, TOPCM mainly focuses on changes made to packets and attempts to detect hostile UAVs by controlling the packet identifier, destination address, the identifier of the next hop, and the identifier of the current UAV. MNRiRIP and MNDA have shown poor performance in terms of malicious node detection rate and accuracy. And, MNDA is the weakest method because it relies only on the delay parameter for the detection and isolation of hostile UAVs on the network.

7.2. Packet delivery rate and packet loss rate

Packet delivery rate (PDR) is a valuable metric for evaluating the data transfer process. See the results of this metric in Fig. 15. According to the graphs presented in this figure, QSR can provide higher PDR than TOPCM (10.67%), MNRiRIP (34.15%), and MNDA (70.07%). On the other hand, PLR is another metric that shows the percentage of missing packets in the data transfer process. See the results of this metric in Fig. 16. This figure proves that QSR has less PLR than TOPCM (51.39%), MNRiRIP (64.07%), and MNDA (71.03%) because the Q-learning-based security system in QSR and the local monitoring system in the secure neighbor discovery phase have great power in identifying neighboring hostile UAVs and preventing them from participating in the routing process. The local monitoring system detects hostile UAVs locally and separates these nodes. In the Q-learning-based secure routing process, a reward function is presented according to the behavior of encapsulation-based wormholes. PLR is considered in this reward function because PLR in these malicious nodes is high due to very high traffic. Therefore, the Q-learning-based routing process will penalize nodes with high PLR. Hence, these UAVs cannot participate in the routing procedure. This has greatly improved PDR in QSR.

7.3. Delay

The results of end-to-end delay (EED) are represented in Fig. 17. It shows that QSR can be successful in EED. However, it has more EED than TOPCM (approximately 6.71%), but EED in QSR is 17.65% and 42.62% lower than MNRiRIP and MNDA, respectively. The reason for the successful performance of QSR compared to MNRiRIP and MNDA is that QSR considers the average single-hop delay between two nodes in the routing path when designing a reward function in the routing procedure. As a result, nodes with high delay will be penalized because this delay may be due to the encapsulation process in the wormhole tunnel. This scale is effective in preventing this attack. However, QSR has more EED than TOPCM. This is because QSR uses the Q-learning algorithm, which increases EED because it takes time to converge the optimal response.

8. Conclusion

This paper suggested a new Q-learning-based secure routing approach (QSR) in FANETs. QSR has particularly focused on the wormhole attack because this attack weakens network performance and provides the conditions to launch other attacks. QSR includes a secure neighbor discovery process to obtain information about local network topology. In the following, the security of this process is guaranteed using a local monitoring system. In QSR, UAVs decide on the best path using a distributed Q-learning algorithm. To achieve this goal, QSR designs the

reward value according to delay, hop count, consumed energy, PLR, PTF, and PRF. Also, QSR considers the dynamic topology of FANET when adjusting the learning parameters and uses a preventative system to avoid routing holes, wormhole attacks, and link failures. The performance of QSR has been studied under various experimental scenarios, and its results are compared with TOPCM, MNRiRIP, and MNDA. These experiments proved that QSR has a very good accuracy, which is 6.71% higher than TOPCM. Moreover, QSR improves the detection rate by 19.01%, 60.12%, and more than three times, PDR by 10.67%, 34.15%, 70.07%, and PLR by 51.39%, 64.07%, and 71.03% compared to TOPCM, MNRiRIP, and MNDA. However, it has more EED than TOPCM (approximately 6.71%).

In future research directions, the convergence speed of the Q-learning-based secure routing scheme must be improved because the Q-learning algorithm is not suitable for an environment with a large state space. To solve this problem, a deep reinforcement learning algorithm such as DQN can be used to handle situations with a large state space. This idea can reduce delay in the data transfer process because deep learning techniques are much faster than traditional learning algorithms. Furthermore, FANET is a dynamic network. Thus, adjusting some parameters, such as the beacon broadcast time, is very important. It is suggested as a promising direction for future research. Moreover, because of the effect of the mobility of UAVs in the simulation results, the proposed routing scheme can be evaluated through extensive simulations to show its performance under various mobility models, such as Gauss Markov Mobility Model. Also, a detailed review of cyberattacks in FANET or other ad hoc networks, such as MANET, can be considered in the future.

CRediT authorship contribution statement

Mehdi Hosseinzadeh: Writing – original draft, Data curation. **Saqib Ali:** Writing – original draft, Methodology. **Husham Jawad Ahmad:** Writing – original draft, Conceptualization. **Faisal Alanazi:** Writing – original draft, Data curation. **Mohammad Sadegh Yousefpoor:** Writing – original draft, Conceptualization. **Efat Yousefpoor:** Writing – original draft, Data curation. **Omed Hassan Ahmed:** Methodology, Data curation. **Amir Masoud Rahmani:** Writing – original draft, Methodology. **Sang-Woong Lee:** Writing – original draft, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

References

[1] A. Rovira-Sugranes, A. Razi, F. Afghah, J. Chakareski, A review of AI-enabled routing protocols for UAV networks: trends, challenges, and future outlook, Ad Hoc Netw. 130 (2022) 102790, https://doi.org/10.1016/j.adhoc.2022.102790.

[2] D.S. Lakew, U. Sa'ad, N.N. Dao, W. Na, S. Cho, Routing in flying ad hoc networks: a comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1071–1120, https://doi.org/10.1109/COMST.2020.2982452.

[3] A.M. Rahmani, S. Ali, E. Yousefpoor, M.S. Yousefpoor, D. Javaheri, P. Lalbakhsh, O.H. Ahmed, M. Hosseinzadeh, S.W. Lee, OLSR+: a new routing method based on fuzzy logic in flying ad-hoc networks (FANETs), Veh. Commun. 36 (2022) 100489, https://doi.org/10.1016/j.vehcom.2022.100489.

[4] L. Abualigah, A. Diabat, P. Sumari, A.H. Gandomi, Applications, deployments, and integration of Internet of drones (iod): a review, IEEE Sens. J. 21 (22) (2021) 25532–25546, https://doi.org/10.1109/JSEN.2021.3114266.

[5] P. Boccadoro, D. Striccoli, L.A. Grieco, An extensive survey on the Internet of drones, Ad Hoc Netw. 122 (2021) 102600, https://doi.org/10.1016/j.adhoc.2021.102600.

[6] O.S. Oubbati, M. Atiquzzaman, P. Lorenz, M.H. Tareque, M.S. Hossain, Routing in flying ad hoc networks: survey, constraints, and future challenge perspectives, IEEE Access 7 (2019) 81057–81105.

[7] B. Alzahrani, O.S. Oubbati, A. Barnawi, M. Atiquzzaman, D. Alghazzawi, UAV assistance paradigm: state-of-the-art in applications and challenges, J. Netw. Comput. Appl. 166 (2020) 102706.

[8] S.W. Lee, S. Ali, M.S. Yousefpoor, E. Yousefpoor, P. Lalbakhsh, D. Javaheri, A.M. Rahmani, M. Hosseinzadeh, An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (fanets), IEEE Access 9 (2021) 129977–130005, https://doi.org/10.1109/ACCESS.2021.3111444.

[9] M. Hosseinzadeh, S. Ali, A.H. Mohammed, J. Lansky, S. Mildeova, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, An energy-aware routing scheme based on a virtual relay tunnel in flying ad hoc networks, Alex. Eng. J. (2024), https://doi.org/10.1016/j.aej.2024.02.006.

[10] K.Y. Tsao, T. Girdler, V.G. Vassilakis, A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks, Ad Hoc Netw. 133 (2022) 102894, https://doi.org/10.1016/j.adhoc.2022.102894.

[11] J.P. Condomines, R. Zhang, N. Larrieu, Network intrusion detection system for UAV ad-hoc communication, https://doi.org/10.1016/j.adhoc.2018.09.004, 2018.

[12] M.K. Hasan, A.A. Habib, Z. Shukur, F. Ibrahim, S. Islam, M.A. Razzaque, Review on cyber-physical and cyber-security system in smart grid: standards, protocols, constraints, and recommendations, J. Netw. Comput. Appl. 209 (2023) 103540, https://doi.org/10.1016/j.jnca.2022.103540.

[13] Z. El-Rewini, K. Sadatsharan, D.F. Selvaraj, S.J. Plathottam, P. Ranganathan, Cybersecurity challenges in vehicular communications, Veh. Commun. 23 (100214) (2020) 10-1016, https://doi.org/10.1016/j.vehcom.2019.100214.

[14] S. Lateef, M. Rizwan, M.A. Hassan, Security threats in flying ad hoc network (FANET), Comput. Intell. Unmanned Aerial Veh. Commun. Netw. (2022) 73–96, https://doi.org/10.1007/978-3-030-97113-7_5.

[15] J. Sharma, P.S. Mehra, Secure communication in IOT-based UAV networks: a systematic survey, Int. Things (2023) 100883, https://doi.org/10.1016/j.iot.2023.100883.

[16] A.S. Nair, S.M. Thampi, Flying ad hoc networks: security, authentication protocols, and future directions, in: Internet of Things and Secure Smart Environments, Chapman and Hall/CRC, 2020, pp. 223–272.

[17] M.S. Yousefpoor, H. Barati, Dynamic key management algorithms in wireless sensor networks: a survey, Comput. Commun. 134 (2019) 52–69, https://doi.org/10.1016/j.comcom.2018.11.005.

[18] M.S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, M. Hosseinzadeh, Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: a comprehensive review, J. Netw. Comput. Appl. 190 (2021) 103118, https://doi.org/10.1016/j.jnca.2021.103118.

[19] M. Hosseinzadeh, J. Yoo, S. Ali, J. Lansky, S. Mildeova, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A cluster-based trusted routing method using fire hawk optimizer (FHO) in wireless sensor networks (WSNs), Sci. Rep. 13 (1) (2023) 13046, https://doi.org/10.1038/s41598-023-40273-8.

[20] J. Lansky, A.M. Rahmani, M.H. Malik, E. Yousefpoor, M.S. Yousefpoor, M.U. Khan, M. Hosseinzadeh, An energy-aware routing method using firefly algorithm for flying ad hoc networks, Sci. Rep. 13 (1) (2023) 1323, https://doi.org/10.1038/s41598-023-27567-7.

[21] M. Hosseinzadeh, O.H. Ahmed, J. Lansky, S. Mildeova, M.S. Yousefpoor, E. Yousefpoor, J. Yoo, L. Tightiz, A.M. Rahmani, A cluster-tree-based trusted routing algorithm using Grasshopper Optimization Algorithm (GOA) in Wireless Sensor Networks (WSNs), PLoS ONE 18 (9) (2023) e0289173, https://doi.org/10.1371/journal.pone.0289173.

[22] M.A. Khan, I. Ullah, N. Kumar, O.S. Oubbati, I.M. Qureshi, F. Noor, F.U. Khanzada, An efficient and secure certificate-based access control and key agreement scheme for flying ad-hoc networks, IEEE Trans. Veh. Technol. 70 (5) (2021) 4839–4851, https://doi.org/10.1109/TVT.2021.3055895.

[23] M. Hosseinzadeh, J. Tanveer, A.M. Rahmani, K. Aurangzeb, E. Yousefpoor, M.S. Yousefpoor, A. Darwesh, S.W. Lee, M. Fazlali, A Q-learning-based smart clustering routing method in flying Ad Hoc networks, J. King Saud Univ, Comput. Inf. Sci. 36 (1) (2024) 101894, https://doi.org/10.1016/j.jksuci.2023.101894.

[24] R. Kumar, B. Sharma, S. Athithan, TBMR: trust based multi-hop routing for secure communication in flying ad-hoc networks, Wirel. Netw. (2023) 1–17, https://doi.org/10.1007/s11276-023-03480-9.

[25] S. Yu, J. Lee, A.K. Sutrala, A.K. Das, Y. Park, LAKA-UAV: lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad-hoc networks, Comput. Netw. 224 (2023) 109612, https://doi.org/10.1016/j.comnet.2023.109612.

[26] Y. Tan, J. Liu, N. Kato, Blockchain-based key management for heterogeneous flying ad hoc network, IEEE Trans. Ind. Inform. 17 (11) (2020) 7629–7638, https://doi.org/10.1109/TII.2020.3048398.

[27] D. Manivannan, S.S. Moni, S. Zeadally, Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETworks (VANETs), Veh. Commun. 25 (2020) 100247, https://doi.org/10.1016/j.vehcom.2020.100247.

[28] K. Khullar, Y. Malhotra, A. Kumar, Decentralized and secure communication architecture for fanets using blockchain, Proc. Comput. Sci. 173 (2020) 158–170, https://doi.org/10.1016/j.procs.2020.06.020.
[29] Q. Su, H. Wang, C. Sun, B. Li, J. Li, Cyber-attacks against cyber-physical power systems security: state estimation, attacks reconstruction and defense strategy, Appl. Math. Comput. 413 (2022) 126639, https://doi.org/10.1016/j.amc.2021.126639.
[30] S.K. Shandilya, S. Upadhyay, A. Kumar, A.K. Nagar, AI-assisted computer network operations testbed for nature-inspired cyber security based adaptive defense simulation and analysis, Future Gener. Comput. Syst. 127 (2022) 297–308, https://doi.org/10.1016/j.future.2021.09.018.
[31] H. Hu, Y. Liu, C. Chen, H. Zhang, Y. Liu, Optimal decision making approach for cyber security defense using evolutionary game, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1683–1700, https://doi.org/10.1109/TNSM.2020.2995713.
[32] K. Kim, J.S. Kim, S. Jeong, J.H. Park, H.K. Kim, Cybersecurity for autonomous vehicles: review of attacks and defense, Comput. Secur. 103 (2021) 102150, https://doi.org/10.1016/j.cose.2020.102150.
[33] X. Sun, F.R. Yu, P. Zhang, A survey on cyber-security of connected and autonomous vehicles (CAVs), IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 6240–6259, https://doi.org/10.1109/TITS.2021.3085297.
[34] M. Hosseinzadeh, J. Yoo, S. Ali, J. Lansky, S. Mildeova, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A fuzzy logic-based secure hierarchical routing scheme using firefly algorithm in Internet of Things for healthcare, Sci. Rep. 13 (1) (2023) 11058, https://doi.org/10.1038/s41598-023-38203-9.
[35] M. Hosseinzadeh, J. Tanveer, A. Masoud Rahmani, E. Yousefpoor, M. Sadegh Yousefpoor, F. Khan, A. Haider, A cluster-tree-based secure routing protocol using dragonfly algorithm (DA) in the Internet of Things (IoT) for smart agriculture, Mathematics 11 (1) (2022) 80, https://doi.org/10.3390/math11010080.
[36] S.A. Khah, A. Barati, H. Barati, A dynamic and multi-level key management method in wireless sensor networks (WSNs), Comput. Netw. (2023) 109997, https://doi.org/10.1016/j.comnet.2023.109997.
[37] S. Islam, S. Badsha, I. Khalil, M. Atiquzzaman, C. Konstantinou, A triggerless backdoor attack and defense mechanism for intelligent task offloading in multi-UAV systems, IEEE Int. Things J. 10 (7) (2022) 5719–5732, https://doi.org/10.1109/JIOT.2022.3172936.
[38] Z. Wang, Y. Li, S. Wu, Y. Zhou, L. Yang, Y. Xu, T. Zhang, Q. Pan, A survey on cybersecurity attacks and defenses for unmanned aerial systems, J. Syst. Archit. 138 (2023) 102870, https://doi.org/10.1016/j.sysarc.2023.102870.
[39] A.E. Omolara, M. Alawida, O.I. Abiodun, Drone cybersecurity issues, solutions, trend insights and future perspectives: a survey, Neural Comput. Appl. (2023) 1–39, https://doi.org/10.1007/s00521-023-08857-7.
[40] A. Mairaj, A.Y. Javaid, Game theoretic solution for an Unmanned Aerial Vehicle network host under DDoS attack, Comput. Netw. 211 (2022) 108962, https://doi.org/10.1016/j.comnet.2022.108962.
[41] J. Sharma, P.S. Mehra, Secure communication in IOT-based UAV networks: a systematic survey, Int. Things (2023) 100883, https://doi.org/10.1016/j.iot.2023.100883.
[42] H.J. Hadi, Y. Cao, K.U. Nisa, A.M. Jamil, Q. Ni, A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs, J. Netw. Comput. Appl. 213 (2023) 103607, https://doi.org/10.1016/j.jnca.2023.103607.
[43] A. Altaweel, H. Mukkath, I. Kamel, GPS spoofing attacks in FANETs: a systematic literature review, IEEE Access (2023), https://doi.org/10.1109/ACCESS.2023.3281731.
[44] F. Tlili, L.C. Fourati, S. Ayed, B. Ouni, Investigation on vulnerabilities, threats and attacks prohibiting UAVs charging and depleting UAVs batteries: assessments & countermeasures, Ad Hoc Netw. 129 (2022) 102805, https://doi.org/10.1016/j.adhoc.2022.102805.
[45] T. Yin, Z. Gu, X. Xie, Observer-based event-triggered sliding mode control for secure formation tracking of multi-UAV systems, IEEE Trans. Netw. Sci. Eng. 10 (2) (2022) 887–898, https://doi.org/10.1109/TNSE.2022.3223978.
[46] A. Derhab, O. Cheikhrouhou, A. Allouch, A. Koubaa, B. Qureshi, M.A. Ferrag, L. Maglaras, F.A. Khan, Internet of drones security: taxonomies, open issues, and future directions, Veh. Commun. (2022) 100552, https://doi.org/10.1016/j.vehcom.2022.100552.
[47] T.T. Nguyen, V.J. Reddi, Deep reinforcement learning for cyber security, IEEE Trans. Neural Netw. Learn. Syst. (2021), https://doi.org/10.1109/TNNLS.2021.3121870.
[48] J.P. Yaacoub, H. Noura, O. Salman, A. Chehab, Security analysis of drones systems: attacks, limitations, and recommendations, Int. Things 11 (2020) 100218, https://doi.org/10.1016/j.iot.2020.100218.
[49] M. Khabbazian, H. Mercier, V.K. Bhargava, Severity analysis and countermeasure for the wormhole attack in wireless ad hoc networks, IEEE Trans. Wirel. Commun. 8 (2) (2009) 736–745, https://doi.org/10.1109/TWC.2009.070536.
[50] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, L. Chang, Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach, in: IEEE Wireless Communications and Networking Conference, March 2005, vol. 2, IEEE, 2005, pp. 1193–1199.
[51] H. Alsulami, Implementation analysis of reliable unmanned aerial vehicles models for security against cyber-crimes: attacks, tracebacks, forensics and solutions, Comput. Electr. Eng. 100 (2022) 107870, https://doi.org/10.1016/j.compeleceng.2022.107870.
[52] O.R. Ahutu, H. El-Ocla, Centralized routing protocol for detecting wormhole attacks in wireless sensor networks, IEEE Access 8 (2020) 63270–63282, https://doi.org/10.1109/ACCESS.2020.2983438.
[53] Y. Liu, M. Dong, K. Ota, A. Liu, ActiveTrust: secure and trustable routing in wireless sensor networks, IEEE Trans. Inf. Forensics Secur. 11 (9) (2016) 2013–2027, https://doi.org/10.1109/TIFS.2016.2570740.
[54] A. Shafique, A. Mehmood, M. Elhadef, Survey of security protocols and vulnerabilities in unmanned aerial vehicles, IEEE Access 9 (2021) 46927–46948, https://doi.org/10.1109/ACCESS.2021.3066778.
[55] H. Guo, J. Li, J. Liu, N. Tian, N. Kato, A survey on space-air-ground-sea integrated network security in 6G, IEEE Commun. Surv. Tutor. 24 (1) (2021) 53–87, https://doi.org/10.1109/COMST.2021.3131332.
[56] M. Hanif, H. Ashraf, Z. Jalil, N.Z. Jhanjhi, M. Humayun, S. Saeed, A.M. Almuhaideb, AI-based wormhole attack detection techniques in wireless sensor networks, Electronics 11 (15) (2022) 2324, https://doi.org/10.3390/electronics11152324.
[57] M. Hosseinzadeh, S. Ali, L. Ionescu-Feleaga, B.S. Ionescu, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, A novel Q-learning-based routing scheme using an intelligent filtering algorithm for flying ad hoc networks (FANETs), J. King Saud Univ, Comput. Inf. Sci. 35 (10) (2023) 101817, https://doi.org/10.1016/j.jksuci.2023.101817.
[58] S. Jamali, R. Fotohi, Defending against wormhole attack in MANET using an artificial immune system, New Rev. Inf. Netw. 21 (2) (2016) 79–100, https://doi.org/10.1080/13614576.2016.1247741.
[59] M. Ezhilarasi, L. Gnanaprasanambikai, A. Kousalya, M. Shanmugapriya, A novel implementation of routing attack detection scheme by using fuzzy and feed-forward neural networks, Soft Comput. 27 (7) (2023) 4157–4168, https://doi.org/10.1007/s00500-022-06915-1.
[60] M. Nayfeh, Y. Li, K. Al Shamaileh, V. Devabhaktuni, N. Kaabouch, Machine learning modeling of GPS features with applications to UAV location spoofing detection and classification, Comput. Secur. 126 (2023) 103085, https://doi.org/10.1016/j.cose.2022.103085.
[61] O. Ceviz, P. Sadioglu, S. Sen, V.G. Vassilakis, A novel federated learning-based intrusion detection system for flying ad hoc networks, arXiv preprint, arXiv:2312.04135, 2023.
[62] N.I. Mowla, N.H. Tran, I. Doh, K. Chae, AFRL: adaptive federated reinforcement learning for intelligent jamming defense in FANET, J. Commun. Netw. 22 (3) (2020) 244–258, https://doi.org/10.1109/JCN.2020.000015.
[63] N.I. Mowla, N.H. Tran, I. Doh, K. Chae, Federated learning-based cognitive detection of jamming attack in flying ad-hoc network, IEEE Access 8 (2019) 4338–4350, https://doi.org/10.1109/ACCESS.2019.2962873.
[64] O. Ceviz, P. Sadioglu, S. Sen, A survey of security in UAVs and FANETs: issues, threats, analysis of attacks, and solutions, arXiv preprint, arXiv:2306.14281, 2023.
[65] S.O. Ajakwe, D.S. Kim, J.M. Lee, Drone transportation system: systematic review of security dynamics for smart mobility, IEEE Int. Things J. (2023), https://doi.org/10.1109/JIOT.2023.3266843.
[66] J. Lansky, S. Ali, A.M. Rahmani, M.S. Yousefpoor, E. Yousefpoor, F. Khan, M. Hosseinzadeh, Reinforcement learning-based routing protocols in flying ad hoc networks (FANET): a review, Mathematics 10 (16) (2022) 3017, https://doi.org/10.3390/math10163017.
[67] J. Lansky, A.M. Rahmani, M. Hosseinzadeh, Reinforcement learning-based routing protocols in vehicular ad hoc networks for intelligent transport system (ITS): a survey, Mathematics 10 (24) (2022) 4673, https://doi.org/10.3390/math10244673.
[68] M.U. Khan, M. Hosseinzadeh, A. Mosavi, An intersection-based routing scheme using Q-learning in vehicular Ad Hoc networks for traffic management in the intelligent transportation system, Mathematics 10 (20) (2022) 3731, https://doi.org/10.3390/math10203731.
[69] A.M. Rahmani, R.A. Naqvi, E. Yousefpoor, M.S. Yousefpoor, O.H. Ahmed, M. Hosseinzadeh, K. Siddique, A Q-learning and fuzzy logic-based hierarchical routing scheme in the intelligent transportation system for smart cities, Mathematics 10 (22) (2022) 4192, https://doi.org/10.3390/math10224192.
[70] H. Jeong, S.W. Lee, M. Hussain Malik, E. Yousefpoor, M.S. Yousefpoor, O.H. Ahmed, M. Hosseinzadeh, A. Mosavi, SecAODV: a secure healthcare routing scheme based on hybrid cryptography in wireless body sensor networks, Front. Med. 9 (2022) 829055, https://doi.org/10.3389/fmed.2022.829055.
[71] V.F. Sangeetha Francelin, J. Daniel, S. Velliangiri, Intelligent agent and optimization-based deep residual network to secure communication in UAV network, Int. J. Intell. Syst. 37 (9) (2022) 5508–5529, https://doi.org/10.1002/int.22800.
[72] K. Singh, A.K. Verma, TBCS: a trust based clustering scheme for secure communication in flying ad-hoc networks, Wirel. Pers. Commun. 114 (2020) 3173–3196, https://doi.org/10.1007/s11277-020-07523-8.
[73] W. Buksh, Y. Guo, S. Iqbal, K.N. Qureshi, J. Lloret, Trust-oriented peered customized mechanism for malicious nodes isolation for flying ad hoc networks, Trans. Emerg. Telecommun. Technol. (2022) e4489, https://doi.org/10.1002/ett.4489.
[74] D. Muruganandam, J. Manickam, M. Leo, An efficient technique for mitigating stealthy attacks using MNDA in MANET (Retracted article. See Dec. 2022), Neural Comput. Appl. 31 (2019) 15–22, https://doi.org/10.1007/s00521-018-3634-7.
[75] M. Hosseinzadeh, A.H. Mohammed, F.A. Alenizi, M.H. Malik, E. Yousefpoor, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A novel fuzzy trust-based secure routing scheme in flying ad hoc networks, Veh. Commun. (2023) 100665, https://doi.org/10.1016/j.vehcom.2023.100665.
[76] R. Fotohi, Securing of Unmanned Aerial Systems (UAS) against security threats using human immune system, Reliab. Eng. Syst. Saf. 193 (2020) 106675, https://doi.org/10.1016/j.ress.2019.106675.

[77] R. Fotohi, E. Nazemi, F.S. Aliee, An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks, Veh. Commun. 26 (2020) 100267, https://doi.org/10.1016/j.vehcom.2020.100267.
[78] M. Hosseinzadeh, J. Tanveer, L. Ionescu-Feleaga, B.S. Ionescu, M.S. Yousefpoor, E. Yousefpoor, O.H. Ahmed, A.M. Rahmani, A. Mehmood, A greedy perimeter stateless routing method based on a position prediction mechanism for flying ad hoc networks, J. King Saud Univ, Comput. Inf. Sci. 35 (8) (2023) 101712, https://doi.org/10.1016/j.jksuci.2023.101712.
[79] X. Pang, M. Liu, Z. Li, B. Gao, X. Guo, Geographic position based hopless opportunistic routing for UAV networks, Ad Hoc Netw. 120 (2021) 102560, https://doi.org/10.1016/j.adhoc.2021.102560.
[80] J. Lansky, A.M. Rahmani, S.M. Zandavi, V. Chung, E. Yousefpoor, M.S. Yousefpoor, F. Khan, M. Hosseinzadeh, A Q-learning-based routing scheme for smart air quality monitoring system using flying ad hoc networks, Sci. Rep. 12 (1) (2022) 20184, https://doi.org/10.1038/s41598-022-20353-x.
[81] S. Hameed, S. Alyahya, Q.A. Minhas, S. Habib, A. Nawaz, S. Ahmed, A. Ishtiaq, M. Islam, S. Khan, Link and loss aware GW-COOP routing protocol for FANETs, IEEE Access 9 (2021) 110544–110557, https://doi.org/10.1109/ACCESS.2021.3101361.
[82] S. Kumar, N.K. Rathore, M. Prajapati, S.K. Sharma, SF-GoeR: an emergency information dissemination routing in flying ad-hoc network to support healthcare monitoring, J. Ambient Intell. Humaniz. Comput. 14 (7) (2023) 9343–9353, https://doi.org/10.1007/s12652-022-04434-3.
[83] G. Gankhuyag, A.P. Shrestha, S.J. Yoo, Robust and reliable predictive routing strategy for flying ad-hoc networks, IEEE Access 5 (2017) 643–654, https://doi.org/10.1109/ACCESS.2017.2647817.
[84] M.Y. Arafat, S. Moh, A Q-learning-based topology-aware routing protocol for flying ad hoc networks, IEEE Int. Things J. 9 (3) (2021) 1985–2000, https://doi.org/10.1109/JIOT.2021.3089759.
[85] L.A.L. da Costa, R. Kunst, E.P. de Freitas, Q-FANET: improved Q-learning based routing protocol for FANETs, Comput. Netw. 198 (2021) 108379, https://doi.org/10.1016/j.comnet.2021.108379.
[86] S. Adams, T. Cody, P.A. Beling, A survey of inverse reinforcement learning, Artif. Intell. Rev. 55 (6) (2022) 4307–4346, https://doi.org/10.1007/s10462-021-10108-x.
[87] R.F. Prudencio, M.R. Maximo, E.L. Colombini, A survey on offline reinforcement learning: taxonomy, review, and open problems, IEEE Trans. Neural Netw. Learn. Syst. (2023), https://doi.org/10.1109/TNNLS.2023.3250269.
[88] Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, J. Morimoto, Deep learning, reinforcement learning, and world models, Neural Netw. 152 (2022) 267–275, https://doi.org/10.1016/j.neunet.2022.03.037.
[89] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, Y. Xu, QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks, Comput. Commun. 150 (2020) 304–316, https://doi.org/10.1016/j.comcom.2019.11.011.
[90] W. Stallings, IEEE 8O2. 11: wireless LANs from a to n, IT Prof. 6 (5) (2004) 32–37, https://doi.org/10.1109/MITP.2004.62.
[91] Y. Chen, N. Zhao, Z. Ding, M.S. Alouini, Multiple UAVs as relays: multi-hop single link versus multiple dual-hop links, IEEE Trans. Wirel. Commun. 17 (9) (2018) 6348–6359, https://doi.org/10.1109/TWC.2018.2859394.
[92] N. Goddemeier, C. Wietfeld, Investigation of air-to-air channel characteristics and a UAV specific extension to the rice model, in: 2015 IEEE Globecom Workshops (GC Wkshps), IEEE, December 2015, pp. 1–5.
[93] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE, January 2000, p. 10.
[94] H. Ghazzai, M.B. Ghorbel, A. Kadri, M.J. Hossain, Energy efficient 3D positioning of micro unmanned aerial vehicles for underlay cognitive radio systems, in: Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May, 2017.
[95] I. Khalil, S. Bagchi, N.B. Shroff, June. LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks, in: 2005 International Conference on Dependable Systems and Networks (DSN'05), IEEE, 2005, pp. 612–621.
[96] T. Limbasiya, K.Z. Teng, S. Chattopadhyay, J. Zhou, A systematic survey of attack detection and prevention in connected and autonomous vehicles, Veh. Commun. (2022) 100515, https://doi.org/10.1016/j.vehcom.2022.100515.