Security Clustering Algorithm Based on Integrated Trust Value for Unmanned Aerial Vehicles Network

Jingxian Zhou[1] and Zengqi Wang[1,2]
[1]Information Security Evaluation Center, Civil Aviation University of China,
Tianjin 300300, China
[e-mail: yzzxtj@aliyun.com]
[2]School of Computer Science and Technology, Civil Aviation University of China,
Tianjin 300300, China
[e-mail: wangzengqi77@163.com]
*Corresponding author: Jingxian Zhou

Received September 29, 2019; revised December 30, 2019; accepted February 13, 2020; published April 30, 2020

Abstract

Unmanned aerial vehicles (UAVs) network are a very vibrant research area nowadays. They have many military and civil applications. Limited bandwidth, the high mobility and secure communication of micro UAVs represent their three main problems. In this paper, we try to address these problems by means of secure clustering, and a security clustering algorithm based on integrated trust value for UAVs network is proposed. First, an improved the k-means++ algorithm is presented to determine the optimal number of clusters by the network bandwidth parameter, which ensures the optimal use of network bandwidth. Second, we considered variables representing the link expiration time to improve node clustering, and used the integrated trust value to rapidly detect malicious nodes and establish a head list. Node clustering reduce impact of high mobility and head list enhance the security of clustering algorithm. Finally, combined the remaining energy ratio, relative mobility, and the relative degrees of the nodes to select the best cluster head. The results of a simulation showed that the proposed clustering algorithm incurred a smaller computational load and higher network security.

Keywords: UAVs network, clustering algorithm, cluster head selection, Bayesian trust model, network security.

1. Introduction

With the rapid development of artificial intelligence, sensors and communication technology, significant progress has been made in drone technology as well. Owing to its simple structure, good concealment, and flexible operation, drones have been widely applied in the military and civilian life [1, 2]. Flying Ad-hoc networks (FANETs) are a typical application of mobile ad-hoc networks (MANETs). FANETs are a temporary communication network composed of a large number of drones to accomplish co-operational tasks[3]. The network has the characteristics of a mobile self-organizing network without a center, and with self-organization, and a dynamic topology. It faces such challenges as frequent topology updates and limited node energy at high dynamics [4]. Network management becomes especially difficult in large-scale and high-speed mobile networks, and clustering is an effective means of optimizing network management [5]. In clustering, the whole network is divided into multiple logical groups or clusters. Each cluster has a leader or cluster head, which is responsible for inter-cluster and intra-cluster communication. Clustering helps to make the network more scalable, reduce routing, overhead and maximize the throughput.

During network clustering, nodes perform computations in order to arrange nearby located nodes into cluster members (CMs) and a cluster head (CH) is elected from each cluster. The process of electing CHs is very important in maintaining the cluster structure [6]. Subsequent changes in the relative positions of CMs can modify the cluster structure. In order to track changes in cluster structure, CHs should always periodically broadcast their existence to its CMs, and each CM should reply back its status to the CH. The lifetime of the cluster is a parameter, among others, that is used for evaluating the performance of a clustering model. The longer the lifetime of the cluster, the less will be the cluster maintenance overhead and more effective the model will be. The lifetime of the cluster depends upon the efficient selection of CHs. In the highly dynamic environment and with the limited processing power of UAVs, computationally expensive techniques are also not suitable for clustering in FANETs.

Moreover, because the cluster network used in an unmanned aerial vehicles (UAVs) adopts an open communication mechanism, it exposes a large surface to adversaries, which poses significant network security challenges to the drone system [7]. For example, DDoS (Distributed Deny of Service) attack: DDoS attack is an attempt to make the UAVs systems unavailable/unreachable by overwhelming it with traffic from multiple sources. Attacks against the UAVs network are classified into external and internal attacks. External attacks refer to attacks aiming at a communication link, such as monitoring and interference [8]. Internal attacks refer to the capture of a single UAVs node. After analyzing the node's code and mastering its network protocol, the attackers insert malicious nodes into numbers of the network, and execute data eavesdropping and behaviors that are destruction to the network, such as hello flood attacks [9] and sybil witch attacks [10]. However, there is no particularly perfect security clustering algorithms for cluster UAVs.

In this paper, we proposed a secure clustering algorithm for the self-organizing networks of UAVs by considering their dynamic structure, poor stability, and vulnerability to attacks. The proposed algorithm based on a trust mechanism to ensure information security in the network clustering process. By improving the k-means++ clustering algorithm, we implement the self-organizing network of UAVs with the maximum stability under a limited bandwidth, and introduced a comprehensive trust value mechanism and an integrated cluster head election mechanism to select credible cluster heads that improve performance.

The rest of the paper is organized as follows. In Section 2, a review is existing clustering algorithm is presented. In Section 3, we provides some fundamental knowledge. The proposed clustering algorithm was described in detail in Section 4. Simulation and verification results are given in Section 5 and Section 6 we conclude the paper.

2. Related Work

Several clustering algorithms have been proposed in research on wireless ad-hoc networks, such as the weighted clustering, the minimum ID, the lowest mobility and the highest node degree. The weighted clustering algorithm (WCA) [11] is different from the other algorithms in that it does not consider only a single factor but integrates various factors, such as node degree, mobility, residual energy, and position of neighboring nodes. It is more suitable for different application scenarios compared with clustering algorithms that consider single factors. In [12], Hussein et al. proposed flexible weighted clustering algorithm which is based on battery power. The basic goal of this algorithm is to prevent nodes having low battery power to be elected as CH. Another method proposed by Kaur and Singh [13] is strength based energy effcient algorithm, where a node with highest energy level is elected as CH. The main goal of this algorithm is to handle the CH election on the basis energy and signal levels.

The minimum ID clustering algorithm [14] was proposed by Lin and Gerla in 1995. In it, each node in the network corresponds to a unique ID number, and the node with the smallest ID number in the adjacent node is selected as cluster head. Gavalas et al. [15] proposed a clustering algorithm in which during clustering stage node IDs are assigned to every node based on weighted value of mobility and power consumption. A node having lowest ID is elected as a CH which controls the topology information and communication between nodes. Every node broadcast its ID by transmitting HELLO message to its neighboring nodes. Each node upon reception of Hello message from its neighbors, compares its ID with the other neighborhood nodes. If the node has the lowest ID among its neighboring nodes, it is elected as a CH. However frequent exchange of control messages can exhaust node's power and slow down the effciency of CH.

The lowest mobility clustering algorithm is used to select the node with the least mobility among adjacent nodes as cluster head. Ni et al. [16] proposed a mobility prediction based clustering algorithm by using estimation of speed of nodes relative to each other. Every node compiles the average relative speed of the neighboring nodes by exchanging Hello packets with each other. The node having lowest relative mobility is selected as a CH and other nodes become the cluster members. However the high mobility of nodes decreases the lifetime of CH. Aftab et al. [17] proposed a self-organization based clustering scheme using zone based group mobility. The algorithm uses the bio-inspired behavioral study of bird flocking for cluster formation and maintenance. Every node in the CH zone calculates the residual energy of other nodes and node having the highest residual energy is elected as a CH.

The highest node degree clustering algorithm [18] is based on the degree of connectivity needed for clustering, and selects the node with the appropriate number of neighboring nodes as cluster head by calculating the numbers of neighbors of nodes. Shi and Luo [19] proposed a mechanism of cluster based location aided dynamic source routing, where CH is elected based on the energy level, relative velocity and degree of connectivity. That UAV is elected as a CH which has higher energy level, low relative speed and a large number of neighbor nodes. Every UAV in a cluster maintains a neighbor table. In [20], mobility prediction clustering algorithm (MPCA) was proposed by Zang et al. for UAVs network. Link expiration time (LET) between

two UAVs, is calculated by adopting GPS information. LET is used to build trie-structure which refers to the lasting neighbor set of a UAV. The UAV having highest value of lasting neighbor set is elected as CH. MPCA is not effcient in energy consumption and it incurs more overhead in case of network maintenance.

Past research has generally assumed that the clustering environment is safe and reliable, and no node is compromised. However, when constructing the cluster structure of UAVs network, because its security policy cannot be formulated in the initialization stage, it is vulnerable to attackers [21]. The adaptive security weighted clustering algorithm (SWCA) [22] proposed by Ma et al. considers the connectivity, energy, mobility, and security of nodes and their neighbors, focusing on the security aspects of the network, but no improvements have been made to it with regard to energy and mobility. Jangra [23] proposed the authenticated routing protocol for ad-hoc networks (ARAN) that uses identity authentication and digital signature technology to ensure end-to-end authentication, message integrity, and non-repudiation between nodes. Sohail and Wang [24] proposed a trust-based routing protocol, three valued secure routing (3VSR), which employs the idea of sensing logic-based trust model to enhance the security solution of vehicular mobile ad-hoc networks.

These algorithms are not energy effcient pertaining the resource constraints of UAVs in FANET. They incur more overheads and have higher clustering time. The changing topology of limited resourced UAVs need effcient cluster head selection and cluster formation mechanism for effective and effcient communication in FANET. A method to improve the security of the UAVs network while reducing the complexity of node computation in the network and maximizing the balance of communication load through the secure network clustering algorithm is the focus of this paper.

3. Fundamental Knowledge

3.1 Clustered UAV Network

The clustered structure does not need to maintain complex routing information, and can quickly respond to changes in the system. It is applicable to scenarios with a large number of nodes and frequent node motion. Compared with the planar structure, the greater the number of nodes, the more advantageous is the clustering structure. Therefore, the structure of the clustered network conforms to the requirements of drone colony networking. There are three types of nodes in a UAVs clustering network: cluster heads, cluster members, and gateway nodes. The cluster head is responsible for managing the corresponding intra-cluster drone nodes, and the gateway node establishes communication between the clusters. A typical UAVs clustering network topology is shown in **Fig. 1**.

3.2 K-Means++ Algorithm

Owing to the frequent motion and wide distribution of the self-organizing UAVs network, the k-means or improved clustering algorithm is first used to divide the network into regions to make the hierarchical network more uniform [25]. The k-means algorithm divides the distributed nodes in the network into clusters. The steps are as follows:

Step1: Select each initial virtual center.

Step2: For each node, calculate the distance to every virtual initial center, and assign the node to the cluster of the nearest virtual center.

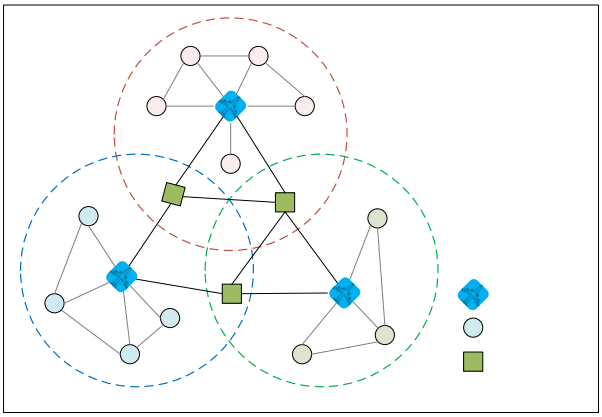


Fig. 1. Structure of typical UAVs clustering network.

Step3: By using the mean method, update the virtual center of each cluster as the initial center of the next iteration.

Step4: For the virtual center of each cluster, if it is stable or moves by less than a given threshold after several iterations, pass the iteration; otherwise, returns to *Step* 2.

The k-means++ algorithm [26] calculates the positional data of all nodes before selecting the initial center, so that the distance between the selected initial centers is longest, and the iterative process of the clustering algorithm is therefore reduced.

3.3 Bayesian Trust Model

Bayesian theory is one of the commonly used theoretical foundations in trust evaluation. Based on the statistical learning in the system information, prior probability is used to estimate the unknown state of the system, and the Bayesian probability model is applied to modify the result. The state output at the next moment is predicted by calculating the expectation of the state probability distribution of the system.

The reputation distribution model is representative of Bayesian theory. Ismail et al. proposed an e-commerce credit model based on reputation distribution [27]. This model ensures that the credit model of each participant in commercial activities complies with a certain distribution, and assesses the confidence level of each participant by calculating the expectation. In research on the reputation- based framework for sensor networks confidence evaluation model, Ganeriwal and Srivastava [28] used mathematical calculations to confirm that information transmission between nodes in wireless sensor networks conforms to a certain distribution.

The probability density is defined as Equation 1:

$$Beta\left(p|\alpha,\beta\right)=\frac{\Gamma\left(\alpha+\beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)}p^{\alpha-1}\left(1-p\right)^{\beta-1} \tag{1}$$

In the equation, α,β are the results of the frequency of a pair of binary events $\left(x,y\right)$, and $\Gamma\left(*\right)$ is the gamma function, and its definition on the set of real integers is: $\Gamma\left(n\right)=\left(n-1\right)!$. The constraints $0\le p\le 1$, $\alpha,\beta>0$ are imposed on α,β and Formula 2 is satisfied:

$$\begin{cases} p \neq 0 & \alpha < 1 \\ p \neq 1 & \beta < 1 \end{cases} \tag{2}$$

It has been proven that the *Beta* probability density conforms to Equation 3:

$$E(p) = \frac{\alpha}{\alpha + \beta} \tag{3}$$

Consider an event with only two outputs (x, y). Suppose that r, s are the frequencies of x, y, respectively, in the results of observation. According to Bayesian theory, by setting the value defined in Equation 4, the predicted probability density of observing x at the $n+1$-th instant can be expressed by the probability of the n-th observation.

$$\alpha = r + 1, \beta = s + 1 \quad r, s > 0 \tag{4}$$

According to research by Feng et al. [29], the Bayesian trust model when applied to wireless sensor networks can effectively detect malicious nodes in the network and resist routing-level attacks like the Hello flood. For the Bayesian trust model in a drone network, the trust value of a node can be calculated based on the expectation of the $Beta$ distribution.

3.4 The Requirements of Secure Clustering Algorithm

The aim of the secure clustering algorithm is to construct and maintain a cluster set that can cover the entire network with low computational and communication overheads while supporting resource management and the routing protocols. But the introduction of security policies increases the complexity and computational cost of clustering algorithms, because of which a balance between security and efficiency is an ever-present challenge to secure clustering algorithms. This section summarizes the requirements of the secure clustering algorithm in term of performance and security by analyzing the characteristics of the self-organizing cluster drone network.

3.4.1 Performance Requirements

The number of clustered nodes in the self-organizing UAVs network is large, and a long time may be needed to form a stable topology. Therefore, it is important to shorten the networking time and simplify the process of node clustering. Owing to the high relative speed of motion, and the frequent modification to network topology caused by a complex environment, it is possible for a node to join or leave the network at any time. This induces more packet loss and route reselection. Therefore, the clustering algorithm needs to not only form a more stable cluster structure, but also to minimize the number of cluster reconstructions. Strong robustness is also indispensable so that abnormal behavior by certain nodes has no significant impact on the entire network. The network resources of UAVs node is limited, because of which the network occupation of the generation and maintenance of the cluster structure should be minimized.

3.4.2 Security Requirements

Because of the lack of a management center and the difficulty of implementing a security authentication policy in an ad-hoc network, the secure clustering algorithm needs to perform entity authentication to ensure that no unauthorized node joins the network, and this certifies the source of the data as well. Wireless communication between drones is vulnerable to eavesdropping and tampering attacks to obtain secret information or damage the network topology. Therefore, to ensure the confidentiality and integrity of information transmission during the clustering process, the cluster head plays an important role in the network. Thus, the selected heads must have a high degree of credibility, because a large amount of secret information can be leaked otherwise. The topology of the self-organizing network changes dynamically, and nodes often join and exit it. The secure clustering algorithm must guarantee the safety of the forwarded data of the newly added nodes as well as the security of the backward data of the nodes that have exited. External nodes may be authorized in various ways to invade the network, steal secret information, or harm the network structure. The secure clustering algorithm thus needs to be able to detect and segregate malicious internal nodes before they can cause serious damage.

4. UAVs Clustering Algorithm Based on Trust Value

The proposed UAVs network clustering algorithm is composed of grid initialization, preliminary clustering, cluster head selection, and maintenance and update of the cluster structure. The procedure is sketched below.

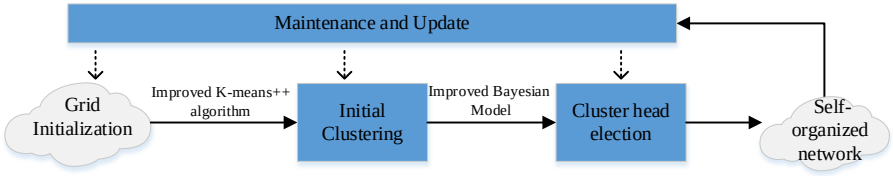


Fig. 2. Flow of UAVs network security clustering algorithm

4.1 Security Measures of Clustering Algorithm

The cryptographic mechanism guarantees the confidentiality, integrity, authentication, availability, and non-repudiation of information in the clustering process through technologies for information encryption and decryption, digital signature, and identity authentication. In this paper, we encrypt the control messages in the clustering process to prevent information from being eavesdropped on, use signatures for key information in case of tampering attacks, and block unauthorized external nodes the through digital certificates. The password mechanism is very effective for resisting external attacks but cannot defend against internal attacks. The trust mechanism can be used not only to select cluster heads with higher credibility, but also to detect and isolate malicious internal nodes. The trust mechanism evaluates the credibility of a node in the network by calculating its reputation, and preferentially elects the node with higher credibility as cluster head. Referring to the idea of the weight-based clustering algorithm, and using the trust value as the principal parameter, the node with a larger trust value is more likely to be selected as cluster head, and is thus reliable.

Moreover, when the node's trust value is below a certain threshold, it is regarded as malicious and is isolated. The trust mechanism cannot defend against external attacks but

helps resist internal attacks. The security clustering algorithm proposed in this paper combines the advantages of the cryptographic mechanism and the trust mechanism to provide internal and external protection for the clustering process.

4.2 Grid Initialization

The target of the network initialization phase is to form the list of neighbors of nodes and collect the parameters required to construct the cluster. The nodes in the network need to exchange multiple HELLO messages periodically. On the one hand, it helps them declare their existence to the neighboring nodes; on the other, the HELLO message is used to obtain the attributes of neighboring nodes, and the parameters required for the subsequent selection of cluster heads and the formation of the neighbors' list. The HELLO message of node vi has the following format:

$$\{IP, Status, Neighbor\_table, Parameters, Timestamp\}_{vi}$$

The UAV cluster first assigns a unique IP address to each UAVs node before clustering, and performs the backup and update of the *IP* list at the task management base station. The $Status$ field is used to mark the state of the node, which is initially NULL. The $Neighbor\_table$ packet contains the neighbor list of the node. Information concerning the neighborhood can be obtained by exchanging the neighbor list and checking for the existence of a given node in the neighbor list of its neighbor node, and whether the link between the neighboring node is unidirectional or bidirectional. The $Parameters$ packet contains the ratio of residual energy, relative mobility, relative node degree, and trust, four parameters required for the selection of the cluster head. The UAVs node can also obtain geographical location through the GPS positioning device [30].

4.3 Initial Clustering

To divide the intra-cluster network more evenly, all nodes in the network are classified into relatively uniform clusters by the k-means++ algorithm before the cluster head selection phase. The k-means++ clustering algorithm converges quickly, and has good scalability. However, the following problems occur for the UAVs network:

(1) The bandwidth of the UAVs communication module is limited. If the number of clusters is too large, the bandwidth cannot be fully exploited and inter-cluster node communication delay increases. If the number of clusters is too small, intra-cluster communication is congested owing to excessive load, affecting data link transmission.

(2) In the cluster structure maintenance phase, the location of the UAVs node changes dynamically, and the motion of a node affects the stability of the inner structure of the cluster.

In view of the security- and performance-related requirements of the UAVs network, this section proposes an improved k-means++ initial clustering algorithm by using the number of clusters and node clustering. The improved k-means++ algorithm consist of the following steps:

Step 1: Determine the optimal number of clusters *K* of the network based on the bandwidth.

Step 2: Calculate the positional data of all nodes, and select a scattered virtual initial center according to the result.

Step 3: Calculate the virtual distance of the link expiration time (LET) and the connection expiration time to the virtual initial center for any node, and assign the node to the cluster of the nearest virtual center;

Step 4: Use the mean method to update the virtual center of each cluster, and use it as the initial center of the next iteration.

Step 5: For the virtual center of each cluster, if its position is stable or it moves less than a given threshold after several iterations, jumps out of the iteration; otherwise, return to *Step* 3.

There are two improvements that need to be addressed in the below.

4.3.1 Determing the Optimal Number of Clusters Based on Bandwidth

Considering the load capacity of the UAV communication module, it is important to set an appropriate value K, that is, the number of clusters in the entire network. If the cluster's head node covers too few nodes in the network, the bandwidth is not completely used, but if there are too many nodes in the overlaid network, congestion occurs owing to excessive load.

Suppose that N is the total number of nodes, and B_1, B_2 are the bandwidths of node-to-node communication within a cluster and inter-cluster communication, respectively. The number of clusters of the entire network is K and Δ is the number of network topologies. To achieve the best throughput and meet the requirements of network load balance, we refer to the average throughput calculation for nodes of an ad-hoc network nodes proposed by Amorim et al. [31]. For any cluster, the upper bound of the average node throughput R_{inner} of the intra-cluster network is:

$$R_{inner} = \sqrt{\frac{8}{\pi}} \cdot \frac{B_1}{\Delta} \cdot \sqrt{\frac{N}{K}} \tag{5}$$

The upper bound $R_{external}$ of the average throughput of inter-cluster communication is:

$$R_{external} = \sqrt{\frac{8}{\pi}} \cdot \frac{B_2}{\Delta} \cdot \sqrt{K} \tag{6}$$

To attain a balance between intra-cluster and inter-cluster throughput, the following constraint must be considered:

$$\frac{K-1}{K} \cdot R_{inner} \le R_{external} \tag{7}$$

The optimal number of clusters K^* is achieved if this formula is the equation:

$$K^* = \frac{B_2}{B_1} \cdot \sqrt{N} + 1 \tag{8}$$

Dividing the UAV network into K^* clusters ensures the optimal use of network bandwidth.

4.3.2 LET Time Between Mobile Nodes

In the clustering iteration of the k-means++, the concept of the link expiration time (LET) between nodes is introduced. For a node m to be clustered and a pseudo-central node n, consider the coordinates (x_m, y_m) and (x_n, y_n), speed vectors (v_m, θ_m) and (v_n, θ_n), and the maximum transmission distance r of nodes. The distance between nodes is equal to the transmission radius r after movement for the duration of the LET along their own directions. If the movement continues, the distance between the nodes exceeds the transmission radius r and the transmission link is interrupted. The definition of $LET_{m,n}$ is shown in **Fig. 3**:

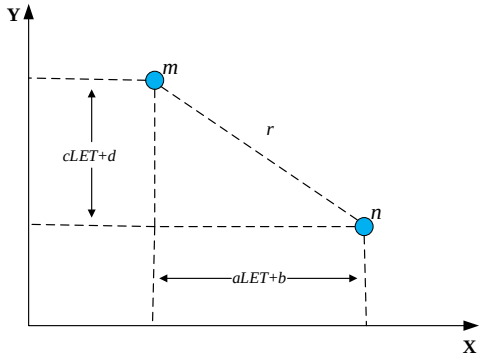


Fig. 3. Link expiration time of nodes

$$r^2 = (aLET + b)^2 + (cLET + d)^2 \tag{9}$$

where

$$a = v_m \cos\theta_m - v_n \cos\theta_n \tag{10}$$

$$b = x_m - x_n \tag{11}$$

$$c = v_m \sin\theta_m - v_n \sin\theta_n \tag{12}$$

$$d = y_m - y_n \tag{13}$$

Thus,

$$LET_{m,n} = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \tag{14}$$

In *Step* 2 of the k-means++ iteration to classify nodes into a certain cluster, a threshold T is set. When $LET_{m,n}$ is greater than T, the node to be clustered m is classified in cluster C_n with pseudo-central node n; when $LET_{m,n}$ is less than or equal to T, m is regarded as a free node. A free node searches for its cluster again, and the $LET_{m,n+1}$ between m and neighboring cluster C_{n+1} is calculated. If the direction of motion of m is toward cluster C_{n+1} and if $LET_{m,n+1} > LET_{m,n}$, the node is assigned to cluster C_{n+1}. Then, the iteration continues. But if the condition is not satisfied, cluster C_{n+1} is set as an alternative cluster for m for reference in the cluster head selection phase. The process is shown in **Fig. 4**.

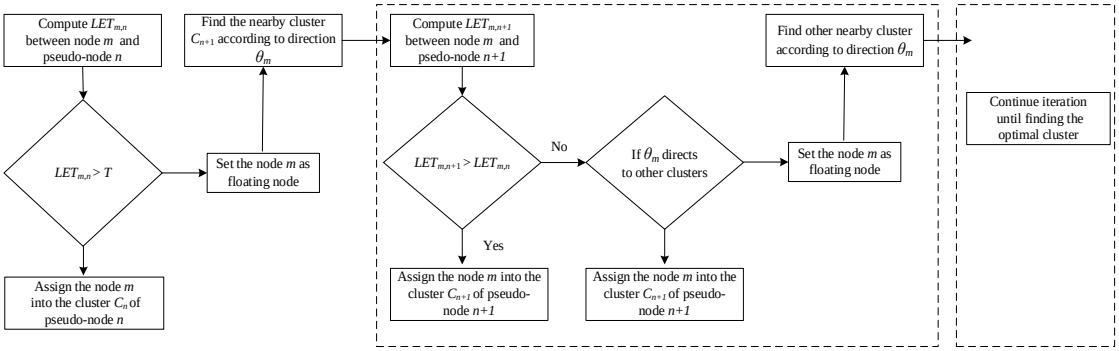


Fig. 4. Flow chart of the clustering of the free nodes

Fig. 5 shows an example. Node 1 and Node 2 are free nodes of cluster C_0. For Node 1, as the moving direction θ_1 is toward cluster C_8, the LET between Node 1 and C_8 is calculated, and the cluster it belongs to is reselected. Node 2 still belongs to cluster C_0 because θ_2 does not point to any other cluster.

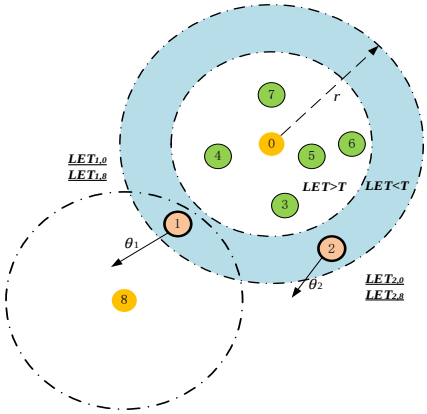


Fig. 5. Clustering assignment of free nodes (1, 2)

By clustering the network initially with refined k-means++, a list of information concerning the drone colony is created. Information form alternative nodes can be referred to during the cluster head selection phase, e.g., data on residual energy, relative mobility, and relative node degree.

4.4 Cluster Head Selection

Based on the initial refined division of the network into clusters by k-means++, nodes of each network cluster can negotiate internally to select the head node of each cluster to manage. But if a malicious node placed by an attacker is chosen as cluster head, the transmission function, the availability of the entire network and the confidentiality of the information in the cluster are affected. Therefore, in the process of cluster head selection, this paper proposes using the distributed trust value of nodes to perform pre-selection to detect and exclude malicious or compromised nodes in the network.

4.4.1 Improved Bayesian Trust Model

We apply the Bayesian trust model to the UAVs network. For any drone node i, if i needs to communicate with node j, i calculates its integrated trust value. This requires reference to the direct trust value fed back by node j and the recommended trust value fed back to i by other nodes in the same cluster.

If only the direct trust value fed back by node j is considered, assuming that the number of normal routing behaviors of the node is $\alpha_{i,j}$ and that of abnormal routing behaviors is $\beta_{i,j}$, Beta distribution fitting is performed to obtain the trust value $T_{i,j}$ of node i for node j:

$$T_{i,j} = E(Beta(\alpha_{i,j}+1, \beta_{i,j}+1)) = \frac{\alpha_{i,j}+1}{\alpha_{i,j}+\beta_{i,j}+1} \tag{15}$$

However, a reputation system that relies solely on direct trust values has a very long convergence time [16], and if node j is a compromised node, node i is vulnerable to attacks from it. In this paper, we introduce the concept of the recommendation trust value. A neighboring node sends the trust table of node j to node i for reference as the recommendation trust value, and this is used calculating total trust value $T_{i,j}$. Assume that the node receives the recommended trust value from $N-2$ nodes (i and j excluded) of the same cluster, $S\left(r_{k,j}\right)$ and $S\left(s_{k,j}\right)$ ($k \in [1, N-2]$, where $r_{k,j}$ is the number of normal route behaviors) are the numbers of normal routing behaviors, $s_{k,j}$ is the number of abnormal route behaviors, and the trust list of node i has already recorded the direct trust value of node j in the previous n iterations. According to the number of normal routing behaviors $r_{k,j}$ and the number of abnormal routing behaviors $s_{k,j}$ of node j in the iteration, node i calculates the trust value of node j according to the following formula:

$$\alpha_{i,j}^{new} = \omega \times \alpha_{i,j} + r_{i,j} + \sum_{k \in N} S(r_{k,j}) \tag{16}$$

$$\beta_{i,j}^{new} = \omega \times \beta_{i,j} + s_{i,j} + \sum_{k \in N} S(s_{k,j}) \tag{17}$$

$$T_{i,j}^{new} = E(Beta(\alpha_{i,j}^{new}+1, \beta_{i,j}^{new}+1)) \tag{18}$$

Where ω is the forgetting factor, $\omega \in (0,1)$. Because the influence of the historical trust value is weakened, the malicious node cannot become the cluster head node by accumulating trust value. To emphasize the importance of recent interactions in the calculation of trust values, this paper sets the forgetting factor as a function of the number of rounds n.

At the same time, to prevent the recommended node from launching a bad-mouthing attack, which is to provide a false trust recommendation value to reduce the trust value of a good node or enhance that of a malicious node in collusion, we used Dempster-Shafer trust theory [32]. By establishing a model of the recommendation trust value, the weight of the trust value directly observed by node i indirectly increases, and the resistance of the overall network to the malicious smash attack improves.

$$S(r_{k,j}) = \frac{2\alpha_{i,k}.r_{k,j}}{(\beta_{i,j}+2)(r_{k,j}+s_{k,j}+2)(2\alpha_{i,k})} \tag{19}$$

$$S(s_{k,j}) = \frac{2\alpha_{i,k}.s_{k,j}}{(\beta_{i,j}+2)(r_{k,j}+s_{k,j}+2)(2\alpha_{i,k})} \tag{20}$$

4.4.2 The Process of Cluster Head Election

4.4.2.1 Trust Value Initialization

When the network is set-up, there is no historical interaction behavior for reference, and all nodes are set to the same trust value in the initial phase. The values range from zero to one and the default value is one. After updating the communication statistics, the trust value of malicious nodes inserted by attackers gradually decrease from one, where the rate of decline depends on the parameters of the model. The trust value of normal nodes gradually approaches one and becomes steady after short-term fluctuations.

4.4.2.2 Updating Trust Value

At the beginning of each cluster head selection phase, every node updates the trust value of its neighboring nodes. For instance, **Fig. 6** shows the process to update $T_{i,j}^{new}$, the trust value of node i for node j:

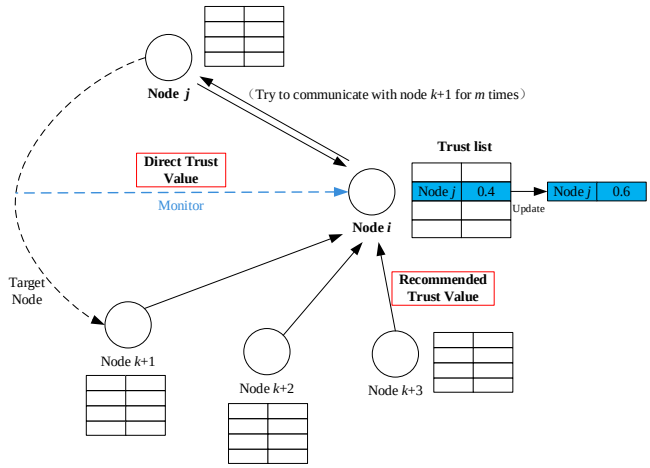


Fig. 6. Schematic diagram of the trust value update process

2.1) Node i retrieves the historical trust value of node j stored in the trust list.

2.2) Node i sends m data packets with $k+1$ target nodes to j, and saves the packets in its own buffer. The setting of m depends on the security level requirement. This value is then compared with the monitored routing packet broadcasted by node j. If the packets are in accordance, the number of normal behaviors are increased by one, and the buffer area i is cleared. If the packets stay for too long in the buffer or do not match, the number of abnormal behaviors of the node is increased by one. Finally, the correct direct trust value is calculated in conjunction with 2.1.

2.3) Other neighbor nodes send their recommended trust values of node j to node i.

2.4) Based on the above information, node i calculates the comprehensive trust value of node j and modifies the record in the trust list.

4.4.2.3 Formation of Secure Cluster Head List

Through the base station or the cluster head node of the previous step, according to the trust list constructed previously, a selection based on the trust value is implemented. Drones that have been is subjected to external attacks and turned into malicious nodes are filtered and eliminated in this step, and the security cluster head list is thus obtained.

3.1) Set the security threshold S of the network, $S \in (0,1)$. All nodes with trust values below threshold S are switched to the pending state and no longer participate in the election of the cluster head node, nor are they used for multi-hop communication.

3.2) Calculate Ave, the mean trust value of all nodes in each cluster. Set the node with trust value $T_{i,j}^{new} > Ave$ as a secure candidate node and store it in the list of candidate secure cluster heads. Send it to the base station to update the routing state database.

4.4.2.4 Comprehensive Determination of Cluster Head Node

After obtaining the secure cluster head list, the algorithm considers the residual energy ratio, relative mobility, and relative node degree to comprehensively calculate the security weights of the node. The node with largest weight is selected as cluster head node.

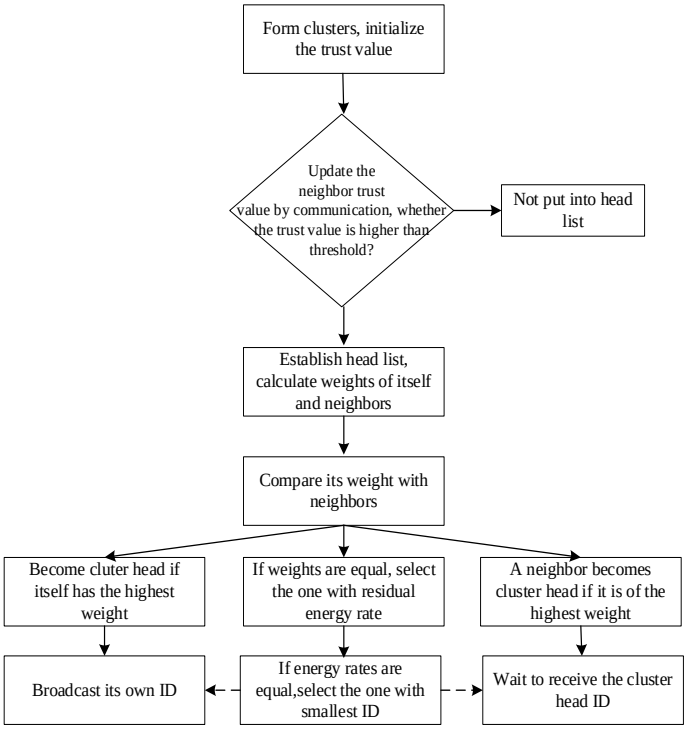


Fig. 7. Cluster head selection algorithm

The weight W_{vi} of node vi is calculated as follows:

$$W_{vi} = w_1.D_{vi} + w_2.E_{vi} + w_3.M_{vi} \tag{21}$$

Where $w_1+w_2+w_3=1$, and D_{vi}, E_{vi} , and M_{vi} represent the relative node degree, residual energy ratio and relative mobility, respectively.

During cluster head selection, the security cluster head list is first formed according to trust value, and the node with the largest weight in the list is designated as the cluster head. If two nodes have the same weight, the one with the largest residual energy ratio is chosen as cluster head. If the residual energy ratio of the two nodes is also identical, the one with the smallest *ID* is chosen as cluster head [33]. The cluster head selection algorithm is presented in **Fig. 7**.

4.5 Maintaining and Updating Cluster Heads

While clustering the network, violent node movement results in frequent variation of the cluster, which may in turn lead to the cluster head being updated and network structure being reconfigured. This incurs a large routing overhead and increases the computational load. A reasonable cluster maintenance mechanism is thus necessary to minimize clustering overhead and maintain cluster stability. The cluster maintenance mechanism features the following three rules.

4.5.1 Exiting of Old Node

If a member node does not receive the HELLO message containing cluster head *ID* information, or if it does not receive the HELLO message periodically broadcast by this node, it can be inferred that the node is not in the range of maximum communication of the cluster head, and should be removed. Moreover, according to the periodic assessment of the comprehensive trust value, nodes with trust values lower than threshold are determined to be malicious, and should be deleted from the cluster. To ensure that an exiting node cannot obtain the message after being removed is to ensure the backward security of the information. The cluster head needs to update the cluster key by generating a new key and encrypting it with the session key shared with each member and unicast to each member node.

4.5.2 Adding a Node

The new nodes includes those newly added to the network and re-entering from other clusters because of distance. A new node attempting to join a cluster, does not influence the status of the cluster head, even if it yields better performance [34]. Furthermore, to prevent a new node from obtaining messages before its entrance and guarantee the forward security of the information, once a new node joins the cluster, the cluster head needs to update the cluster key by generating a new key that is encrypted by the session key shared by each member (including the new node), and unicast to each member node.

4.5.3 Changing Cluster Head

When the cluster head does not receive the HELLO message containing member *ID* information in the cluster, or if it exits the network for other reasons, it gives up its position as cluster head. At this time, members of the cluster are restored.

5. Simulation and Verification

In this section, we used the Python Spyder environment to simulate the initial clustering of static UAVs network and subsequently evaluate the trust values. We implemented a dynamic simulation by using MATLAB on the Python interface, and finally used Python scripts to measure the indicators of statistical performance. The experimental hardware platform was the Intel Core I7-5700 workstation.

5.1 Simulation and Analysis of Improved K-Means++ Initial Clustering Algorithm

We simulated a network of 150 nodes in a three-dimensional space of 6 km × 6 km × 1 km using a Python script. Assuming that the maximum communication distance was 1000 m, the ratio of intra-cluster bandwidth to inter-cluster bandwidth was 1:2. The optimal number of clusters calculated in this case was seven. The results are shown in **Fig. 8**. The free node threshold T was set to 20 s, and the node with the asterisk arrow is the free node obtained by iteration.

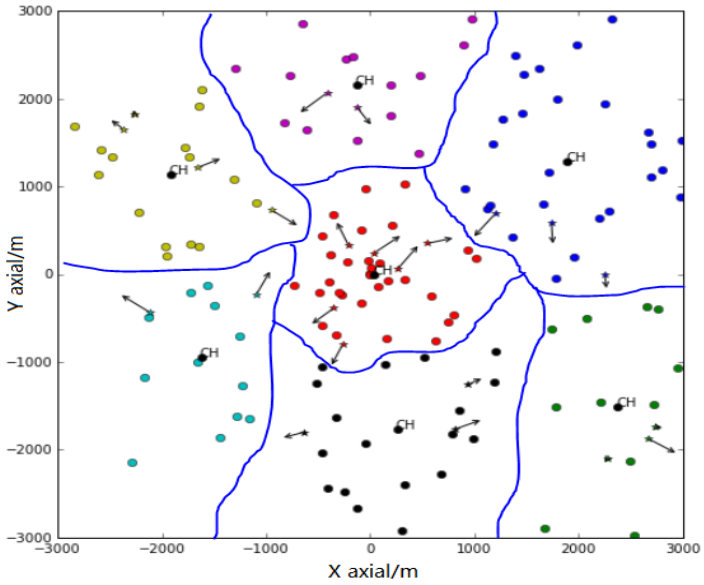


Fig. 8. Schematic diagram of initial clustering based on improved k-means++

Through the analysis of the results of the simulation of the initial clustering, we found that the improved k-means++ algorithm had the following three advantages: (1) The structures of the seven clusters were reasonable and evenly divided, which facilitated the subsequent maintenance and management of the cluster structure. (2) The numbers of object nodes in each cluster were relatively close to one another, almost all in the interval of 15-25. This distribution of nodes led to continued uniform utilization of the bandwidth and ensured the stability of the cluster structure under dynamic changes. (3) Only a minority of nodes were free, and most were in the edge region of each cluster. Thus, the complexity of the clustering algorithm was reduced to some extent, and the efficiency of clustering thus improved.

The statistics of the relationship between the free threshold T and the total number of free nodes are shown in **Fig. 9**. At a fixed free threshold, the higher the average speed of the nodes, the greater the total number of free nodes because this made the network more dynamic, and

increased the uncertainty of node ownership. This result indicates that the improved k-means++ algorithm can effectively reflect dynamic changes in UAVs network clustering.

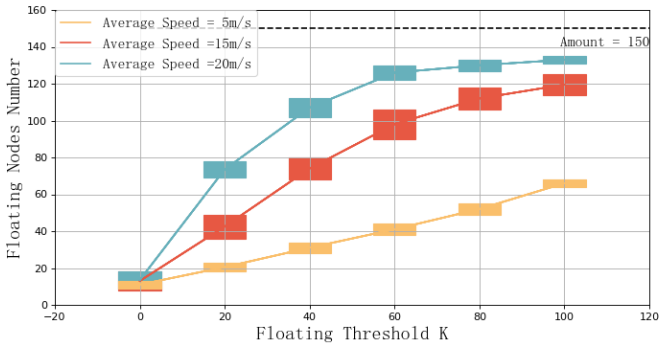


Fig. 9. Comparison in terms of free threshold *K* and total number of free nodes

A comparison between the improved algorithm and the k-means++ algorithm is shown in **Fig. 10**. The improved algorithm had a shorter convergence time for the same network communication bandwidth ratio. At a bandwidth ratio of 1:2, for example, the convergence time of the improved algorithm was 0.65 s and that of the k-means++ algorithm was 0.95 s. In case of an identical number of clusters and convergence time, the bandwidth ratio of the improved algorithm was higher, and consequently the bandwidth requirement for communication within the cluster was lower in the cluster communication process. For example, if the number of clusters was 10 and the convergence time was 0.75 s, the bandwidth ratio required by the improved algorithm was 1:2.2 and that required by the k-means++ algorithm was 1:1.6.

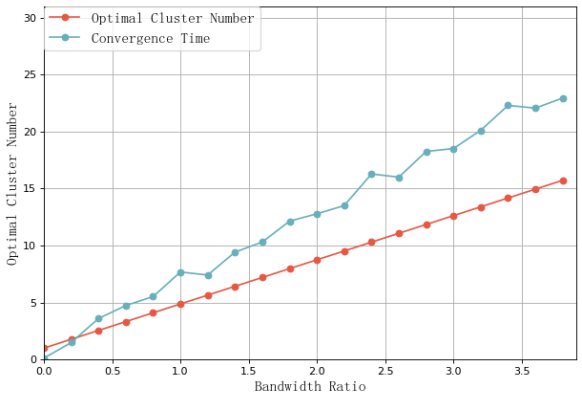


Fig. 10. Comparison in terms of convergence time.

5.2 Simulation and Analysis of Cluster Head Selection Algorithm

In this section, a simulation of cluster head selection based on the improved Bayesian model was performed on the initial clustering results of the improved k-means++ algorithm. Because the cluster head selection algorithm proposed in this paper is based on the premise of the existence of a malicious node, its generality is well preserved. We assume that the rate of

malicious nodes is R_{mal}, and malicious nodes randomly initiate selective forwarding or denial-of-service attacks.

We wrote a Python script to count the number of normal and abnormal behaviors between network nodes, and used the improved Bayesian model cluster head selection algorithm to monitor and update the initial cluster heads of the network. By setting the malicious node rate R_{mal} sequentially to 20%, 40% and 60%, and assuming that a malicious node initiated a route attack in the fourth round, we obtained the relationship between the comprehensive trust value of the malicious node and the number of communication rounds as shown in **Fig. 11**.

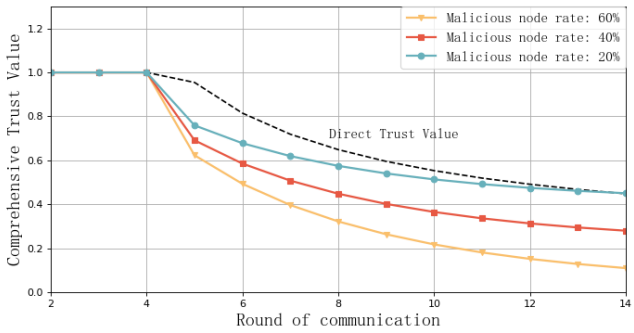


Fig. 11. Comparison of changes in the comprehensive trust value of malicious nodes (different rates of malicious nodes).

Fig. 11 shows that starting from the fourth round, the higher the rate of malicious nodes, the more quickly the Bayesian model that comprehensively recommends the trust value could reduce the total trust value of the malicious node. This model thus more promptly identified and eliminated malicious nodes, and prevented the capture of the cluster head.

To weaken dependence on the historical trust values of nodes, a forgetting factor ω is introduced to the improved Bayesian model proposed in this paper to avoid malicious nodes from being selected as cluster heads by accumulating a higher historical trust value. The forgetting factor is a function of the number of communication rounds n. To compare how the different forgetting factor configurations affect the calculation of the integrated trust value, different forgetting factor functions $\omega(n)$ were set to calculate the comprehensive trust value at a malicious node rate of R_{mal} = 40%. Because the forgetting factor is a function of the number of rounds, the average function value was used as the comparison parameter. The relationship between the comprehensive trust value and the number of communication rounds under different averages is shown in **Fig. 12**.

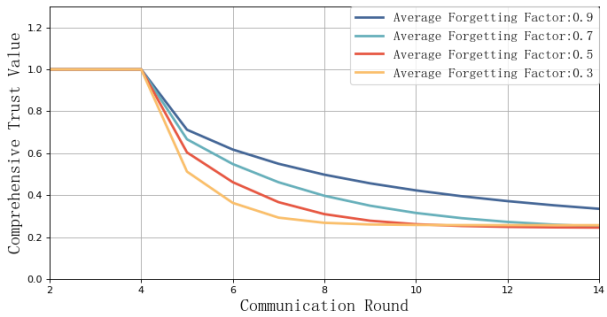
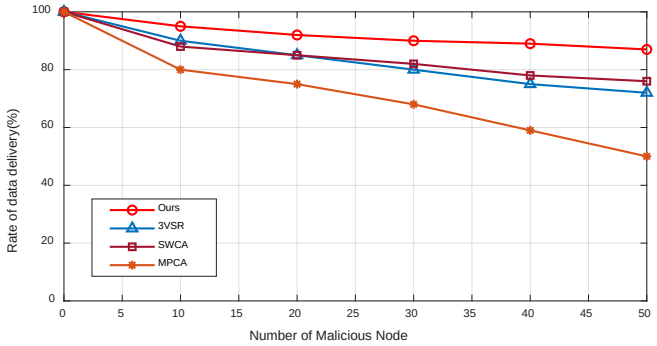


Fig. 12. Comparison of variations in the comprehensive trust value of malicious nodes (different average forgetting factors)

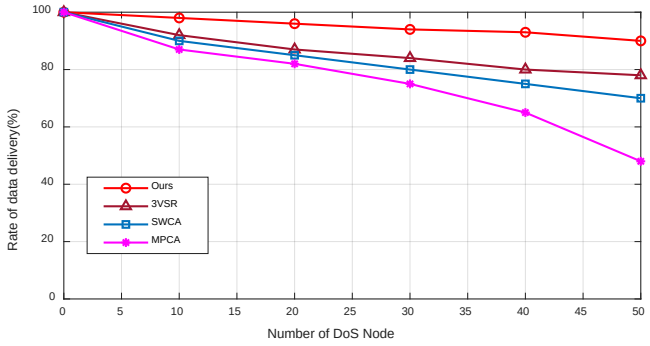
As shown in **Fig. 12**, after four rounds of attacks, the lower average value of the forgetting factor decreased with the comprehensive trust value. Note also that, because to the Bayesian trust model's requirement for the reference to the historical trust value as basis for identifying a malicious node, in case of a low average forgetting factor, such situations such as communication interference can be easily mistaken as a malicious node attack, which reduces the accuracy of the model in identifying malicious behavior. In the case of a relatively high average forgetting factor, the comprehensive trust value decreases more quickly, which helps better identify the attack behavior of malicious nodes in the given round of communication. Therefore, for the integrated drone network cluster head selection algorithm, a higher average value of the forgetting factor is necessary.

5.3 Performance Comparison

The security and stability of the network is also an important criterion to assess the advantages and disadvantages of a clustering algorithm. In this section, the cluster head selection algorithm based on the improved Bayesian model proposed is compared with MPCA[20], SWCA[22] and 3VSR[24] through simulation analysis. We assumed that the number of malicious nodes is 0-50, and the application communication data monitoring module was applied to collect the data delivery rate of all nodes. The total time for the simulation was 500 s, and the results are shown in **Fig. 13**.



(a)



(b)

Fig. 13. Comparison of changes in rate of data delivery under different attack pattern: (a) Malicious node attack. (b) DoS node attack.

Fig. 13a shows that due to selective forwarding attacks by malicious nodes on the network, the data delivery rate began to decline rapidly. In case a malicious node is selected as cluster head, the data delivery rate drops more quickly. Since our election algorithm has better ability to detect and tackle malicious attacks it outperform other algorithms. Similarly, the rate of data delivery is higher in our algorithm under DoS node attack pattern as can be seen in **Fig. 13b**. Since our algorithm uses improved bayesian trust model to discover and isolated DoS nodes in time, then the rate of data delivery can be maintained above 90%.

Computational cost is also an important factor that needs to be considered for clustering algorithms. Comparing SWCA[22] and 3VSR[24] with the proposed cluster head selection algorithm, the relationship between the average time needed to eliminate malicious nodes is shown in **Fig. 14**. To ensure fairness, we assumed that the malicious node rate R_{mal} was 40% for the simulated UAVs network.

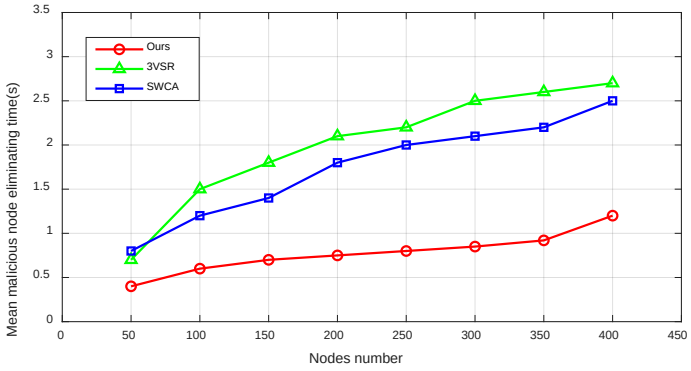


Fig. 14. Comparison of calculated time changes

As shown in the **Fig. 14**, the cluster head selection algorithm proposed in this paper cost less than other algorithms in terms of the time needed to locate and eliminate malicious nodes. For a small-scale network, it can save approximately 50.6% time. As the number of nodes increased, the culling time needed by 3VSR and SWCA increased at a higher rate. Thus, the proposed algorithm is more efficient. For example, in a network with a 350 nodes, the average time taken by the clustering algorithm proposed in this paper to identify malicious nodes was only 1/3 of the time taken by SWCA.

6. Conclusion

In this paper, focusing on highly dynamic UAVs network, we proposed an improved k-means++ clustering algorithm for the cluster head selection process in hierarchical UAVs network. We developed a trust clustering algorithm that could eliminate malicious nodes, and was suitable for UAVs network with limited computing power by enhancing the Bayesian model. The results of simulation indicated that the improved k-means++ clustering algorithm could be applied to quickly moving UAVs network, providing a stable initial clustering structure for the subsequent cluster head election and update. The secure clustering algorithm based on trust value efficiently identified malicious nodes and excluded them from cluster selection process to enhance the probability that a normal drone node was selected as cluster head. The results of comparison with the other three existing approaches showed that the cluster head selection algorithm based on the improved Bayesian model required fewer computational resources, and had higher network security and stability.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under No.61601467, the Project of Civil aviation safety capacity under Nos. PESA2018082, PESA2019074, and the fundamental research funds for the central universities under No. 3122018C036.

References

[1] Y. Qu, F. Zhang, X. Wu, and B. Xiao. “Cooperative geometric localization for a ground target based on the relative distances by multiple UAVs,” *Science China Information Sciences*, vol. 62, no. 1, pp. 10204:1-10204:10, 2019. Article (CrossRef Link)

[2] Y. Yuan, L. Cheng, Z. Wang, and C. Sun, “Position tracking and attitude control for quadrotors via active disturbance rejection control method,” *Science China Information Sciences*, vol. 62, no. 1, pp. 10201:1-10201:10, 2019. Article (CrossRef Link)

[3] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, “Taking drones to the next level: Cooperative distributed unmanned aerial vehicular networks for small and mini drones,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 73-82, 2017. Article (CrossRef Link)

[4] W. Zafar, and B. M. Khan, “Flying Ad-hoc networks: technological and social implications,” *IEEE Technology & Society Magazine*, vol. 35, no. 2, pp. 67-74, 2016. Article (CrossRef Link)

[5] C. Cooper, D. Frankline, M. Ros, F. Safaei, M. Abolhasan, “A comparative survey of VANET clustering techniques,” *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 657-681, 2017. Article (CrossRef Link)

[6] A. Farhan, R. Ali, K. Muhammad, et al., “Energy aware cluster-based routing in flying ad-hoc networks,” *Sensors*, vol. 18, no. 5, pp.1413-1428, 2018. Article (CrossRef Link)

[7] H. Shakhatreh et al., “Unmanned Aerial Vehicles: A survey on civil applications and key research challenges,” *IEEE Access*, vol. 7, pp. 48572-48634, 2019. Article (CrossRef Link)

[8] S. Hagerman, A. Andrews, and S. Oakes, “Security testing of an unmanned aerial vehicle (UAV),” in *Proc. of IEEE Cybersecurity Symposium* (*CYBER-SEC*), pp.26-31, 2016. Article (CrossRef Link)

[9] A. Gupta, and M. Hussain, “Distributed cooperative algorithm to mitigate hello flood attack in cognitive radio ad hoc networks (CRAHNs),” in *Proc. of the First Int. Conf. Computation. Intelligence and Informatics* (*AISC*), vol. 507, pp.255-263, 2016. Article (CrossRef Link)

[10] A. K. Sharma et al., "Sybil attack prevention and detection in vehicular ad hoc network," in *Proc. of Int. Conf. Computing, Communication and Automation* (*ICCCA*). *IEEE*, pp. 594-599, 2016. Article (CrossRef Link)
[11] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile Ad hoc networks," *Clustering Computing*, vol. 5, no. 2, pp.193-204, 2002. Article (CrossRef Link)
[12] A. H. Hussein, A. O. A. Salem, and S. A. Yousef, "A flexible weighted clustering algorithm based on battery power for mobile ad hoc networks," in *Proc. of IEEE Int. Symp. Ind. Electron.* (*ISIE*), pp. 2102-2107, 2008. Article (CrossRef Link)
[13] S. Kaur, R. C. Gangwar, and R. Singh, "A strength based energy efficient algorithmic approach in MANET," in *Proc. of Int. Conf. Soft Comput. Techn. Implement.* (*ICSCTI*), pp. 141-145, 2015. Article (CrossRef Link)
[14] M. Gerla, and T. C.Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Network*, vol. 1, no. 3, pp. 255-265, 1995. Article (CrossRef Link)
[15] D. Gavalas, G. Pantziou, C. Konstantopoulos, and B. Mamalis, "Lowest-ID with adaptive ID reassignment: A novel mobile ad-hoc networks clustering algorithm," in *Proc. of 1st Int. Symp. Wireless Pervasive Comput., Phuket, Thailand, Jan.*, p. 1-5, 2006. Article (CrossRef Link)
[16] M. Ni, Z. Zhong, and D. Zhao, "MPBC: a mobility prediction-based clustering scheme for Ad hoc networks," *IEEE Trans on Vehicular Technology*, vol. 60, no. 9, pp. 4549-4559, 2011. Article (CrossRef Link)
[17] F. Aftab, Z. Zhang, and A. Ahmad, "Self-organization based clustering in MANETs using zone based group mobility," *IEEE Access*, vol. 5, pp. 27464-27476, 2017. Article (CrossRef Link)
[18] C. Tselikis et al., "Empirical study of clustering algorithms for wireless ad-hoc networks," in *Proc. of Int. Conf. Systems Signals and Image Processing. New York: IEEE Press, Jun*, pp. 1-6, 2009. Article (CrossRef Link)
[19] N. Shi and X. Luo, "A novel cluster-based location-aided routing protocol for UAV fleet networks," *Int. J. Digit. Content Technol. Appl.*, vol. 6, no. 18, pp. 376-383, 2012. Article (CrossRef Link)
[20] C. Zang and S. Zang, "Mobility prediction clustering algorithm for UAV networking," in *Proc. of IEEE GLOBECOM Workshops* (*GC Workshops*), *Houston, TX, USA*, pp. 1158-1161, 2011. Article (CrossRef Link)
[21] M. Devaprakas, and F. Mazen, "A framework for detection of sensor attacks on small unmanned aircraft systems," in *Proc. of IEEE Int. Conf. Unmanned Aircraft Systems* (*ICUAS*), pp. 1189-1198, 2017. Article (CrossRef Link)
[22] Y. Q. Ma, and X. Y. Li, "Adaptive security weighted clustering algorithm for ad-hoc networks," *Computer Engineering and Design*, vol. 9, no. 10, pp. 3346-3350, 2014. Article (CrossRef Link)
[23] S. Jangra, and N. Goel, "e-ARAN: Enhanced authenticated routing for ad hoc networks to handle selfish nodes," in *Proc. of IEEE Int. Conf. Advances In Engineering, Science And Management* (*ICAESM-2012*), pp. 144-149, 2012. Article (CrossRef Link)
[24] M. Sohail, and L. M. Wang, "3VSR: Three valued secure routing for vehicular ad-hoc networks using sensing logic in adversarial environment," *Sensors*, vol. 18, no. 3, pp. 1-24, 2018. Article (CrossRef Link)
[25] M. Li, D. Xu, D. M. Zhang and T. Zhang, "A streaming algorithm for k-means with approximate coreset," *Asia-Pacific Journal of Operational Research (APJOR)*, vol. 36, no. 1, pp.1-18, 2019. Article (CrossRef Link)
[26] E. T. Khalaf, M. N. Mohammad, K. Moorthy, "Robust partitioning and indexing for iris biometric database based on local features," *IET Biometrics*, vol. 7, no. 6, pp.589-597, 2018. Article (CrossRef Link)
[27] R. Ismail, C. Boyd, A. Josang, and S. Russell, "A security architecture for reputation systems," in *Proc. of Int. Conf. E-Commerce and Web Technologies, LNCS*, vol. 2738, pp.176-185, 2003. Article (CrossRef Link)
[28] S. Ganeriwal, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 3, Article. 15, 2008. Article (CrossRef Link)

[29] J. Z. Feng, D. Q. Xiao, and B. Yang, "Reputation system for wireless sensor networks based on β distribution," *Journal of Computer Applications*, vol. 27, no. 1, pp. 111-113, 2007. Article (CrossRef Link)
[30] A. R. Vetrella, and F. Causa, "Multi-UAV Carrier Phase Differential GPS and Vision-based Sensing for High Accuracy Attitude Estimation," *Journal of intelligent & robotic systems*, vol. 93, pp. 245-260, 2019. Article (CrossRef Link)
[31] R. Amorim et al., "Radio channel modelling for UAV communication over cellular networks," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 514-517, 2017. Article (CrossRef Link)
[32] K. Sanzgiri, D. Laflamme, B. Dahill, et al., "Authenticated routing for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598-610, 2006. Article (CrossRef Link)
[33] A. Bentaleb, S. Harous, and A. Boubetra, "A weight based clustering scheme for mobile ad hoc networks," in *Proc. of Int. Conf. advances in mobile computing & multimedia* (*MoMM*), pp. 161-166, 2013. Article (CrossRef Link)
[34] S. Singhal, and A. K. Daniel, "Cluster head selection protocol under node degree, competence level and goodness factor for mobile ad-hoc network using AI technique," in *Proc. of Int. Conf. Advanced Computing & Communication Technologies*, pp. 415-420, 2014. Article (CrossRef Link)

Jingxian ZHOU received the B.S. degree in mathematics from Xuchang University in 2004, the M.S. degree in Mathematics from Zhengzhou University in 2010, and the Ph.D. degree in cryptography from Beijing University of Posts and Telecommunications, in 2013. Now, he is an associate research fellow with Information Security Evaluation Center, Civil Aviation University of China. His research interests are security authentication protocol, data privacy protection and security architecture for the Internet of Things.

Zengqi Wang received a Bachelor of Science degree from Henan Normal University, Xinxiang, China. She is currently pursuing master degree in the School of Computer Science and Technology, Civil Aviation University of China. Her current research interests include threat intelligence analysis and data processing.