sensors

MDPI

Article

DLSMR: Deep Learning-Based Secure Multicast Routing Protocol against Wormhole Attack in Flying Ad Hoc Networks with Cell-Free Massive Multiple-Input Multiple-Output †

Yushintia Pramitarini [1], Ridho Hendra Yoga Perdana [1], Kyusung Shim [2] and Beongku An [3,*]

[1] Departement of Software and Communications Engineering in Graduate School, Hongik University, Sejong City 30016, Republic of Korea; yushintia@mail.hongik.ac.kr (Y.P.); hendra@mail.hongik.ac.kr (R.H.Y.P.)
[2] School of Computer Engineering & Applied Mathematics, Hankyong National University, Anseong City 17579, Republic of Korea; kyusung.shim@hknu.ac.kr
[3] Departement of Software and Communications Engineering, Hongik University, Sejong City 30016, Republic of Korea
* Correspondence: beongku@hongik.ac.kr
† This paper is an extended version of our paper published in Pramitarini, Y.; Perdana, R.H.Y.; Shim, K.; An, B. Particle Swarm Optimization-Based Clustering Algorithm to Support QoS Routing Protocol in Flying Ad-Hoc Networks with CF-mMIMO. In Proceedings of the 11th International Conference on Green and Human Information Technology (ICGHIT 2023), Bangkok, Thailand, 31 January–2 February 2023; pp. 214–219.

Abstract: The network area is extended from ground to air. In order to efficiently manage various kinds of nodes, new network paradigms are needed such as cell-free massive multiple-input multiple-output (CF-mMIMO). Additionally, security is also considered as one of the important quality-of-services (QoS) parameters in future networks. Thus, in this paper, we propose a novel deep learning-based secure multicast routing protocol (DLSMR) in flying ad hoc networks (FANETs) with cell-free massive MIMO (CF-mMIMO). We consider the problem of wormhole attacks in the multicast routing process. To tackle this problem, we propose the DLSMR protocol, which utilizes a deep learning (DL) approach to predict the secure and unsecured route based on node ID, distance, destination sequence, hop count, and energy to avoid wormhole attacks. This work also addresses key concerns in FANETs such as security, scalability, and stability. The main contributions of this paper are as follows: (1) We propose a deep learning-based secure multicast routing protocol (DLSMR) to establish a high-stability multicast tree and improve security performance against wormhole attacks. In more detail, the DLSMR protocol predicts whether the route is secure based on network information such as node ID, distance, destination sequence, hop count, and remaining energy or not. (2) To improve the node connectivity and manage multicast members, we propose a top-down particle swarm optimization-based clustering (TD-PSO) protocol to maximize the cost function considering node degree, cosine similarity, cosine distance, and cluster head energy to guarantee convergence to the global optima. Thus, the TD-PSO protocol provides more strong connectivity. (3) Performance evaluations verify the proposed routing protocol establishes a secure route by avoiding wormhole attacks as well as by providing strong connectivity. The TD-PSO clustering supports connectivity to enhance network performance. In addition, we exploit the impact of the mobility model on the network metrics such as packet delivery ratio, routing delay, control overhead, packet loss ratio, and number of packet losses.

Keywords: CF-mMIMO; clustering; deep learning; flying ad hoc networks; secure multicast routing; security; wormhole attack

Citation: Pramitarini, Y.; Perdana, R.H.Y.; Shim, K.; An, B. DLSMR: Deep Learning-Based Secure Multicast Routing Protocol against Wormhole Attack in Flying Ad Hoc Networks with Cell-Free Massive Multiple-Input Multiple-Output. *Sensors* **2023**, *23*, 7960. https://doi.org/10.3390/s23187960

Academic Editor: Yingjie Jay Guo

Received: 14 August 2023
Revised: 11 September 2023
Accepted: 15 September 2023
Published: 18 September 2023

1. Introduction

Unmanned aerial vehicle (UAV) communications play an important role in modern network infrastructure, particularly in flying ad hoc networks (FANETs) integrated

with cell-free massive multiple-input multiple-output (CF-mMIMO) [1,2]. These modern communications setups rely on effective routing methods which have a direct impact on performance and efficiency. Recently, security has emerged as a significant concern, warranting special attention to prevent potential breaches [3–8].

Given its airborne position, the UAV can cover large areas of the ground, providing superior network coverage over traditional ground-mounted access points (APs) [9,10]. Therefore, clustering or member management is very important, ensuring optimal utilization of the UAV's wide range of capabilities and maintaining network efficiency [11,12]. One of the major security threats to these networks is wormhole attacks which are characterized by capturing enemy nodes and tunnelling packets to other locations on the network [13]. This misleading tactic breaks regular data transmission pathways and modifies network topology causing substantial network performance and reliability disruptions. Consequently, understanding these attacks and developing robust defenses against them is essential to ensure secure and reliable UAV communications.

A multicast routing protocol is a key technique in ensuring reliable data communication to multiple destinations [14]. The multicast routing protocol can select the best next node to establish the optimal route from the source to the multiple destinations [15]. Particularly in the FANET environment, UAVs move dynamically so that there can be a risk of broken connectivity and attack from eavesdroppers [16]. The authors of [16] proposed a method to detect a black hole in a short time. The authors of [17] studied a hybrid authentication scheme with a digital signature to improve the security performance in the UAV and ground node against wormhole attacks. However, the proposed method has a high processing time to authenticate the secure node.

In this context, deep learning models present an intriguing opportunity [18,19]. When the deep learning models are applied to FANETs, these models can offer sophisticated route verification mechanisms. These mechanisms predict the next secure node in the route, ensuring that the chosen path is free from wormholes before initiating data transmission. The use of deep learning to improve security in UAV communications forms the basis of this paper.

1.1. Related Works

Some works have been studied to improve network stability in FANETs. The authors of [20] proposed bio-inspired clustering for the FANET environment. They develop a bio-inspired clustering protocol to improve energy efficiency (EE) and manage UAV mobility. However, it does not consider safety considerations to reduce collisions and enhance stability. In [21], the authors proposed an intelligent cluster routing scheme (CRSF) to address UAV communication issues. Additionally, the CRSF can improve the stability performance in UAV communication. Nevertheless, the CRSF has a high control overhead during the clustering process. The authors of [22] used clustering as a solution to improve network stability. In addition, they also studied EE localization and clustering for UAV wildfire monitoring. The authors of [23] proposed a down–up particle swarm optimization to improve the scalability of the FANET system. However, this work only considered the optimization approach to improve stability performance.

Regarding secure multicast routing protocols, improved security performance in the FANET system was studied in [24]. Scalable and predictive routing (SP-GMRF) was proposed to predict the next node. Based on node position, the SP-GMRF provided the shortest distance to each destination node. However, SP-GMRF does not consider the clustering protocol, so it has a high control overhead. The authors of [25] proposed a distributed tree-based routing (DPTR) for FANETs, forming a network that avoids fragmentation. In this approach, a tree-like structure connects all the UAVs in the networks. Nevertheless, the routing delay is high since every node calculates the entire routing tree. In [26], the authors studied swarm EE multicast routing (SEMRP) for UAV swarms for UAVs in group formation. The SEMRP can reduce packet loss and packet re-transmission, and delay performance. However, the SEMRP has a high control overhead during the routing process.

The authors of [27] proposed a credible neighbor discovery (CRFNE) algorithm to shield messages against wormhole attacks in wireless sensor networks (WSNs). The CRFNE can detect wormhole attacks during route discovery. However, the CRFNE has a high processing time due to the greater number of steps to calculate the wormhole threshold to detect the wormhole node. Based on the AODV routing protocol, the authors of [28] designed a wormhole-immune routing protocol. However, the DAWA protocol has a high control overhead because the DAWA protocol broadcasts the packet discovery to all nodes in the network. Meanwhile, the authors of [29] proposed a hybrid price auction (HPA)-based routing protocol to avoid sinkhole attacks. In addition to that, the HPA protocol can improve the security, routing delay, and scalability performance. Nevertheless, they only consider sinkhole attacks in the network. On the other hand, the DL is applied to improve end-to-end (E2E) delay in aeronautical ad hoc networks (AANETs) [30]. The authors of [30] explored a DL-based multicast routing protocol for mapping the local geographic information observed by the forwarding node into the information required for determining the optimal next hop. However, they have not considered wormhole attacks in the networks. Additionally, the authors of [31] studied reinforcement learning-assisted secure routing to minimize delays and map geographic information using Q learning. Thus, the proposed protocol can select the secure route efficiently. However, they did not mention which kind of attack was considered.

The above-mentioned works partially addressed the raised network issues such as secure multicast routing, network scalability, and deep learning approaches. In more detail, some studies have examined the effect of DL on secure routing [30,31]; others have studied the design of secure multicast routing against eavesdropped attack [24,30]; very few works have explored the effect of DL techniques in the secure multicast routing protocol specifically against wormhole attacks on security, scalability, and stability performance, which is a critical performance metric in 5G wireless networks.

1.2. Motivation and Contributions

Based on the abovementioned, secure and multicast routing in FANETs have been studied considerably. However, there is still a gap in the literature regarding secure multicast routing against wormhole attacks in FANETs with CF mMIMO networks. To fill this gap, we proposed a DL-based secure multicast routing (DLSMR) protocol against wormhole attacks that allows the protocol to predict the next secure node within a short time. Furthermore, the combination with TD-PSO-based clustering allows the protocol to find the optimal cluster header. The goal is to improve the security, scalability and stability performance, and achieve an optimal secure route that meets the requirements of 5G and beyond 5G wireless networks. The main contributions of this article can be summarized as follows:

- We propose a deep learning-based secure multicast routing (DLSMR) protocol to establish a high-stability multicast tree to avoid wormhole attacks in FANETs with CF-mMIMO. Specifically, we utilize a deep learning model to predict whether the next node is a wormhole or not. Additionally, we use various network parameters to establish routes that support more strong connectivity.
- We design a novel top-down particle swarm optimization (TD-PSO)-based clustering protocol in FANETs to reduce control overhead and improve route connectivity. The proposed TD-PSO, considering the node position, velocity, direction, and remaining energy, forms a cluster to optimize the cost function by combining the remaining energy weight, cosine similarity, cosine distance, and node degree. This strategic approach leads to electing cluster heads. Furthermore, to ensure communication continuity between cluster heads when the subsequent ones fall outside of the transmission range, our TD-PSO protocol also designates gateway nodes.
- The performance evaluations show that the proposed DLSMR with TD-PSO protocol can establish a more robust route against wormhole attacks than the benchmark protocol. Additionally, the proposed TD-PSO clustering supports stronger connec-

tivity as clustering changes the network topology hierarchically. In addition to that, we also compare the proposed protocol under two different mobility models (reference point group mobility and random waypoint) to show the effectiveness of the proposed protocol.

The remainder of the paper is organized as follows. Section 2 introduces the overview of wormhole attack. Section 3 introduces the particle swarm optimization theory. Section 4 introduces the proposed routing protocol that consists of the basic concept of the proposed routing protocol, the proposed TD-PSO-based clustering protocol, the proposed DLSMR, and proposed deep-learning design. Section 5 presents the performance evaluation that consists of simulation environments and parameters, performance metrics, and numerical results. Section 6 concludes the paper. For ease of presentation, Abbreviations summarizes the main abbreviations used in this paper.

2. Overview of Wormhole Attack

A wormhole attack is a type of network security threat that affects wireless networks, including flying ad hoc networks (FANETs). In particular, FANETs consist of a group of unmanned aerial vehicles (UAVs) that communicate with each other to form a self-organizing and self-healing network. Due to their dynamic topology, limited resources, and open nature, FANETs are particularly vulnerable to various security threats, including wormhole attacks. In a wormhole attack, an attacker creates a tunnel between two malicious nodes, capturing packets at one end and replaying them at the other end almost instantly [32,33]. Wormhole attacks creates the illusion of a shorter and more efficient route between the malicious nodes. When any transmitted packet reaches one of the attacker nodes, that node forwards the packet to its distant malicious counterpart through legitimate intermediate nodes. Although these intermediate nodes are not directly involved in the communication, their resources get drained because of their unwitting participation in the wormhole attack.

In the multicast routing process, each destination node sends a join request to the cluster head (CH) within its cluster. The source node sends a multicast route request (MRREQ) via unicast to each CH to find multiple destination nodes. Upon receiving the MRREQ, the UAV gateway node (UGW) updates its routing table and rebroadcasts the MRREQ to neighboring nodes. When a CH with a relevant multicast group ID receives an MRREQ, it updates its routing table and sends a Route Reply (RREP) packet back to the source node via unicast. Wormhole attackers also forward MRREQ and RREP packets. They exploit this routing process by sending fake RREPs with significantly higher sequence numbers than normal nodes. This makes the attackers appear to offer the freshest route to the destination, even without consulting their routing table. As a result, other nodes update their routing tables based on this misleading information, causing severe disruptions in network functionality.

3. The Particle Swarm Optimization Theory

The particle swarm optimization (PSO) algorithm is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [34]. In this algorithm, a swarm of particles explores a multi-dimensional search space to find optimal solutions. Given an optimization function $f(X)$, where X is an n-dimensional random vector, these particles serve as candidate solutions for the optimization problem. Each particle i is characterized by its velocity $V_i = (v_{i1}, v_{i2}, ..., v_{ij})$ and position $P_i = (p_{i1}, p_{i2}, ..., p_{ij})$, $i = 1, 2, ..., q$, $j = 1, 2, ..., n$, and n and q represent the dimensions and swarm size, respectively. Each particle represents a candidate solution and searches for the global optimum in the problem space. To find the optimal solution, each particle moves towards the previous best position (*pbest*) and the global best position (*gbest*) in the cluster can be mathematically expressed as [35]

$$pbest(i,t) = \arg \min_{k=1,\cdots,t} [f(P_i(k))], i \in 1,2,\cdots,N_p,$$
$$gbest(t) = \arg \min_{\substack{i=1,\cdots,N_p \\ k=1,\cdots,t}} [f(P_i(k))], \quad (1)$$

where N_p denotes the total number of particles, t denotes the current iteration number, and f denotes the fitness function. In each generation, particles i adjusts its velocity V_i and position X_i according to the following formula [36]:

$$v_i(t+1) = wv_i(t) + c_1 r_1(t)(pbest(i,t) - P_i(t)) + c_2 r_2(t)(gbest(t) - P_i(t)),$$
$$P_i(t+1) = P_i(t) + v_i(t+1), \quad (2)$$

where w is the inertia weight used to balance the global exploration and local exploitation, r_1 and r_2 are uniformly distributed random variables within range $[0,1]$, and c_1 and c_2 are positive constant parameters called acceleration coefficients. As the algorithm progresses, particles share their best-known positions with some or all of the swarm. This collaborative sharing helps the guide of the group toward optimal solutions. The extent to which this division affects individual particles depends on the specific environmental topology used.

4. the Proposed Secure Multicast Routing Protocol: DLSMR

In this section, we describe the structural characteristics of our network, the specifications of the UAV and ground node, and the existence and implications of wormhole attacks within the UAV layer.

Our network consists of two types of nodes $\mathcal{K} = \{F_k\}_{k=1}^{K}$ represented by a set of flying nodes F and $\mathcal{L} = \{G_l\}_{l=1}^{L}$ represented by a set of ground nodes G. These nodes are organized into clusters, $\mathsf{CL}_c \triangleq \{c = 1,2,\cdots,C\}$, through a top-down methodology that enhances network scalability and performance. UAVs have a distinct set of characteristics that are crucial for our protocol. These flying nodes are mobile and have the ability to cover large ground areas, providing superior network coverage compared to traditional ground nodes. However, their movement speed, direction, and remaining energy must be accurately tracked to maintain network efficiency and stability. In contrast, ground nodes serve a vital purpose as points of connection. They receive, process, and transmit data to UAVs, playing an important role in establishing and maintaining the network's functionality. One of the major security threats in our setup is the presence of wormhole attacks within the UAV layer. These attacks involve malicious nodes that tunnel packets from one location to another within the network. This deceptive maneuver disrupts regular data transmission pathways and modifies network topology, causing significant disturbances to the network's performance and reliability. To counteract these challenges, we propose a novel DLSMR protocol and a TD-PSO-based clustering protocol. The DLSMR protocol is designed to predict secure routes and avoid wormhole nodes, thereby securing data transmission. Concurrently, the TD-PSO clustering protocol groups UAVs and ground nodes into clusters based on their positions, speed, direction, and remaining energy, providing stable connectivity and improving overall network performance. Through these techniques, we aim to build a secure, scalable, and efficient communication network that effectively counteracts wormhole attacks while satisfying the unique characteristics and needs of UAVs and ground nodes.

4.1. The Basic Concept of the Proposed Secure Multicast Routing and Clustering Protocol

In this section, we explain the basic concepts of our proposed secure multicast routing and clustering protocol. The process of establishing routes is divided into two steps, which are as follows:

- **Step 1 (Clustering Process):** Firstly, we consider the entire network as a single cluster, which is then recursively divided into smaller clusters based on the weight of the node degree, cosine similarity, cosine distance, and remaining energy. We identify flying nodes as potential cluster heads in this top-down clustering process as they inherently possess stronger capabilities within the cluster. Moreover, the flying node with the highest remaining energy is elected as the CH since it has to work harder than the member nodes as noted in [23]. A flying node that is not selected as a cluster head receives two or more cluster head information (CHI) packets from the CH, and then that node becomes an UGW. Otherwise, the ground nodes can only serve as cluster members. To select the optimal cluster head, the cluster member candidates use the PSO algorithm to optimize costs. Unlike the traditional clustering method of [23], we introduce a TD-PSO protocol to improve control overhead during cluster formation.
- **Step 2 (Multicast Routing Process):** Following the clustering process, each destination node that wants to receive data transmission sends a join request to the CH within the cluster. Then, a source node sends a MRREQ to each cluster head by unicast to find the multiple destination nodes. After that, CH will broadcast the MRREQ packet to each CH to find the multiple destination nodes. When the UGW receives the MRREQ packet, it updates its routing table and re-broadcasts the MRREQ packet to the neighbor nodes. When a CH who has a multicast group ID receives an MRREQ packet, they update the routing table and reply a RREP packet to the previous node by unicast. In addition to that, the wormhole attacker also forwards the MRREQ and RREP packets. In this work, we develop a novel DL framework to predict the secure next node while establishing the route from a source to the multicast group destinations. Based on the DL framework, we can train a DNN model to learn wormhole nodes' characteristics and distinguish them from legitimate ones. Each flying node in the network utilizes a deep learning framework to find the next secure nodes for multiple destinations. We consider the node ID, node's position, destination ID, destination sequence, hop count, and the remaining energy of the node as the trainable input parameter, and the secure and unsecured node ID, secure and unsecured status as output parameters of the DNN model. When a node receives a RREP packet, the contents of the relevant RREP packet will be used in the DL framework. After training, the DNN model produces secure and unsecured node ID and their status as output. For example, as we can see in Figure 1, when the CH receives two RREP packets which are from the neighbor nodes and the wormhole nodes, the proposed DLSMR protocol can establish a secure multicast route from $S-CH_1-UGW_{(1,4)}, CH_2$ and also $S-CH_1-UGW_{(1,4)}, CH_2, CH_3$ with a multicast tree to avoid the wormhole attack.

4.2. The Proposed Clustering Protocol: Top-down Particle Swarm Optimization (TD-PSO)

4.2.1. The Basic Concept of the TD-PSO

As shown in Figure 2, we design the top-down particle swarm optimization-based clustering protocol which ensures network connectivity and reduces the control overhead in FANETs. The proposed clustering protocol considers the problem of the join weight of the node degree, cosine similarity, cosine distance, and remaining energy to form clusters. Additionally, we use the higher remaining energy among the candidate nodes to select the cluster head. In this work, we assume only UAV can become the CH because it has the greatest resources compared to the ground node. Then, the selected CHs transmit packets through inter-cluster forwarding. In most cases, this top-down approach can ensure network connectivity and coverage. Unlike the bottom-up method, the top-down method can reduce the number of control overhead because the number of flying nodes (FN) is greater than the number of ground node (GN) nodes with CF-mMIMO properties. In order to simplify, we will refer to UAVs and ground users as nodes. The proposed TD-PSO protocol assumes that each node can know its location information by using the global positioning system (GPS). The following subsubsection will explain the proposed TD-PSO protocol in detail.

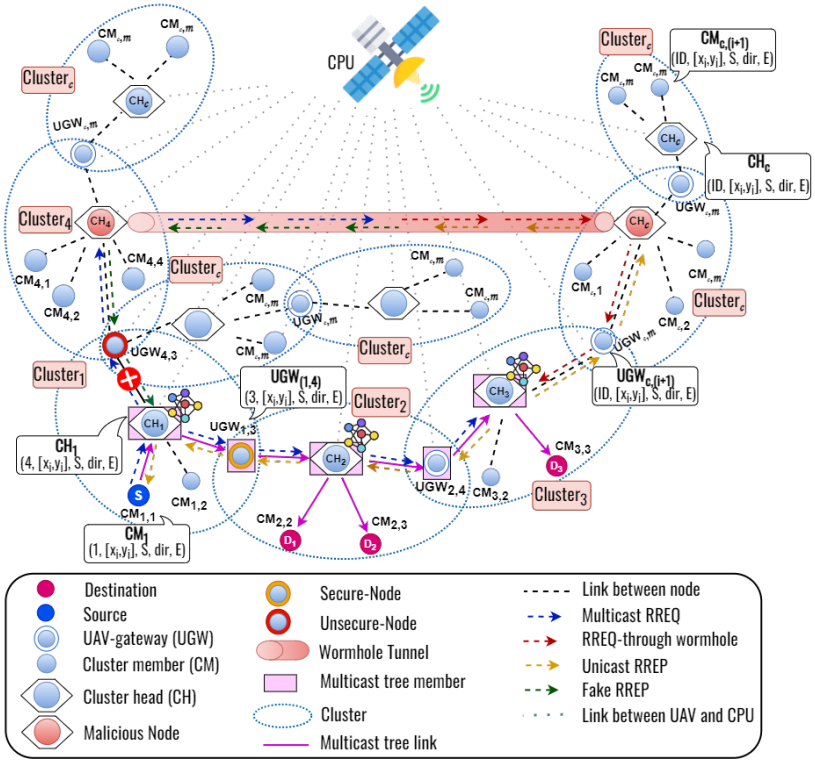


Figure 1. The basic concepts of the proposed DLSMR protocol.

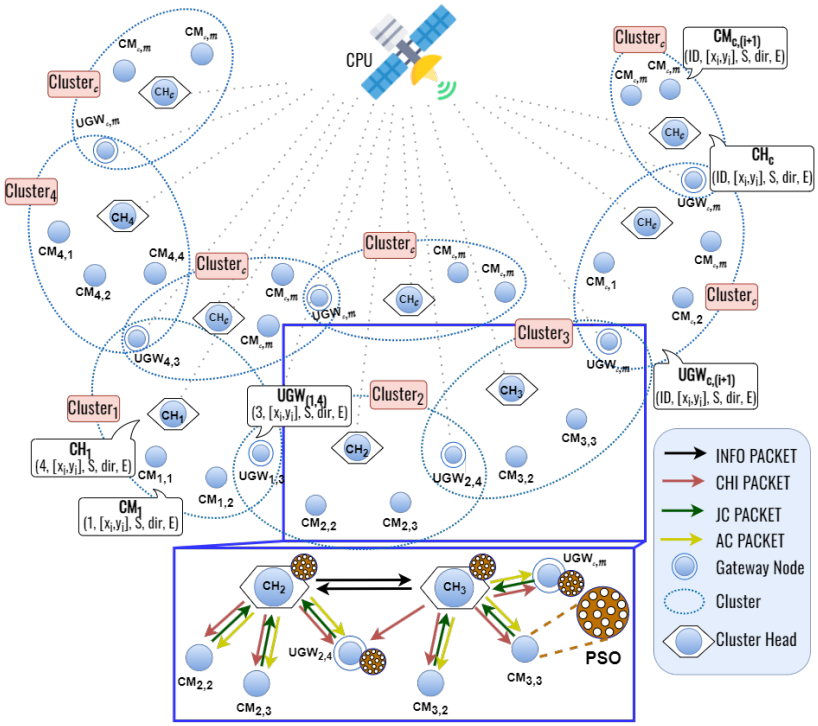


Figure 2. The basic concepts of the TD-PSO-based clustering protocol.

4.2.2. The Proposed Clustering Protocol: TD-PSO

The proposed TD-PSO protocol considers node position, node speed, node direction, and remaining energy to form the clusters and elect the cluster head. Figure 3 illustrates the flowchart of the proposed TD-PSO clustering protocol.

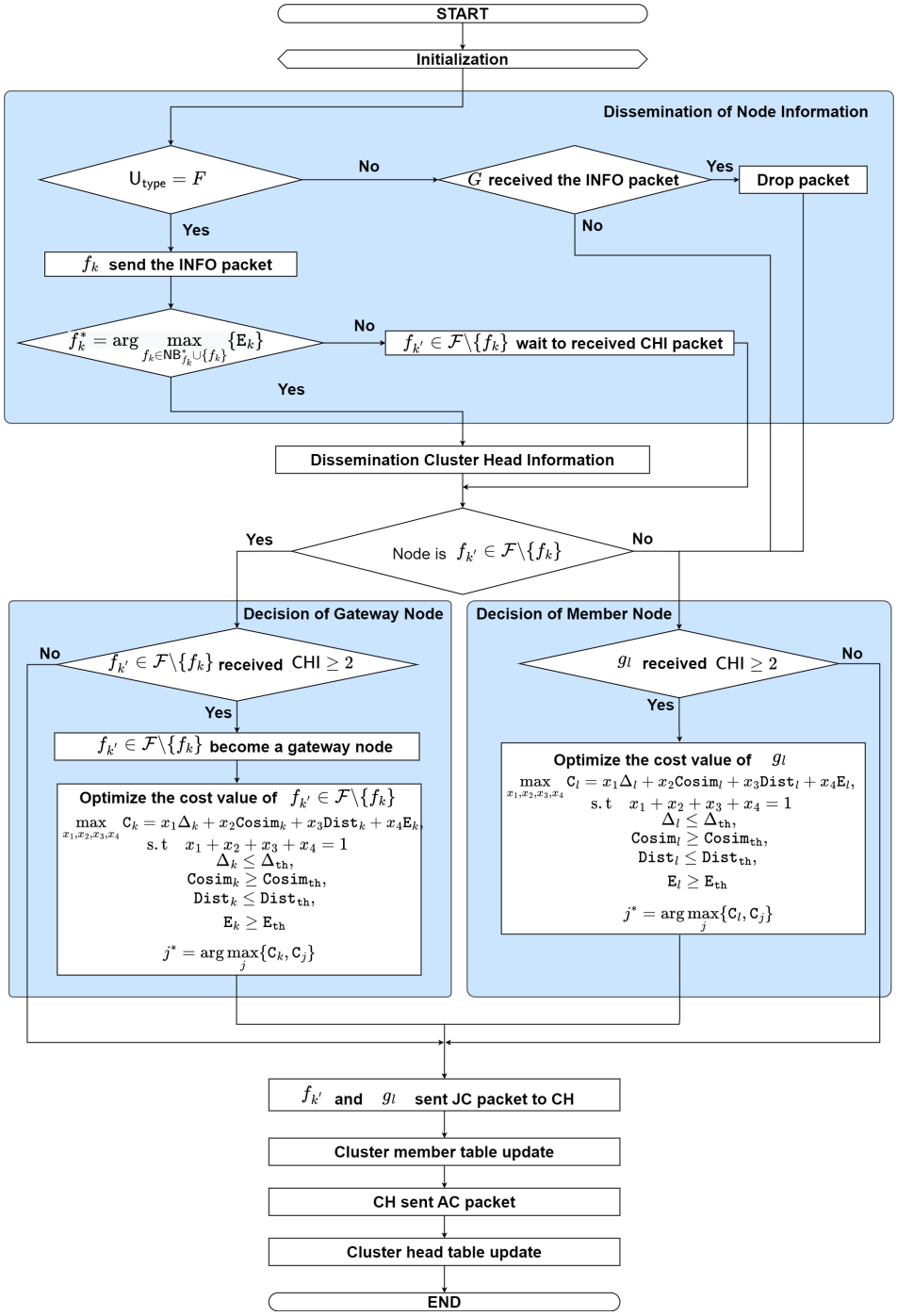


Figure 3. The flowchart of the proposed TD-PSO-based clustering protocol.

The procedure of forming a cluster and electing the cluster head can be explained as follows:

- **Step 0: Initialization**
  The nodes turn on and operate independently when the simulation starts. Then, go to **step 1**.
- **Step 1: Dissemination of Node Information**
  Each flying node f_k periodically estimates its information such as speed, position, direction, and remaining energy. Then, f_k generates and broadcasts information (INFO) packets to its neighbor UAV nodes. The INFO packet contains the following fields:

$$\langle \mathtt{Type}, \mathtt{S_{ID}}, \mathtt{D_{ID}}, \mathtt{E}, \mathtt{U_{type}} \rangle$$

where Type represents packet type, $\mathtt{S_{ID}}$ represents source node ID, $\mathtt{D_{ID}}$ represents destination node ID, E represents the remaining energy of each node, and $\mathtt{U_{type}}$ represents the type of node (0 is G, 1 is F). Then, go to **step 2**.

– If ground node g_l receives an INFO packet, then it drops the packet; go to **step 4**. Otherwise, go to **step 4**.

- **Step 2: Election of Cluster Head**
  In order to support strong connectivity, the cluster head is selected by the most remaining energy among the candidate flying nodes, which mathematically can be expressed as

$$f_k^* = \arg \max_{f_k \in \mathsf{NB}_{f_k}^* \cup \{f_k\}} \{\mathtt{E}_k\}. \tag{3}$$

  – If $f_k = f_k^*$, the node f_k becomes the cluster head; go to **step 3**.
  – Otherwise, wait to receive the CHI packet and go to **step 4**.

- **Step 3: Dissemination of Cluster Head Information**
  The node f_k becomes the cluster head, then generates and broadcasts the CHI packet to be announced to its neighbor nodes. The CHI packet contains the following fields:

$$\langle \mathtt{Type}, \mathtt{S_{ID}}, \mathtt{D_{ID}}, \mathtt{Pos}, \mathtt{Dir}, \mathtt{S}, \mathtt{E}, \mathtt{U_{type}} \rangle$$

  where Type represents packet type, $\mathtt{S_{ID}}$ represents the source node ID, $\mathtt{D_{ID}}$ represents the destination node ID, Pos represents the node's position, Dir represents the direction of the node, S represents the speed of the node, E represents the remaining energy of the node, and $\mathtt{U_{type}}$ represents the type of the node (0 is G, 1 is F).

  – If $f_{k'} \in \mathcal{F} \backslash \{f_k\}$ receive two or more CHI packets, then go to **step 5**.
  – Otherwise, go to **step 4**.

- **Step 4: Decision of Member Node**
  When ground node g_l receives two or more CHI packets, g_l will decide which cluster head follows by calculating the cost function of the cluster head candidate. The ground node will select the cluster head candidate with the largest cost value. The objective is to maximize the cost value by considering weight value under node degree, cosine similarity, cosine distance, and energy, which can be formulated as

$$\max_{\{x_1,x_2,x_3,x_4\}} \quad \mathtt{C}_l = x_1\Delta_l + x_2\mathtt{Cosim}_l + x_3\mathtt{Dist}_l + x_4\mathtt{E}_l, \tag{4a}$$

$$\text{s.t} \quad x_1 + x_2 + x_3 + x_4 = 1, \tag{4b}$$

$$\Delta_l \leq \Delta_{\mathtt{th}}, \tag{4c}$$

$$\mathtt{Cosim}_l \geq \mathtt{Cosim}_{\mathtt{th}}, \tag{4d}$$

$$\mathtt{Dist}_l \leq \mathtt{Dist}_{\mathtt{th}}, \tag{4e}$$

$$\mathtt{E}_l \geq \mathtt{E}_{\mathtt{th}}, \tag{4f}$$

where (4b) denotes that the total weight of the particle must be equal to one, (4c) denotes that the node degree difference must be lower than or equal to the node degree threshold, (4d) denotes that the cosine similarity between two nodes must be greater than or equal to the cosine similarity threshold, (4e) denotes that the cosine distance between nodes must be lower than or equal to the cosine distance threshold, and (4f) denotes that the energy of the cluster head must be greater than or equal to the energy threshold. We consider four factors which consist of the node degree difference, cosine similarity, the cosine distance between two nodes, and the remaining energy of its node. The node degree of its nodes can be written as

$$\mathtt{D}_i = \sum_{j=1, j\neq i}^{n} \mathtt{H}_i^j \tag{5}$$

$$\mathtt{H}_i^j = \sum_{i=0}^{n} \{\mathtt{dist}(i,j)\} < \mathtt{R}_i\} \tag{6}$$

where $\mathtt{dist}(i,j)$ can be defined as

$$\mathtt{dist}(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{7}$$

$\mathtt{D}_i$ stands for the node degree of the node f_i, $\mathtt{R}_i$ stands for the communication range of the node i, and $\mathtt{dist}(i,j)$ stands for distance between node i and j. Then, the average node degree can be expressed as

$$\overline{\mathtt{D}}_i = \frac{\sum_{i=1}^{n} \mathtt{D}_i}{n}. \tag{8}$$

The self-adaptive node degree variance is calculated by subtracting the node measure from its average measure, which can be expressed as:

$$\Delta_i = |\mathtt{dist}(i,j)\overline{\mathtt{D}}_i| \tag{9}$$

The second factor is cosine similarity between two nodes which can be defined as [29,37]

$$\mathtt{CoSim(i,j)} = \frac{\sum_{n_i=1}^{n_I} \overrightarrow{V}_{n_i} \overrightarrow{V}_j}{\sqrt{\sum_{i=1}^{N} \overrightarrow{V}_i^2} \sqrt{\sum_{j=1, m\neq i}^{N} \overrightarrow{V}_j^2}}, \tag{10}$$

where $\overrightarrow{V}_i$ and $\overrightarrow{V}_j$ are the i-th and j-th nodes' vector information, respectively. Each node $\overrightarrow{V}_i$ is related with a mobility vector information metric value (i.e., speed, direction, and position) $\overrightarrow{V}_i = (\overrightarrow{V}_1, \overrightarrow{V}_2, \cdots, \overrightarrow{V}_j)$, where $\overrightarrow{V}_i$ constitutes the vector values which indicate link information between nodes. Under a constrained communication distance, we can control the cluster member by considering the maximum cosine similarity. Afterwards, the third factor is the cosine distance of the node, which is used to find the distance between two nodes and can be formulated by [38]

$$\mathtt{CoDis(i,j)} = \{1 - \mathtt{CoSim(i,j)}\}. \tag{11}$$

Thus, the selected cluster member can be mathematically formulated as

$$j^* = \arg\max_j \{\mathtt{C}_i, \mathtt{C}_j\} \tag{12}$$

where $\mathtt{C}_i$ represents the cost of the node i and $\mathtt{C}_j$ represents the the cost of the neighbor cluster heads near node j. After that, g_l sends the join cluster (JC) packet to the CH. The JC packet contains the following fields:

$$\langle \mathtt{Type}, \mathtt{S_{ID}}, \mathtt{D_{ID}}, \mathtt{U_{type}} \rangle$$

where $\mathtt{Type}$ represents packet type, $\mathtt{S_{ID}}$ represents source node ID, $\mathtt{D_{ID}}$ represents destination node ID, and $\mathtt{U_{type}}$ represents the type of node. Then, go to **step 5**.

– Otherwise, when g_l only receives one CHI packet, then sends the JC packet to the CH and go to **step 6**.

- **Step 5: Decision of Gateway Node**
  The flying node $f_{k'} \in \mathcal{F} \backslash \{f_k\}$ receives two or more CHI packets and become a gateway node. Similar to the ground nodes, the flying node $f_{k'}$ will decide which cluster head follows by calculating the cost function of the cluster head candidate. When the value

of the cost function is larger, the relationship between the cluster head and node is better than other relationships which can be mathematically expressed as

$$\max_{\{x_1,x_2,x_3,x_4\}} \mathtt{C}_k = x_1\Delta_k + x_2\mathtt{Cosim}_k + x_3\mathtt{Dist}_k + x_4\mathtt{E}_k, \tag{13a}$$

$$\text{s.t} \quad x_1 + x_2 + x_3 + x_4 = 1, \tag{13b}$$

$$\Delta_k \leq \Delta_{\mathtt{th}}, \tag{13c}$$

$$\mathtt{Cosim}_k \geq \mathtt{Cosim}_{\mathtt{th}}, \tag{13d}$$

$$\mathtt{Dist}_k \leq \mathtt{Dist}_{\mathtt{th}}, \tag{13e}$$

$$\mathtt{E}_k \geq \mathtt{E}_{\mathtt{th}}, \tag{13f}$$

where (13b) denotes that the total weight of the particle must be equal to one, (13c) that denotes the node degree difference must be lower than or equal to the node degree threshold, (13d) denotes that the cosine similarity between two nodes must be greater than or equal to the cosine similarity threshold, (13e) denotes that the cosine distance between nodes must be lower than or equal to the cosine distance threshold, and (13f) denotes that the energy of the cluster head must be greater than or equal to the energy threshold. Thus, the selected cluster member can be mathematically formulated as

$$j^* = \arg\max_j \{\mathtt{C}_k, \mathtt{C}_j\} \tag{14}$$

where $\mathtt{C}_k$ represents the cost of the node k and $\mathtt{C}_j$ represents the cost of the neighbor cluster heads near node j. Then, $f_{k'}$ sends the JC packet to the CH; go to **step 6**.

– Otherwise, when $f_{k'}$ only receives one CHI packet, then it sends the JC packet to the CH; go to **step 6**.

- **Step 6: Cluster Member Table Updates**
  When node $f_k^{'}$ and g_l choose the CH, they update the cluster member (CM) table as shown in Table 1 where $\mathsf{CM_{ID}}$ is a unique identifier assigned to each node within the cluster, $\mathsf{CH_{ID}}$ represents the identifier of the cluster head that the cluster member belongs to, and $\mathsf{U_{type}}$ represents the type of user. Then, go to **step 7**.
- **Step 7: Cluster Head Table Updates**
  When f_k receives a JC packet, f_k replies with an accept packet (AC) to the transmitted node, and updates the cluster head table as shown in Table 2 where $\mathsf{CH_{ID}}$ is a unique identifier assigned to each cluster head in the network, $\mathsf{CM_{ID}}$ represents the identifier of the cluster member that belongs to the cluster managed by the cluster head, and $\mathsf{U_{type}}$ represents the type of user. Finally, the cluster has been formed. The AC packet contains the following fields:

  $$\langle \mathtt{Type}, \mathtt{S_{ID}}, \mathtt{D_{ID}} \rangle$$

  where $\mathtt{Type}, \mathtt{S_{ID}}, \mathtt{and}\ \mathtt{D_{ID}}$ represent the packet type, source node ID, and destination node ID, respectively.

  – Otherwise, the **AC** packet will be dropped.

In the end, the clustering process is finished. When the source node needs to send a data transmission to multiple destination nodes, the network will start the routing process based on the routing protocol algorithm. The list of packets for the proposed TD-PSO clustering protocol is summarized in Table 3.

Table 1. Cluster member table of the proposed TD-PSO-based clustering protocol.

$\mathtt{CM}_{\mathtt{ID}}$	$\mathtt{CH}_{\mathtt{ID}}$	$\mathtt{U}_{\mathtt{type}}$

Table 2. Cluster head table of the proposed TD-PSO-based clustering protocol.

$\mathtt{CH}_{\mathtt{ID}}$	$\mathtt{CM}_{\mathtt{ID}}$	$\mathtt{U}_{\mathtt{type}}$

Table 3. List of packets for the TD-PSO clustering protocol.

Packet Name	Full Name	Field Information
INFO	Information	$\mathtt{Type}, \mathtt{S}_{\mathtt{ID}}, \mathtt{D}_{\mathtt{ID}}, \mathtt{E}, \mathtt{U}_{\mathtt{type}}$
CHI	Cluster Head Information	$\mathtt{Type}, \mathtt{S}_{\mathtt{ID}}, \mathtt{D}_{\mathtt{ID}}, \mathtt{Pos}, \mathtt{Dir}, \mathtt{S}, \mathtt{E}, \mathtt{U}_{\mathtt{type}}$
JC	Join Cluster	$\mathtt{Type}, \mathtt{S}_{\mathtt{ID}}, \mathtt{D}_{\mathtt{ID}}, \mathtt{U}_{\mathtt{type}}$
AC	Accept Cluster	$\mathtt{Type}, \mathtt{S}_{\mathtt{ID}}, \mathtt{D}_{\mathtt{ID}}$

4.2.3. Top-Down Particle Swarm Optimization Model

This subsection will explain in detail the particle swarm optimization model used in the clustering protocol.

In the TD-PSO algorithm, each individual node in the population is called a particle and moves in the search space. Particles have memory and, thus, they retain part of their previous state. Each particle's movement is the composition of a velocity and two randomly weighted influences. The two randomly weighted influences are individual and have the tendency to return to their best previous positions and sociality, or the tendency to move towards their neighborhood's best previous position. As briefly mentioned above, clustering involves gathering similar objects that must first be defined.

Let us assume the objective function $\mathtt{C}_i$ where $i \in \{\mathtt{FN}, \mathtt{GN}\}$ is the global optima $\mathtt{C}_i^*$ of optimization with optimal value x_l, $l \in \{1, 2, 3, 4\}$. The proposed TD-PSO algorithm to find the optimal values, i.e., x_1, x_2, x_3, x_4, can be summarized as Algorithm 1, where p represents the number of parameters, q denotes the number of particles, x_{pq} represents the position of particle q for parameter p, v_{pq} represents the velocity of particle q for parameter p, r_1 and r_2 are random values with range [0, 1] to avoid premature convergence, w denotes the weight of a particle with range [0.4, 0.9], c_1 and c_2 are acceleration factors, and t is the number of iterations.

Algorithm 1 The TD-PSO algorithm to find global optimal points in problems (4) and (13)

Output: Optimal solution $\mathtt{C}^*$ and x_l^*

Initialization :
Set $p \leftarrow 4$. $\leftarrow$ number of parameters
Set $q \leftarrow 50$. $\leftarrow$ number of particles
Set $(x_{pq}, v_{pq}) \leftarrow$ randomly with constraint.
Set $w \leftarrow$ randomly with range $[0.4, 0.9]$.
Set $(r_1, r_2) \leftarrow$ randomly with range $[0, 1]$.
Set $(c_1, c_2) \leftarrow 2$.
for $\kappa = 0, 1, 2, \cdots$ **do**
    calculate problem, find $\mathtt{C}^*, x_l^*$;
    if $\mathtt{C}^* < \mathtt{C}^{(t)}$ **then**
        $\mathtt{C}^* \leftarrow \mathtt{C}^{(t)}$
        $x_p^* \leftarrow x_p^{(t)}$
    end if
    $v_{pq}^{(t+1)} = w.v_{pq}^{(t)} + c_1.r_1(x_{bl}^{(t)} - x_{pq}^{(\kappa)}) + c_1.r_1(g_{bl}^{t} - x_{pq}^{(t)})$
    $x_{pq}^{(t+1)} = x_{pq}^{(t)} + v_{lj}^{(t)}$
end for

4.3. The Proposed Deep Learning-Based Secure Multicast Routing Protocol: DLSMR

We describe the proposed DLSMR protocol to establish a high-stability multicast tree and improve security performance as shown in Figure 4. The objective function of this approach is to predict the secure and unsecured nodes during the routing process. Due to the node's mobility changing rapidly in FANETs, it makes it easy to attack the network. In particular, wormhole attacks pose a high risk when the nodes can manipulate packets during routing and data transmission. Furthermore, also, even when we consider unicast transmission, it makes the control overhead very high. Thus, this paper proposes deep learning-based secure multicast routing to avoid wormhole attacks and improve control overhead in FANETs. It differs from previous works for solving wormhole attacks using bio-inspired, position-based, and distance methods. In this approach, we consider deep learning to predict the secure and unsecured nodes quickly during the routing process. The use of this process can mitigate these threats, ensuring reliable and secure communication between nodes. Additionally, multicast routing protocols can accommodate this dynamic nature by allowing for frequent group membership changes and route updates. As can be observed in Figure 4, a source node (S) needs to establish a multicast tree to the multiple destination nodes (D). After the clustering process is completed, each D that wants to receive data transmission sends a join request to the CH in the cluster. Then, S sends a MRREQ packet to each CH by unicast to find the multiple destination nodes. After that, CH will broadcast the MRREQ packet to each CH to find the multiple destination nodes. When UGW receives the MRREQ packet, it updates its routing table and re-broadcasts the MRREQ packet to the neighbor nodes. When a CH who has a multicast group ID receives an MRREQ packet, it updates the routing table and replies a RREP packet to the previous node by unicast. We design the DL framework with the node ID, node position, destination ID, destination sequence, hop count, and remaining energy as input parameters, and secure and unsecured node ID, secure and unsecured status as output parameters. By using a DL framework, we can determine whether the next node is secure or not to establish a secure multicast route from S to multiple destinations as detailed in Section 4.4.

Figure 5 illustrates the flowchart of the proposed secure multicast routing protocol, which can be summarized as follows:

Route Request Process:

- **Step 1: Initialization**
  After the clustering process is completed, each destination node (multicast member node) in a cluster that wants to receive certain data sends a join request (JREQ) to its cluster head by unicast. The JREQ contains the following:

$$\langle \mathsf{Type}, \mathsf{S}_{\mathsf{ID}}^{\mathsf{JREQ}}, \mathsf{D}_{\mathsf{ID}}, \mathsf{Status} \rangle$$

  where Type denotes packet type, $\mathsf{S}_{\mathsf{ID}}^{\mathsf{JREQ}}$ identifies the node that wants to join a multicast group, D_{ID} denotes the node ID that wishes to join, and Status denotes the status of the node (join, leave, etc.). Next, the cluster head stores the multicast ID (M_{ID}) associated with this request in its own table; go to **step 2**.
- **Step 2: Source Node Operation for Route Request: Generates and Sends Route Request Packet**
  A node that wants to send data to the multicast group becomes the S. S initiates the process by generating a MRREQ packet and unicasts this MRREQ to its CH. The MRREQ packet contains the following fields:

$$\langle \mathsf{Type}, \mathsf{S}_{\mathsf{ID}}^{\mathsf{MRREQ}}, \mathsf{MG}_{\mathsf{ID}}, \mathsf{M}_{\mathsf{ID}}, \mathsf{S}_{\mathsf{Seq}}, \mathsf{Hop}, \mathsf{TTL} \rangle$$

  where Type represents packet type, $\mathsf{S}_{\mathsf{ID}}^{\mathsf{MRREQ}}$ represents the source node ID, $\mathsf{MG}_{\mathsf{ID}}$ represents the multicast group ID, M_{ID} represents the multicast ID, $\mathsf{S}_{\mathsf{Seq}}$ is the source sequence number, Hop denotes the number of hops between two nodes, and TTL is the time to live of the packet in the network. Otherwise, go to **step 3**.

- **Step 3: Intermediate Node Operation at Cluster Head for Route Request**
  When CH_c receives the MRREQ packet, CH_c will first check the MRREQ packet.
  - **Step 3.1:** If the S_{Seq} at the received MRREQ is larger than that of the routing table, then go to **step 3.2**.
    - Conversely, if the S_{Seq} at the received MRREQ equals the S_{Seq} at the routing table and the M_{ID} at the received MRREQ is equal to the M_{ID} at the routing table, or the M_{ID} is greater than the M_{ID} at the routing table and $\{hop + 1\}$ at the received MRREQ is less than hop at the routing table, then go to **step 3.2**. Otherwise, the packet will be dropped.
  - **Step 3.2:** If the TTL at the received MRREQ is greater than or equal to the TTL at the routing table, then go to **step 3.3**. Otherwise, the packet will be dropped.
  - **Step 3.3:** If the cluster member ID (CM_{ID}) is the same as the MG_{ID}, CH_c records the sender's ID, updates the routing table, and broadcasts MRREQ to NB_i; then, go to **step 5**.
    - Otherwise, CH_c records the sender's ID and updates the routing table and broadcasts MRREQ to the neighbor node (NB_i) in its cluster or the next cluster heads; go to **step 4**.
- **Step 4: Intermediate Node Operation at Gateway for Route Request**
  When the gateway node UGW_c receives a MRREQ packet from NB_i, UGW_c records the sender's ID and updates the routing table, then UGW_c broadcasts the MRREQ to their neighboring CH nodes until $TTL \geq TTL^{th}$; go to **step 3**.

Route Reply Process:

- **Step 5: Cluster head Operation for Route Reply: Generates and Sends Route Reply Packet**
  CH generates and replies a RREP packet to the previous node by unicast transmission. The RREP packet contains the following fields:

$$\langle Type, S_{ID}^{RREP}, D_{ID}^{RREP}, M_{ID}, D_{Seq}, E, Pos, Hop \rangle$$

  where Type represents the packet type, S_{ID} represents the source node ID, D_{ID} represents the destination ID, M_{ID} represents the multicast ID, D_{Seq} is the destination sequence number which is the number of attempts to confirm control messages, E represents the remaining energy of the node, Pos represents the position of the node, and Hop represents the number of hops to D. Then, go to **step 6**.
- **Step 6: Intermediate Node Operation at Gateway Node for Route Reply**
  The intermediate node NB_i records the sender's ID of RREP packets and updates its routing table when it receives the RREP packet. Then, NB_i forwards the RREP packet to the previous node; go to **step 7**. Otherwise, NB_i waits until it receives the RREP packet.
- **Step 7: Intermediate Node Operation at CH for Route Reply**
  When the CH_c receives RREP, then go to **step 7.1**. Otherwise, CH_c waits until it receives the RREP packet.
  - **Step 7.1:** When the CH_c receives the RREP, then CH_c checks the S_{ID}^{RREP} in the RREP packet. The S_{ID}^{RREP} in the RREP packet means the destination node ID. So, if the S_{ID}^{RREP} equals S_{ID}^{tab}, then go to step **step 7.2**. Otherwise, CH_c records the sender's ID, updates the routing table, and forwards to the previous node by unicast transmission; then, go to **step 8**.
  - **Step 7.2:** When the S_{ID}^{RREP} equals S_{ID}^{tab}, we will predict the best secure next node with input parameter consisting of the node ID, position of the node, destination ID, destination sequence number, and remaining energy. The output parameters consists of the NID_{sec}^{1}, NID_{nsec}^{2}, and the status of the node (secure or unsecured) to select the secure next node as the best next node while establishing the route from S to multiple D_i by using a DNN model; then, go to step **step 7.3**.

- **Step 7.3:** If the $Stat_{sec}$ equals 1, then CH_c can determine the secure route to be pursued by the secure next node. Then, CH_c records the NID_{sec}, updates the routing table, and forwards to the previous node. The multicast tree routing table can be summarized in Table 4, where S_{ID} is the source ID, MG_{ID} is the multicast group ID, S_{Seq} is the source sequence number, D_{Seq} is the destination sequence number , M_{ID} is the multicast node ID, PN_{ID} is the previous node ID, NN_{ID} is the next node ID, and Hop represents the number of hops to D. The routing table will be used to determine the next node to the multicast group D that data packets will pass through during the data transmission process. Then, go to **step 8**. Otherwise, the packet will be dropped and the process is ended.

- **Step 8: Source Node Operation for Route Reply**
  If S receives all RREP packets from the multicast group, then go to the data transmission process in **step 9**. Otherwise, go to step **Step 8.1:**.
  - **Step 8.1:** If Timer $\leq 2 \times TTL^{th}$, wait for all RREP packets from the multicast group until Timer $> 2 \times TTL^{th}$. Otherwise, go back to **step 2**.

Data Transmission Process:

- **Step 9: Data Transmission at Source Node**
  When S receives all RREP packets from the multicast group D_i, the S multicasts the data packet to the next hops based on the deep learning framework. If a node of the tree receives a data packet, it will forward the data packet to the multicast group in the same way as the source.

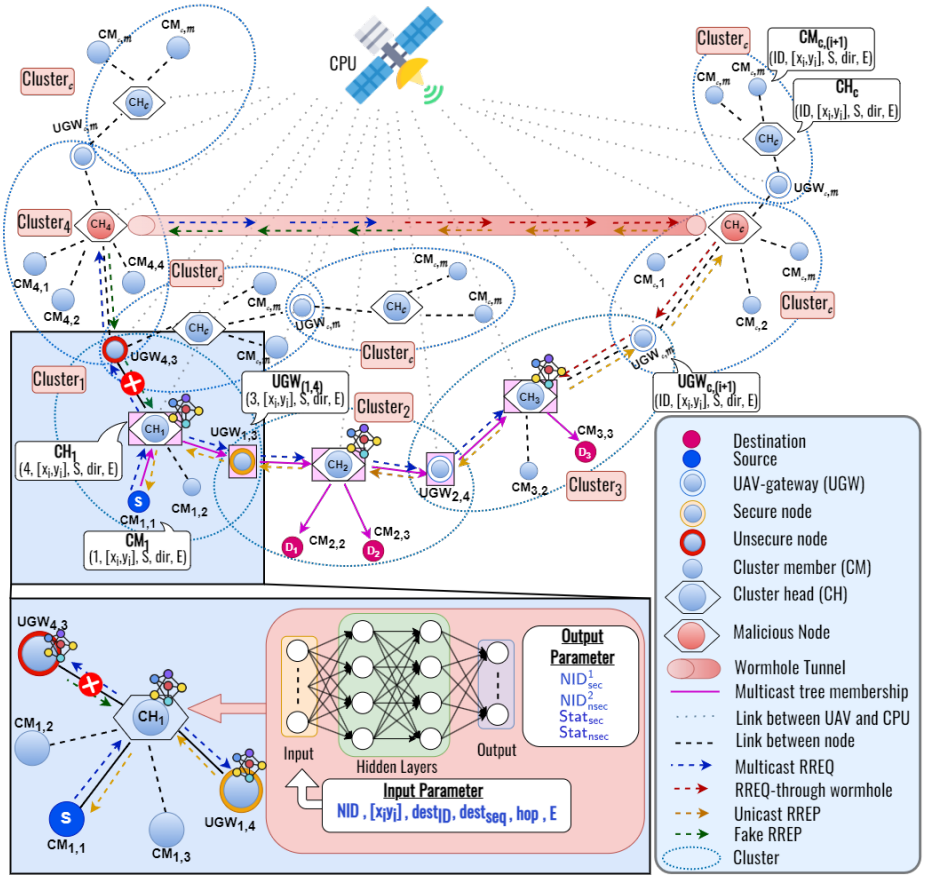


Figure 4. The basic concepts of the deep learning− (-) based secure multicast routing (DLSMR) protocol.

Table 4. Multicast tree routing table of the proposed DLSMR protocol.

S_{ID}	MG_{ID}	M_{ID}	S_{Seq}	D_{Seq}	PN_{ID}	NN_{ID}	Hop

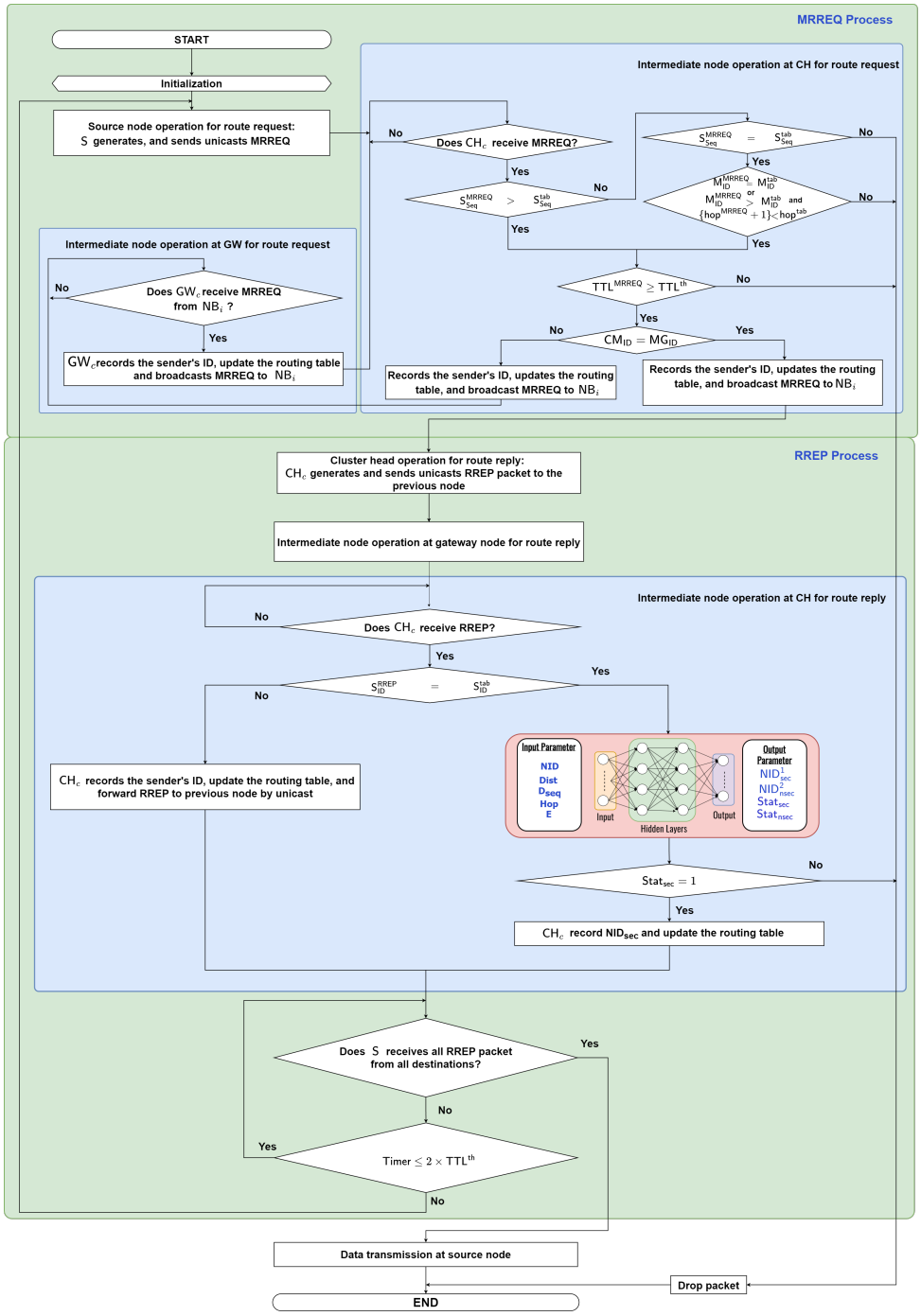


Figure 5. Flowchart of the proposed deep learning-based secure multicast routing (DLSMR) protocol.

The list of packets for the DLSMR protocol are summarized in Table 5.

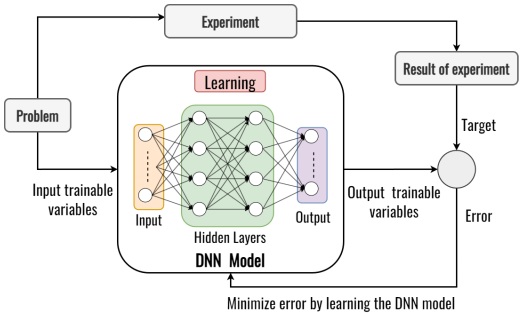
Table 5. List of packets for the DLSMR protocol.

Packet Name	Full Name	Field Information
JREQ	Join Request	$\text{Type}, S_{ID}^{JREQ}, D_{ID}, \text{Status}$
MRREQ	Multicast Route Request	$\text{Type}, S_{ID}^{MRREQ}, MG_{ID}, M_{ID}, S_{Seq}, \text{Hop}, \text{TTL}$
RREP	Route Reply	$\text{Type}, S_{ID}^{RREP}, D_{ID}^{RREP}, M_{ID}, D_{Seq}, E, \text{Pos}, \text{Hop}$

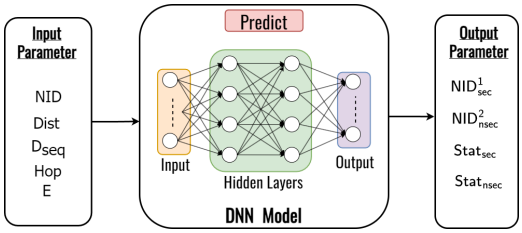
4.4. The Proposed Deep Learning Design

As shown in Figure 1, a wormhole attack occurs when an attacker creates a tunnel between two malicious nodes in the network, allowing them to capture packets at one end and replay them at the other end instantly. A wormhole creates the illusion of a shorter and more efficient route between the two malicious nodes. To solve this problem, we develop a novel deep learning framework to predict the secure next node while establishing the route from the source to multiple destinations in FANETs.

Each node in the network can determine which node in the network is the most secure for multiple destinations through a deep learning framework. In this subsection, we design the deep learning framework to capture the relation between network parameters and system performances as shown in Figure 6. The main objective of this work is to predict the secure next node in the proposed DLSMR protocol. In this work, we utilize multivariate regression, which is more challenging than single regression. A deep learning model includes two phases, the training phase and the testing phase.



(**a**) Training



(**b**) Testing

Figure 6. Comparison of packet loss ratio and number of packet losses as a function of node speed.

In the training phase, the input parameters consist of the node identifier (NID), the distance between two nodes (Dist) which are denoted in Equation (7), the destination sequence (D_{seq}), the number of hops (Hop), and the remaining energy of the node (E). According to these input parameters, the model predicts the secure node ID, the unsecured node ID, the secure status, and the unsecured status as the output.

In a training iteration, as shown in Figure 6a, an error is obtained by comparing the deep learning output with the target and the simulation result obtain four outputs. Then, the error is minimized by updating the weights and biases on the neurons using back-propagation, which continues until the iteration is satisfied. The trainable deep learning framework is tested using a new input variable to predict the secure and unsecured next nodes as shown in Figure 6b. In our design, we use a feed-forward neural network with 1×5 dimensional input layers, L hidden layers, and 1×4 dimensional output layers to obtain the two kinds of ID and status of the node as shown in Figure 7.

Therefore, the proposed deep learning design can reduce complexity and also can predict the secure next node in real-time based on the information of its node as the input parameter. The layer structure used in the deep learning design to improve system performance is shown in Table 6.

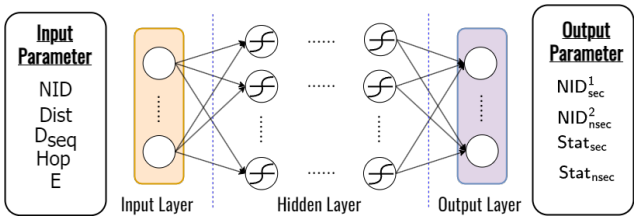


Figure 7. The structure of the deep learning framework for predicting the secure and unsecured next node.

Table 6. Model of the deep neural network to predict the secure next node.

	Size	Activation Function
Input	5	-
Layer 1	150	ELU
Layer 2	100	ELU
Layer 3	200	ELU
Layer 4	150	ELU
Layer 5	100	ELU
Output	4	LINEAR

5. Performance Evaluation

5.1. Simulation Environments and Parameters

In this section, we evaluate the performance of the proposed routing protocol, DLSMR. To illustrate our proposed protocol, we deploy 30, 50, and 100 nodes within an urban area of 1000 m $\times$ 1000 m and a transmission range of 250 m. In addition to that, we deploy a wormhole pair consisting of two wormhole nodes. These wormhole nodes are placed randomly within the network for each simulation run. We establish a 'tunnel' between these randomly positioned wormhole nodes to simulate the wormhole attack, allowing them to shortcut the normal network routing. In this scenario, we simulate our proposed protocol under two different mobility models, namely random waypoint mobility (RWP) [39] and reference point group mobility (RPGM) [40], to evaluate its performance in FANETs. Every result in this simulation is an average from 200 sections, with pausing and moving times set at 3 s and 5 s, respectively. Furthermore, the mobile nodes are initially randomly distributed around the simulated area and move at different speeds (15 km/h, 30 km/h, 45 km/h, 60 km/h, and 75 km/h). The MAC protocol is modeled using the IEEE 802.11a standard and uses a receiver signal strength indicator (RSSI) threshold of -80 dbm for communication range to make it more practical. The main reason for considering RSSI is that the value of RSSI fluctuations obtained has taken into consideration its effect on changes in channel conditions, including multi-path fading [41].

The simulation experiments are conducted using the NS3 simulator. We summarize the simulation environment and parameters in Table 7. Additionally, we measured the accuracy between the predicted secure next node and the output data of the test set by calculating the root mean square error (RMSE) of the proposed deep learning framework. The RMSE can be written as [42]

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(p_{(t)} - \widehat{p}_{(t)})^2} \tag{15}$$

where n denotes the number of samples in the test set, $p_{(t)}$ denotes the predicted value of the t-th observation in the dataset, and $\widehat{p}_{(t)}$ denotes the observed value for the t-th observation in the dataset. It is more likely that the predicted secure next node and observation are closely matched when the RMSE is smaller.

Table 7. Parameters and simulation environments.

Parameters	Value
Simulator	NS-3 simulator
Simulation area	1000×1000 m^2
Packet size	1024 bits
Mobility model	RPGM and RWP
Transmission range	250 m
Simulation time	200 s
Session length	5 s
Number of nodes	[30, 50, 100]
Wormhole pairs	1 (Wormhole nodes 2)
Node's speed range	[15:15:75] (km/h)
Receive signal strength indicator (RSSI) threshold	−80 dBm
MAC protocol	802.11a

5.2. Performance Metrics

The performance metrics used in this paper for performance evaluation purposes are the following [29]:

- Packet delivery ratio (PDR): This is defined as the number of data packets delivered to multicast destinations over the number of data packets supposed to be delivered to multicast destinations. This ratio represents the effectiveness of the routing strategy.
- Control overhead: This refers to the average number of control packets sent to nodes during the route creation process per session per node per multicast data delivered.
- Delay: This is defined as the average delay to establish a multicast route from source to multicast destinations per one session.
- The average number of cluster head changing: This refers to the number of cluster heads changing per cluster per session on average.
- Packet loss ratio: This is defined by the proportion of data packets that are lost during transmission from the sender in a multicast group. A lower packet loss ratio indicates better performance and reliability of the secure multicast routing protocol.

5.3. Numerical Results

In this subsection, we present numerical results to validate the efficacy of our proposed DLSMR protocol. The simulation settings are outlined in Table 7. We use the NS3 simulation, where the algorithm is run for 200 s with 5 s for each session. For the DNN model, the dataset is generated over 1,000,000, 90% of which are used for training while the remaining 10% are allocated for validation. Additionally, we construct 100 distinct datasets to evaluate the performance of the trained DNN model. Our objective is to predict the secure next node through the DNN model accurately. The parameters employed for DNN training are detailed in Table 8.

Table 8. DNN training parameters.

Parameters	Value
Dataset	1,000,000
Epoch	50
Batch size	256
Optimizer	Adam
Initial learning rate	0.00001

To demonstrate the effectiveness of the proposed algorithm (DLSMR with TD-PSO), we compare its performance with the multicast ad hoc on-demand distance vector (MAODV) routing protocol with or without TD-PSO clustering protocol.

First, we evaluate the impact of the number of iterations on the maximum cost of the TD-PSO clustering algorithm in Figure 8. As can be observed in Figure 8, the proposed

Algorithm 1 converges at the fifth iteration with the number of population (nPop) of 100, which shows the proposed TD-PSO for the clustering protocol is efficient to find the optimal value. The reason is that the algorithm has a higher chance of generating a solution close to the global optimum during the initial stages. Moreover, the algorithm with a population of 100 outperforms the algorithm with a population of 50. The reason is that when the population size increases from 50 to 100, the algorithm has a larger set of candidate solutions to find the optimal solution.

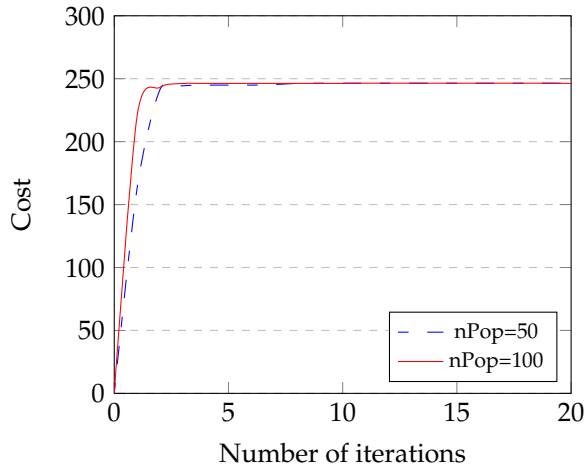


Figure 8. Convergence of Algorithm 1 for maximize cost problem as a function of the number of iterations.

Figure 9 shows the comparison of the average number of cluster head changes in each session as a function of node speed. As can be seen in Figure 9, when the number of nodes increases from 30 to 100, the average number of cluster head changes increases. Furthermore, when node speed increases from 15 to 75 km/h, the average number of cluster head changes also increases. The reason is that, when the node's speed increases, the node's relative position can change quickly. Therefore, the nodes' arrangement and connections can also be changed rapidly. In the clustering protocol, if nodes are moving quickly, the optimal choice for the cluster head can change frequently as nodes move in and out of range of each other. This might require frequent re-election of the cluster head, increasing the average number of cluster head changes. Additionally, higher speeds might lead to increased link breakages, which would necessitate the formation of new clusters, further increasing the frequency of cluster head changes. Furthermore, when comparing the mobility models, the proposed protocol with the RPGM model outperforms that with the RWP model. In RPGM, nodes in a group follow a predefined reference point, resulting in more stable network topology and less frequent cluster head changes. Additionally, RPGM also captures the correlation of movement between nodes in a group, while in the RWP model each node moves independently, leading to more frequent changes in network topology and causing more cluster head changes. Therefore, in the proposed clustering protocol in a scenario where the number of nodes (30, 50, 100) and their speed increases, the average number of cluster head changes increases slightly when the network topology changes rapidly. Nevertheless, the number of cluster head changes in each session is less than one. This means the clustering protocol has a very high level of stability.

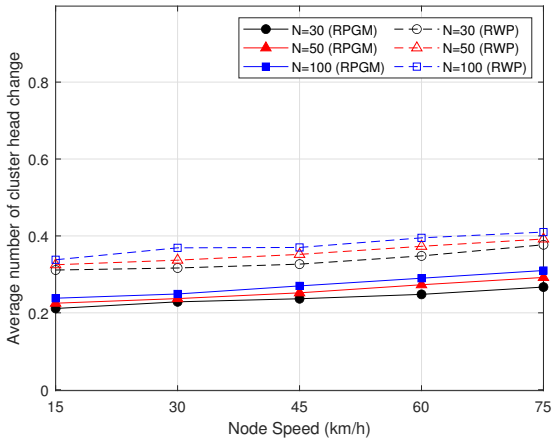


Figure 9. Average number of cluster heads changing as a function of node speed.

In Figure 10, we analyze the impact of the number of hidden neurons on the DNN model with the different numbers of hidden layers. As we can see in Figure 10, the RMSE with one hidden layer will decrease from 0.3164 to 0.0138, with two hidden layers will decrease from 0.0583 to 0.006, and with five hidden layers will decrease from 0.036 to 0.000124 when the number of hidden neurons increases from 5 to 250. It can be explained that the DNN model with more neurons performs better than the DNN model with fewer neurons. Furthermore, the more hidden layer of the DNN model performs better than the less hidden layer.

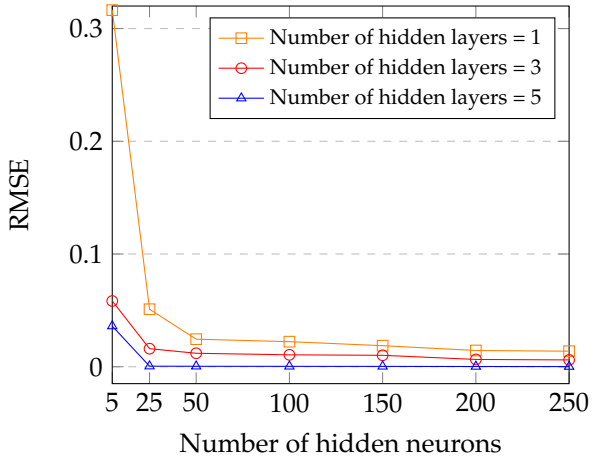


Figure 10. Impact of the number of hidden neurons on the DNN model with different numbers of hidden layers.

Figure 11 illustrates the PDR versus the variation in node speed. As can be observed in Figure 11, when the speed of the node increases, the PDR is decreased. The reason is that the entire network becomes more unstable as node speed increases. This instability leads to more frequent disruptions of the multicast tree structure and potential packet losses, decreasing the PDR. Again, when comparing the two mobility models, it is evident that the RPGM protocol results in superior performance compared to the RWP model, which exhibits a high PDR. However, it should be noted that the reduction in PDR is significantly less in the case of the DLSMR+TD-PSO protocol compared to other schemes. Therefore, the DLSMR+TD-PSO protocol with the RPGM model shows the highest level of reliability in maintaining a high PDR.

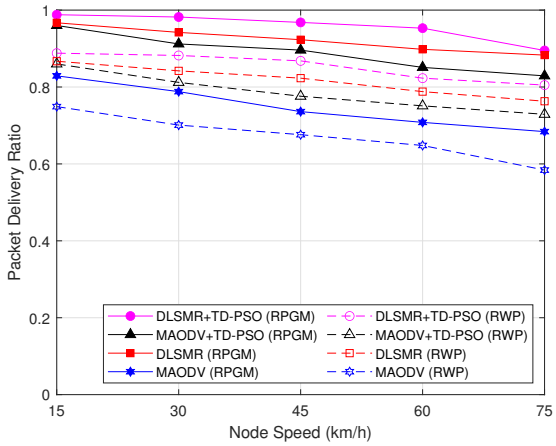


Figure 11. Packet delivery ratio as a function of node speed.

Figure 12 reveals the impact of node speed on routing delays, including the latency time for cluster construction per session as a function of node speed. As can be observed in Figure 12, the routing delay increases when the node speed increases. The reason is that nodes will move more dynamically when the node's speed increases, consequently increasing the time to establish the route. Once again, the protocols under the RPGM model demonstrate higher stability than protocols under the RWP models because the coordinated group movement in RPGM reduces sudden changes in network topology, thereby reducing route establishment delays. On the other hand, the proposed DLSMR+TD-PSO protocol only involves UAVs as CHs and GW nodes to determine the route to be followed by packets. Thus, the proposed DLSMR+TD-PSO protocol with the RPGM model can send packets with a minimum delay compared to other schemes.

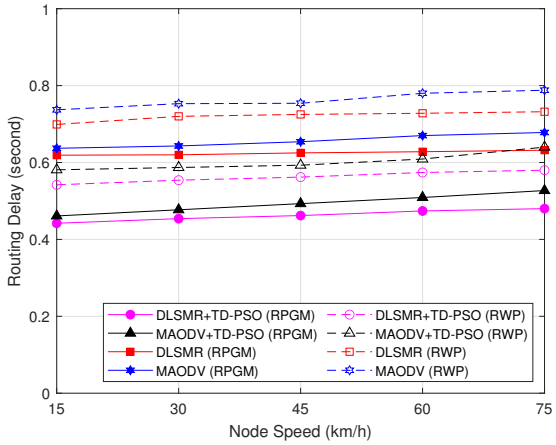


Figure 12. Routing delay as a function of node speed.

Figure 13 presents a comparison of the control overhead as a function of node speed, including the control overhead for cluster formation per node per session. As can be seen, the control overhead increases a little bit when the speed of the node increases. Essentially, this is due to the fact that, when the node's speed increases, more packets will be needed for route establishment and thus the overhead will rise. Again, comparing the different mobility models, the RPGM model demonstrates better control overhead due to high stability than the RWP model. In addition to that, the proposed TD-PSO clustering protocol can reduce the control overhead in all schemes. This means that the TD-PSO clustering protocol only involves CH and GW in the routing process, and the control overhead decrease is much less in the scheme with TD-PSO clustering protocol compared with schemes without TD-PSO clustering protocol. On the other hand, the proposed DLSMR

routing protocol outperforms the MAODV protocol. Thus, the DLSMR+TD-PSO protocol with the RPGM model presented has the best performance which can improve connectivity and provide a stable connection compared to the other schemes regarding control overhead.

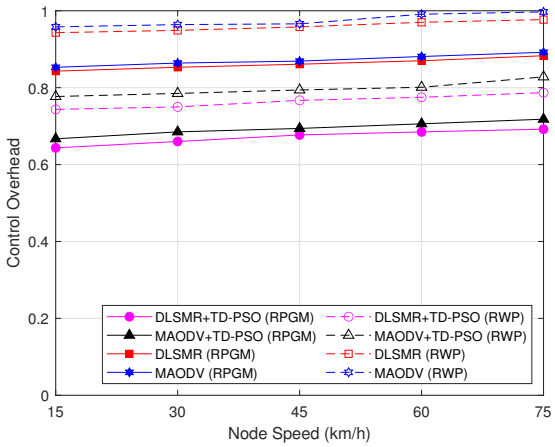


Figure 13. Control overhead as a function of node speed.

In Figure 14, we demonstrate the PDR as a function of multicast group size to evaluate the scalability of the DLSMR+TD-PSO protocol. It can be observed that the PDR has a nearly constant value and is unaffected when the multicast group size increases. The reason is that the DLSMR+TD-PSO protocol is capable of delivering packets to multiple destinations at the same time, ensuring consistent PDR even as the size of the multicast group increases. The protocol under the RPGM model shows superior performance compared to the RWP model. This is primarily because nodes in RPGM move in groups, which results in a more predictable and less chaotic network. Consequently, this leads to fewer packet losses and a better PDR, despite an increase in destinations. Furthermore, the DLSMR+TD-PSO protocol accurately predicts the secure next node during the routing process, even in a dynamic network. TD-PSO also helps the optimization of the allocation of network resources, thereby ensuring efficient data delivery. As a result, we can conclude that the DLSMR+TD-PSO protocol with the RPGM model has strong scalability in terms of PDR.

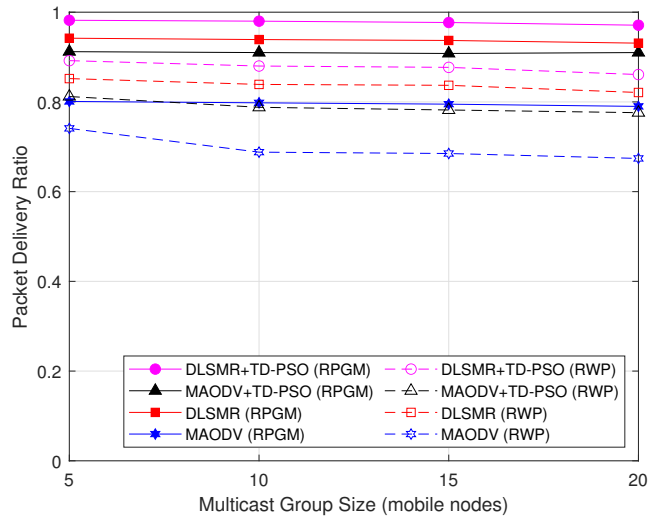
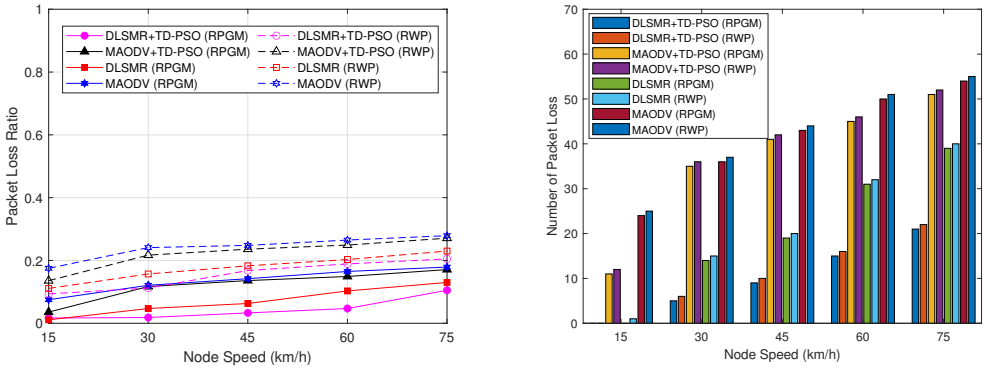


Figure 14. PDR as a function of multicast group size.

Then, we will turn our attention to look at the security performance. Figure 15 illustrates the comparison of the average packet loss ratio and number of packet losses in each session as a function of node speed as shown in Figure 15a and Figure 15b, respectively.

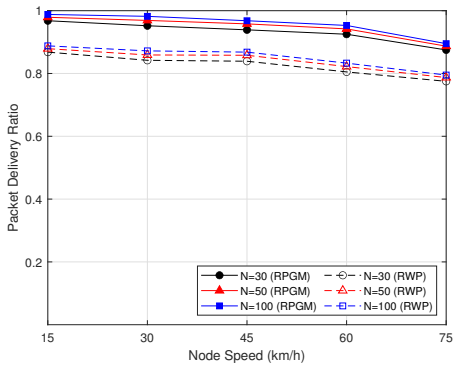
According to Figure 15a, when the node's speed increases, the ratio of packet loss increases. Furthermore, also in Figure 15b, when the node's speed increases, the average number of packet losses increases. The reason is that, when the node's speed increases, the node's location frequently changes, thus causing packets to be transmitted directly to wormhole nodes. Furthermore, the proposed TD-PSO clustering protocol can reduce the number of links between nodes. Therefore, it can improve the packet loss ratio and the number of packet loss performances. On the other hand, the RPGM model instead of RWP can help the mitigation of the movement issue due to the collective and predictable movement of nodes resulting in fewer route changes and more stable connections, also reducing the likelihood of packet loss. Thus, the proposed DLSMR+TD-PSO protocol with the RPGM model is proven to be secure from a network security perspective.



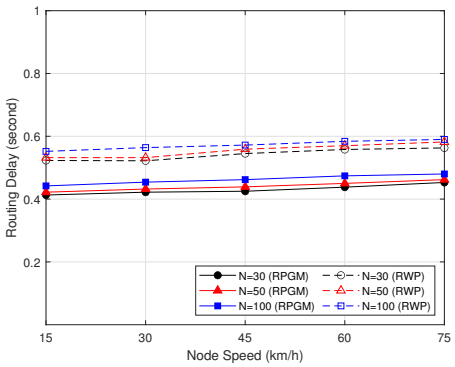
(**a**) Packet Loss Ratio vs. Node Speed (**b**) Number of Packet Loss vs. Node Speed

Figure 15. Comparison of packet loss ratio and number of packet losses as a function of node speed.

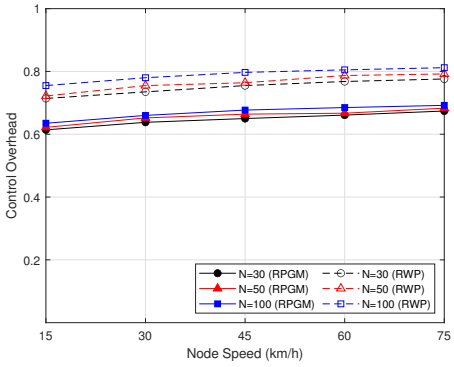
Figure 16 shows the impact of the node density of the proposed DLSMR+TD-PSO protocol on the network metrics. We set the minimum number of nodes at 30, 50, and the maximum of 100. As can be seen in Figure 16a, the PDR decreases when the node speed increases, while the PDR slightly increases when the number of nodes increases. This behavior is influenced by the more predictable movement patterns in the RPGM model, which leads to less frequent route changes and fewer packet losses. Furthermore, we evaluate the routing delay as a function of node speed with different numbers of nodes, as shown in Figure 16b. It is observed that, when the number of nodes and speed of nodes increases, the routing delay increases slightly but not significantly. This means that when the number of nodes and node speed increases, the number of hops also increases, which can cause the routing process to take longer. In addition to that, the effect of node speed on the control overhead is illustrated in Figure 16c. It can be seen that when the node speed and the number of nodes increases, the control overhead increases. The reason is that the network density increases when nodes increase, resulting in more frequent packets to establish routes. Therefore, increased node speed or increased number of nodes requires more resources to establish a multicast route; consequently, the control overhead will be increased. Despite these challenges, it can be proved that the proposed DLSMR+TD-PSO protocol with RPGM mobility model has good scalability and effectively manages to improve the PDR, routing delay, and control overhead as the number of nodes increases.



(**a**) PDR vs. Node Speed



(**b**) Routing delay vs. Node Speed



(**c**) Control overheads vs. Node Speed

Figure 16. The impact of the number of nodes on the network performance metric.

6. Conclusions

In this paper, we proposed a DLSMR and TD-PSO protocol in FANETs with CF-mMIMO to establish a secure multicast route that improves node connectivity against wormhole attacks. The proposed DLSMR protocol utilized a DL approach to predict the secure and unsecured route based on various parameters such as node ID, distance, destination sequence, hop count, and energy which can avoid wormhole attacks. To enhance node connectivity, we proposed a TD-PSO clustering protocol that employed particle swarm optimization to find the global optimal points to maximize the cost function. This function considered the weight of the remaining energy, cosine similarity, cosine distance, and node degree, which led to electing the cluster head candidate. Furthermore, we also compared the protocol performance under two different mobility models (RPGM and RWP). The performance evaluation showed that the proposed DLSMR protocol with TD-PSO clustering protocol with the RPGM model can establish highly stable multicast trees that are robust to wormhole attacks. The proposed DLSMR protocol has better security performance than the MAODV protocol as a benchmark against wormhole attacks. This is indicated by the PDR and number of packet loss performance values which are better than MAODV. Simultaneously, the TD-PSO protocol can improve node connectivity and manage multicast members efficiently with good control overhead, number of cluster head changes, and routing delay. Overall, the proposed DLSMR protocol with TD-PSO clustering protocol under the RPGM model guarantees high stability, low routing delay, low packet loss ratio, low number of packet losses, low control overhead, and high PDR. To expand this work, we are trying to develop a secure routing protocol with a cross-layer design to protect the information against eavesdropper attacks using the DL technique.

Author Contributions: Y.P.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing—original draft, Visualization. R.H.Y.P.: Conceptualization, Validation, Writing—review and editing. K.S.: Conceptualization, Methodology, Validation, Writing—review and editing. B.A.: Conceptualization, Methodology, Validation, Investigation, Resources, Funding acquisition, Project administration, Visualization, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A2B5B01001190). Beongku An is the corresponding author.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Accept cluster
CF-mMIMO	Cell-free massive MIMO
CH	Cluster head
CHI	Cluster head information
CM	Cluster member
DL	Deep learning
DLSMR	Deep learning secure multicast routing
DNN	Deep neural network
FANET	Flying ad hoc network
FN	Flying nodes
GN	Ground nodes
INFO	Information
JC	Join cluster
MAODV	Multicast ad hoc on-demand distance vector
MRREQ	Multicast route request
PDR	Packet delivery ratio
PSO	Particle swarm optimization
RREP	Route reply
RMSE	Root mean square error
RSSI	Received signal strength indicator
RWP	Random waypoint mobility
RPGM	Reference point group mobility
TD-PSO	Top-down particle swarm optimization
UAV	Unmanned aerial vehicle
UGW	UAV gateway node

References

1. Geraci, G.; Garcia-Rodriguez, A.; Giordano, L.G.; Lopez-Perez, D.; Bjoernson, E. Supporting UAV Cellular Communications through Massive MIMO. In Proceedings of the 2018 IEEE International Conference on Communications (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
2. Garcia-Rodriguez, A.; Geraci, G.; Lopez-Perez, D.; Giordano, L.G.; Ding, M.; Bjornson, E. The Essential Guide to Realizing 5G-Connected UAVs with Massive MIMO. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
3. Shumeye Lakew, D.; Sa'ad, U.; Dao, N.N.; Na, W.; Cho, S. Routing in Flying Ad Hoc Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1071–1120. [CrossRef]
4. Cho, J.; Sung, J.; Yoon, J.; Lee, H. Towards Persistent Surveillance and Reconnaissance Using a Connected Swarm of Multiple UAVs. *IEEE Access* **2020**, *8*, 157906–157917. [CrossRef]
5. Lin, M.; Chen, T.; Ren, B.; Chen, H.; Zhang, M.; Guo, D. CADer: A Deep Reinforcement Learning Approach for Designing the Communication Architecture of System of Systems. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3405–3417. [CrossRef]

6. Wu, Q.; Xu, J.; Zeng, Y.; Ng, D.W.K.; Al-Dhahir, N.; Schober, R.; Swindlehurst, A.L. A Comprehensive Overview on 5G-and-Beyond Networks with UAVs: From Communications to Sensing and Intelligence. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2912–2945. [CrossRef]
7. Lee, R.K.; Kitts, C.A.; Neumann, M.A.; Mcdonald, R.T. Multiple UAV Adaptive Navigation for Three-Dimensional Scalar Fields. *IEEE Access* **2021**, *9*, 122626–122654. [CrossRef]
8. Zhao, N.; Lu, W.; Sheng, M.; Chen, Y.; Tang, J.; Yu, F.R.; Wong, K.K. UAV-Assisted Emergency Networks in Disasters. *IEEE Wirel. Commun.* **2019**, *26*, 45–51. [CrossRef]
9. Al-Emadi, S.; Al-Mohannadi, A. Towards Enhancement of Network Communication Architectures and Routing Protocols for FANETs: A Survey. In Proceedings of the 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, 4–6 September 2020; pp. 1–10.
10. Chriki, A.; Touati, H.; Snoussi, H.; Kamoun, F. FANET: Communication, mobility models and security issues. *Comput. Netw.* **2019**, *163*, 106877. [CrossRef]
11. Yang, X.; Yu, T.; Chen, Z.; Yang, J.; Hu, J.; Wu, Y. An Improved Weighted and Location-Based Clustering Scheme for Flying Ad Hoc Networks. *Sensors* **2022**, *22*, 3236. [CrossRef]
12. Arafat, M.Y.; Moh, S. Localization and Clustering Based on Swarm Intelligence in UAV Networks for Emergency Communications. *IEEE Internet Things J.* **2019**, *6*, 8958–8976. [CrossRef]
13. Zhang, J.; Feng, X.; Liu, Z. A Grid-Based Clustering Algorithm via Load Analysis for Industrial Internet of Things. *IEEE Access* **2018**, *6*, 13117–13128. [CrossRef]
14. Ashour, O.; St-Hilaire, M.; Kunz, T.; Wang, M. A Survey of Applying Reinforcement Learning Techniques to Multicast Routing. In Proceedings of the 2019 IEEE 10th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 10–12 October 2019; pp. 1145–1151.
15. Yadav, A.; Tripathi, S. QMRPRNS: Design of QoS multicast routing protocol using reliable node selection scheme for MANETs. *Peer-Netw. Appl.* **2017**, *10*, 897–909. [CrossRef]
16. Motamedi, M.; Yazdani, N. Detection of black hole attack in wireless sensor network using UAV. In Proceedings of the 2015 7th Conference on Information and Knowledge Technology (IKT), Urmia, Iran, 26–28 May 2015; pp. 1–5.
17. Agron, D.J.S.; Ramli, M.R.; Lee, J.M.; Kim, D.S. Secure Ground Control Station-based Routing Protocol for UAV Networks. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 16–18 October 2019; pp. 794–798.
18. Munawar, S.; Ali, Z.; Waqas, M.; Tu, S.; Hassan, S.A.; Abbas, G. Cooperative Computational Offloading in Mobile Edge Computing for Vehicles: A Model-Based DNN Approach. *IEEE Trans. Veh. Technol.* **2023**, *72*, 3376–3391. [CrossRef]
19. Tran, T.N.; Nguyen, T.V.; Shim, K.; da Costa, D.B.; An, B. A Deep Reinforcement Learning-Based QoS Routing Protocol Exploiting Cross-Layer Design in Cognitive Radio Mobile Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 13165–13181. [CrossRef]
20. Khan, A.; Aftab, F.; Zhang, Z. BICSF: Bio-Inspired Clustering Scheme for FANETs. *IEEE Access* **2019**, *7*, 31446–31456. [CrossRef]
21. Khan, A.; Khan, S.; Shahzad, F.; Zhang, Z.; Abuassba, A. Intelligent Cluster Routing Scheme for Flying Ad Hoc Networks. *Sci. China Inf. Sci.* **2021**, *64*, 182305. [CrossRef]
22. Arafat, M.Y.; Moh, S. Bio-Inspired Approaches for Energy-Efficient Localization and Clustering in UAV Networks for Monitoring Wildfires in Remote Areas. *IEEE Access* **2021**, *9*, 18649–18669. [CrossRef]
23. Pramitarini, Y.; Hendra, R.; Perdana, Y.; Shim, K.; An, B. Exploiting TAS schemes to Enhance the PHY-security in Cooperative NOMA Networks : A Deep Learning Approach. In Proceedings of the 2023 International Conference on Artificial Intelligence in Information and Communication, Bali, Indonesia, 27–29 December 2023; pp. 199–204.
24. Hussen, H.R.; Choi, S.C.; Park, J.H.; Kim, J. Predictive geographic multicast routing protocol in flying ad hoc networks. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719843879. [CrossRef]
25. Sharma, V.; Kumar, R.; Kumar, N. DPTR: Distributed priority tree-based routing protocol for FANETs. *Comput. Commun.* **2018**, *122*, 129–151. [CrossRef]
26. Cheriguene, Y.; Djellikh, S.; Bousbaa, F.Z.; Lagraa, N.; Lakas, A.; Kerrache, C.A.; Karim Tahari, A.E. SEMRP: An Energy-efficient Multicast Routing Protocol for UAV Swarms. In Proceedings of the 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Prague, Czech Republic, 14–16 September 2020; pp. 1–8.
27. Luo, X.; Chen, Y.; Li, M.; Luo, Q.; Xue, K.; Liu, S.; Chen, L. CREDND: A Novel Secure Neighbor Discovery Algorithm for Wormhole Attack. *IEEE Access* **2019**, *7*, 18194–18205. [CrossRef]
28. Jamali, S.; Fotohi, R. DAWA: Defending against wormhole attack in MANETs by using fuzzy logic and artificial immune system. *J. Supercomput.* **2017**, *73*, 5173–5196. [CrossRef]
29. Pramitarini, Y.; Perdana, R.H.Y.; Tran, T.N.; Shim, K.; An, B. A Hybrid Price Auction-Based Secure Routing Protocol Using Advanced Speed and Cosine Similarity-Based Clustering against Sinkhole Attack in VANETs. *Sensors* **2022**, *22*, 5811. [CrossRef]
30. Liu, D.; Zhang, J.; Cui, J.; Ng, S.X.; Maunder, R.G.; Hanzo, L. Deep-Learning-Aided Packet Routing in Aeronautical Ad Hoc Networks Relying on Real Flight Data: From Single-Objective to Near-Pareto Multiobjective Optimization. *IEEE Internet Things J.* **2022**, *9*, 4598–4614. [CrossRef]
31. Lu, J.; He, D.; Wang, Z. Secure Routing in Multihop Ad-Hoc Networks With SRR-Based Reinforcement Learning. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 362–366. [CrossRef]

32. Tahboush, M.; Agoyi, M. A Hybrid Wormhole Attack Detection in Mobile Ad-Hoc Network (MANET). *IEEE Access* **2021**, *9*, 11872–11883. [CrossRef]
33. Bhosale, S.A.; Sonavane, S.S. Wormhole Attack Detection System for IoT Network: A Hybrid Approach. *Wirel. Pers. Commun.* **2022**, *124*, 1081–1108. [CrossRef]
34. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the Proc. ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
35. Zhang, Y.; Balochian, S.; Agarwal, P.; Bhatnagar, V.; Housheya, O.J. Artificial intelligence and its applications. *Math. Probl. Eng.* **2014**, 840491. [CrossRef]
36. Zhang, Y.; Shuihua, W.; Genlin, J. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Math. Probl. Eng.* **2015**, 931256. [CrossRef]
37. Nahar, A.; Sikarwar, H.; Das, D. CSBR: A Cosine Similarity Based Selective Broadcast Routing Protocol for Vehicular Ad-Hoc Networks. In Proceedings of the 2020 IFIP Networking Conference (Networking), Virtual, 20–25 June 2020; pp. 404–412.
38. Abosamra, G.; Oqaibi, H. Using Residual Networks and Cosine Distance-Based K-NN Algorithm to Recognize On-Line Signatures. *IEEE Access* **2021**, *9*, 54962–54977. [CrossRef]
39. Bhatia, T.K.; Tyagi, S.; Gusain, A.; Sharma, K. A Study on the Flying Ad-hoc Networks: Related Challenges, Routing Protocols and Mobility Models. In Proceedings of the 2022 11th International Conference on System Modeling & Advancement in Research Trends, SMART 2022, Moradabad, India, 16–17 December 2022; pp. 438–444.
40. Papavassiliou, S.; An, B. Supporting multicasting in mobile ad hoc wireless networks: Issues, challenges, and current protocols. *Wirel. Commun. Mob. Comput.* **2002**, *2*, 115–130. [CrossRef]
41. Halder, T.; Das, A.K.; Banerjee, S.; Sarkar, S.; Basak, A.; Ray, A.M.; Chakravarty, D. A Hybrid Localization Approach for UAV based on AODV and RSSI in FANETs. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 7914–7917.
42. Pramitarini, Y.; Perdana, R.H.Y.; Shim, K.; An, B. Particle Swarm Optimization-based Clustering Algorithm to Support QoS Routing Protocol in Flying Ad-hoc Networks with CF-mMIMO. In Proceedings of the 11th International Conference on Human Interaction and Emerging Technologies, Bangkok, Thailand, 31 January–2 February 2023; pp. 214–219.