ETD: An Efficient Time Delay Attack Detection Framework for UAV Networks

Wenbin Zhai, Liang Liu, Youwei Ding, Shanshan Sun, and Ying Gu

***Abstract*— In recent years, Unmanned Aerial Vehicle (UAV) networks are widely used in both military and civilian scenarios. However, due to the distributed nature, they are also vulnerable to threats from adversaries. Time delay attack is a type of internal attack which maliciously delays the transmission of data packets and further causes great damage to UAV networks. Furthermore, it is easy to implement and difficult to detect due to the avoidance of packet modification and the unique characteristics of UAV networks. However, to the best of our knowledge, there is no research on time delay attack detection in UAV networks. In this paper, we propose an Efficient Time Delay Attack Detection Framework (ETD). First, we collect and select delay-related features from four different dimensions, namely delay, node, message and connection. Meanwhile, we utilize the pre-planned trajectory information to accurately calculate the real forwarding delay of nodes. Then, one-class classification is used to train the detection model, and the forwarding behaviors of all nodes can be evaluated, based on which their trust values can be obtained. Finally, the K-Means clustering method is used to distinguish malicious nodes from benign ones according to their trust values. Through extensive simulation, we demonstrate that ETD can achieve higher than 80% detection accuracy with less than 2.5% extra overhead in various settings of UAV networks and different routing protocols.**

***Index Terms*— Time delay attack, UAV networks, lightweight, one-class classification, trajectory information, K-means clustering.**

I. Introduction

UNMANNED Aerial Vehicle (UAV) networks are widely used in both military and civilian scenarios such as battlefield surveillance, disaster response, farmland monitoring, etc [1]. UAVs often cooperate with each other to collect data in the form of clusters, and the ground station gathers data from UAVs for further processing. The data packet transmission between remote power-constrained UAVs and the ground station is generally made over multiple hops, thus forming a multi-hop UAV network. Meanwhile, many multi-hop routing protocols for UAV networks have been proposed to efficiently deliver packets to the destination, such as Epidemic routing [2], Spray and Wait routing [3], Probabilistic routing [4] and MaxProp routing [5].

Multi-hop UAV networks are flexible, however, they also suffer from many security threats, including external and internal threats [6]. Compared with external threats, internal threats are more challenging and destructive for UAV networks, attackers can launch an intrusion inside the network. For instance, attackers can invade some UAVs in the network and then use these compromised UAVs to attack the network, such as packet drop attack, flood attack, replay attack and tamper attack [7]. Attacks launched by invaded internal UAVs (i.e., internal attacks) could not be solely resolved by conventional security mechanisms, such as cryptographic approaches [8].

Time delay attack is a type of internal attack where malicious nodes deliberately delay the transmission of received packets before forwarding them to the destination. In UAV networks, time delay attack has the following two key characteristics: (1) It is ubiquitous and harmful to UAV networks, especially in time-sensitive application scenarios. Taking the military reconnaissance and strike mission as an example, if packets containing the target information are maliciously delayed, wrong decisions may be made, and even the mission will fail. (2) It is easy to implement and difficult to detect. Unlike packet tamper attack, time delay attack does not need to break the cryptographic protection and modify the packets. Meanwhile, compared with packet drop, replay and flood attacks, time delay attack is more covert, making it difficult to detect.

However, in recent years, existing works on internal attack detection in UAV networks focus on packet drop attack, flood attack, replay attack and tamper attack [9], [10], [11]. To the best of our knowledge, there is no research on time delay attack in UAV networks. Although there are a few studies on time delay attack in other networks [12], [13], [14], [15], UAV networks have many unique characteristics, such as high mobility, sparse distribution, intermittent connectivity, unstable link quality and store-carry-forward (SCF) mechanism, which cause the existing time delay attack detection approaches inefficient and inapplicable in UAV networks.

Manuscript received 22 January 2022; revised 20 January 2023 and 18 February 2023; accepted 18 April 2023. Date of publication 3 May 2023; date of current version 12 May 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2700500 and Grant 2021YFB2700502; in part by the Open Fund of Key Laboratory of Civil Aviation Smart Airport Theory and System, Civil Aviation University of China, under Grant SATS202206; in part by the National Natural Science Foundation of China under Grant U20B2050 and Grant 82004499; in part by the Natural Science Foundation of Jiangsu Province under Grant BE2020106; in part by the Public Service Platform for Basic Software and Hardware Supply Chain Guarantee under Grant TC210804A; and in part by the Fund of Key Laboratory of Complex Electronic System Simulation under Grant 614201002022205. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Edgar Weippl. *(Corresponding author: Liang Liu.)*

Wenbin Zhai, Liang Liu, and Shanshan Sun are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: wenbinzhai@nuaa.edu.cn; liangliu@nuaa.edu.cn; sunshanshan@nuaa.edu.cn).

Youwei Ding is with the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing 210023, China (e-mail: ywding@njucm.edu.cn).

Ying Gu is with the School of Engineering and Applied Sciences, Columbia University, New York, NY 10032 USA (e-mail: yinggu97@gmail.com).

Digital Object Identifier 10.1109/TIFS.2023.3272862

We illustrate time delay attack in UAV networks in more detail. As shown in Fig. 1, there are two UAVs u_1, u_2 and a ground station g_0. At t_1, UAV u_1 sends a data packet m with a delay constraint t_d to g_0. As shown in Fig. 1(a), since u_2 is closer to the destination g_0 than u_1 and u_1 can communicate with u_2, u_1 will send m to u_2 at t_1, and u_2 will receive m from u_1 at t_2. Then, since there is no UAV which can communicate with u_2 at t_2, u_2 will *store and carry* m until it encounters g_0 at t_3. If u_2 is a benign node, it will *forward* m to g_0 at t_3 and g_0 will receive m at t_4, and the delivery delay of m is $t_4 - t_1$, as shown in Fig. 1(c). However, if u_2 is a malicious node and launches a time delay attack, as shown in Fig. 1(b), m will be maliciously delayed by τs. Since u_2 can still communicate with g_0 at $t_3{}'$ (i.e., $t_3 + \tau$), u_2 will send m at $t_3{}'$ and g_0 will receive m at $t_4{}'$ (i.e., $t_4 + \tau$), the delivery delay of m will become $t_4{}' - t_1$ (i.e., $t_4 - t_1 + \tau$), as shown in Fig. 1(d). If $t_4 - t_1 \leq t_d \leq t_4{}' - t_1$, the time delay attack will cause the packet m to not be delivered in time.

The unique characteristics make time delay attack detection in UAV networks face the following challenges: (1) Due to the intermittent connectivity and sparse distribution, a relatively short malicious delay introduced by the attacker may not cause obviously abnormal fluctuations of the delivery delay. (2) Due to the SCF mechanism, the malicious time delay attack is likely to be misjudged as the normal SCF behavior of UAVs. (3) Due to the complex architecture and high dynamics, there are many factors related to the forwarding delay, such as the load of the forwarding nodes (e.g., the queuing delay of the packet), the link quality and so on. It is difficult to construct the mathematical models between these factors and the corresponding forwarding delay.

To overcome the above challenges, in this paper, we propose an Efficient Time Delay Attack Detection Framework (ETD). First, in order to detect time delay attack accurately and efficiently, we evaluate the forwarding delay of nodes rather than the delivery delay of messages. Meanwhile, we utilize the pre-planned trajectory information of UAVs to eliminate the adverse impact of the duration that UAVs store and carry the packets, based on which the real forwarding delay of nodes can be estimated. In addition, we select delay-related features from four different dimensions, namely delay, node, message and connection.

Then, since there are many delay-related features in UAV networks and the collection of malicious samples is not a trivial task, in this paper, one-class classification is utilized for the detection model training without the requirement of samples of anomalous behaviors. With the trained detection model, each forwarding behavior of the node will be evaluated, and then the trust value of each node in the network can be calculated. Finally, the K-Means clustering method is further utilized to distinguish malicious nodes from benign ones according to their trust values. We summarize our key contributions as follows:

- We model time delay attack in UAV networks and demonstrate its uniqueness, concealment and destructiveness. As far as we know, we are the first to study time delay attack in UAV networks.
- We propose an Efficient Time Delay Attack Detection Framework (ETD). First, we select delay-related features from four different dimensions. Meanwhile, the pre-planned trajectory information is utilized to evaluate the real forwarding delay of nodes. Based on these delay-related features and the real forwarding delay, one-class classification is used for model training, and then the K-Means clustering method is further utilized to identify malicious nodes.
- We implement extensive simulations on the Opportunistic Network Environment (ONE) simulator [16]. The experimental results show that ETD can achieve higher than 80% detection accuracy with less than 2.5% extra overhead in various settings of UAV networks and different routing protocols.

The remainder of the paper is organized as follows. Section II summarizes state-of-the-art in malicious node detection. Section III formalizes the system model, including the network model and attack model. In Section IV, our proposed ETD is described in detail. We provide the performance evaluation of ETD through extensive simulations in Section V, and conclude this paper in Section VI.

II. Related Work

In this section, we first introduce some internal cyberattacks in UAV networks, and to the best of our knowledge, there has been no research on time delay attack detection in UAV networks. Subsequently, we summarize state-of-the-art on time delay attack detection in static networks, such as Cyber-physical systems (CPSes) and Precision Time Protocol (PTP), and then illustrate their shortcomings used for UAV networks. Table I outlines the internal attacks discussed in the existing surveys.

A. Attacks in UAV Networks

With the rapid development of wireless communication technology, UAV networks are praised for the flexibility and scalability, and have become a research hotspot in recent years. However, they are also vulnerable to various attacks, which can be divided into internal and external attacks according to the source of the attack. External attacks are carried out by unauthorized users [17], whereas internal attacks are launched by legitimate but malicious nodes inside the network [18]. Therefore, time delay attack belongs to internal attacks, however, existing works on internal attack detection in UAV networks focus on packet drop attack, flood attack, replay attack and tamper attack [19].

The authors in [9] study a colluding packet drop attack, in which attackers cooperate with each other to launch attacks and cover up their misbehavior. They utilize the recorded encounter information and forwarding ratio of nodes to suspect and then further confirm malicious nodes in the network. In [11], three types of flood attacks are discussed and a trust-based approach is proposed to detect malicious nodes. The misbehavior of malicious nodes will be manifested and lead to the loss of their reputation metrics.

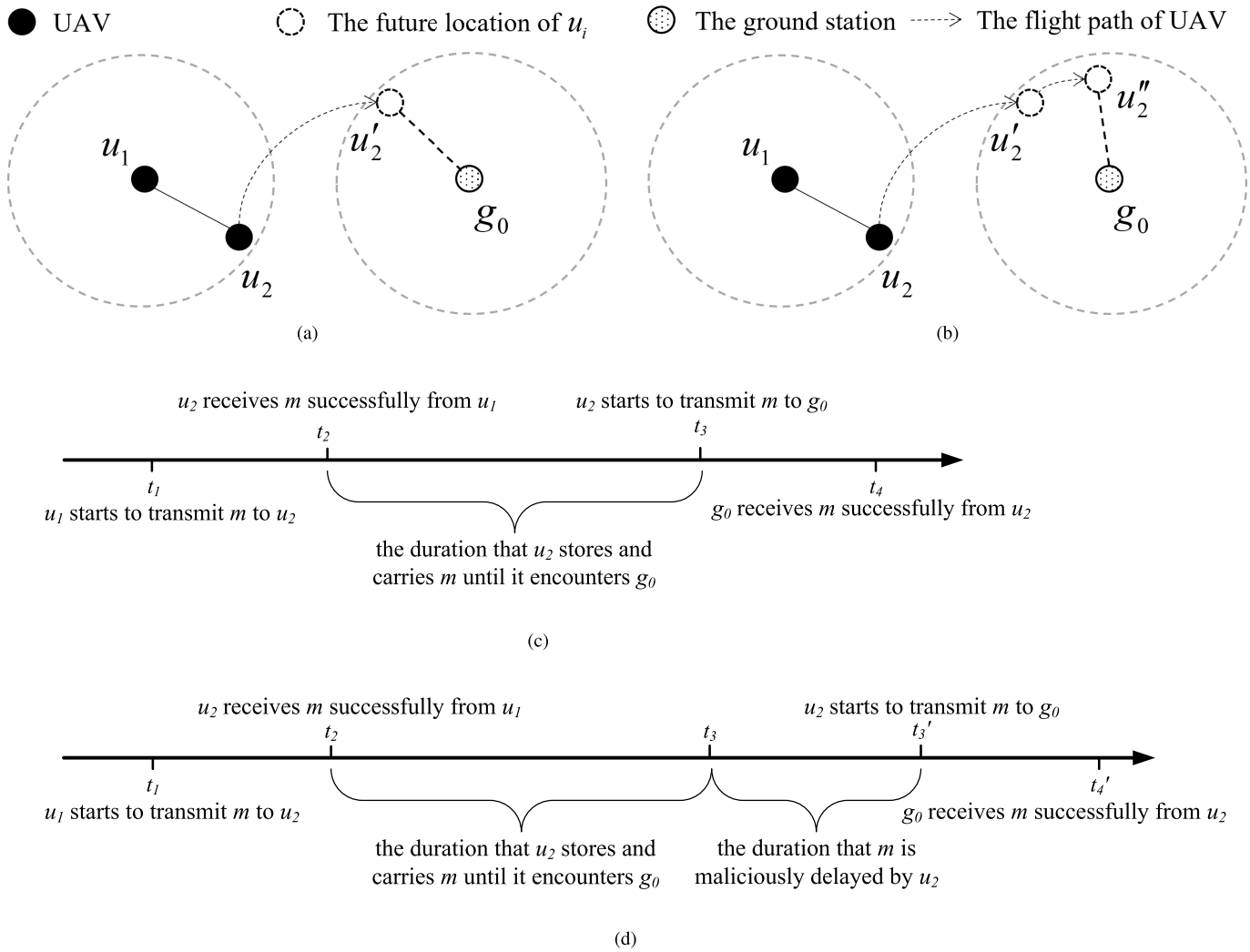


Fig. 1. Examples of the normal and abnormal forwarding behaviors in the UAV network. u_1 and g_0 are benign nodes. (a) The forwarding behaviors when u_2 is a benign node. (b) The forwarding behaviors when u_2 is a malicious node. (c) The timeline of forwarding behaviors when u_2 is a benign node. (d) The timeline of forwarding behaviors when u_2 is a malicious node.

In [20], [21], and [22], the authors consider a hybrid attack in which attackers can launch packet drop attack, tamper attack and replay attack simultaneously. They utilize the information exchange between nodes to evaluate the trustworthiness of nodes, and then the K-Means clustering is used to distinguish nodes into benign and malicious ones. The authors in [23] consider an advanced attack where malicious nodes only launch the above hybrid attack on data packets sent to specific neighbor nodes. They reduce the reputation model of all nodes and edges into a multiple linear regression problem, and then use the support vector machine (SVM) algorithm to identify malicious edges, thereby further confirming malicious nodes. In [24], an intelligent attack is proposed, in which adversaries only implement the hybrid attack on data packets that satisfy certain conditions. Regression and clustering algorithms are used to evaluate the trustworthiness of nodes and distinguish malicious nodes from benign ones.

However, as far as we know, there is no research on time delay attack in UAV networks. Compared with other internal attacks (e.g., packet drop, flood, replay and tamper attacks), time delay attack is easier to implement and more difficult to detect, as it avoids the manipulation of packets.

B. Time Delay Attack Detection in Other Networks

Although there is no research on time delay attack in UAV networks, in recent years, time delay attack has attracted great attention from researchers due to its concealment and great threat, and have been widely studied in other networks, such as wired networks and static wireless sensor networks (WSNs).

CPSes [25] are classic time-sensitive systems, which are usually in the form of wired or wireless static networks. The control signals in CPSes have stringent timeliness requirements, which makes CPSes vulnerable to time delay attack. Many approaches have been proposed to detect time delay attack in CPSes. The authors in [12] propose a perturbation term which estimates the measurement deviation of the load and the frequency, and then use it to detect the time delay attack. In [14], Machining Learning (ML) is used to evaluate the impact of time delay attacks on system stability and security, and then two-tiered mitigation measures are developed correspondingly to detect and defend the attack. The authors

in [15] utilize the Recurrent Neural Networks (RNN) to access the effect of time delay attack, then detect and characterize the attack. Meanwhile, a deep learning model is used to efficiently process the long-term sequence data. Based on this work, in [26], the authors further improve the practicability of the system by real-time processing and online analysis of data from CPS sensors. Moreover, different strategies of the detection model are proposed which can be adjusted dynamically based on different objectives. In addition, other types of internal attacks in CPSes are summarized in [27], [28], [29], [30], [31], and [32].

PTP [33] is a precise time synchronization protocol introduced in IEEE 1588 Standard. It is mainly used for the time synchronization in packet-based networks, and can achieve sub-microsecond transmission and synchronization accuracy, which also makes them vulnerable to time delay attack. In order to ensure its security, detection mechanisms against time delay attack for PTP are needed. The authors in [13] conduct a quantitative analysis of time delay attack and show the vulnerability of PTP to the attack. Then, for detection and mitigation, a new type of the PTP clock is utilized to response and mitigate time delay attack. The authors in [34] use the redundant paths and participants between the primary clock and the secondary clocks to calculate the relative offset rate and time of the secondary clocks. The clocks that drift faster and more will be suspected under attack. Furthermore, some other types of internal attacks in PTP are summarized in [35], [36], [37], [38], [39], and [40].

However, the unique characteristics and SCF mechanism of UAV networks make time delay attack more covert, complicated and destructive than that in other networks, which causes the existing detection approaches inefficient and inapplicable. Therefore, it is significant to study time delay attack detection in UAV networks.

III. System Model

In this section, we formulate our system model. First, the UAV network model is described, including the node model and transmission path model. Second, we propose and analyze the time delay attack model in UAV networks, which is different from that in conventional wired networks and static WSNs.

A. Network Model

The application scenarios we consider in this paper are search and rescue missions. Many UAVs searching in an area will send data packets back to the ground station as needed. Without loss of generality, we abstract the three-dimensional space into a Euclidean space, ignoring the vertical space [41]. The trajectories of UAVs are pre-planned and can be obtained in advance through mission planning and path planning [42], [43], [44]. Even if UAVs re-plan the trajectories during the mission, their trajectories can also be obtained by the ground station in advance [45], [46], [47]. Based on the pre-planned trajectories, the ground station can further calculate encounters between UAVs. For ease of representation, in this paper, we abstract the communication between UAVs as an encounter point [45], [46]. For example, as shown in Fig. 2, UAVs are flying along with their pre-planned trajectories, and UAV u_1 will encounter u_2 at position e_1 between $5s$ and $13s$, which means u_1 and u_2 can communicate with each other between $5s$ and $13s$, and so on.

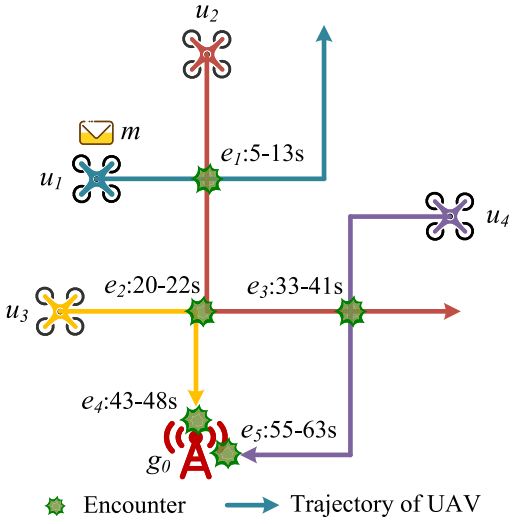


Fig. 2. An example of the UAV network.

1) Node Model: In this paper, malicious nodes in UAV networks can launch time delay attack with a certain probability. The ground station is a trusted authority that collects data packets from UAVs. For convenience, both "UAV" and "node" represent a UAV. A node can be represented as:

$$Node =< id, P_{TDA} > \tag{1}$$

where id represents the unique identifier of the node, such as u_1, u_2 in Fig. 2, and P_{TDA} is the probability of the node carrying out the time delay attack. For a benign node, $P_{TDA} = 0$, whereas $0 < P_{TDA} \leq 1$ when it is a malicious node.

2) Path Model: The transmission path of a data packet m can be represented as:

$$\begin{aligned} Path = \langle &(node_1, node_2, t_1^s, t_2^r), (node_2, node_3, t_2^s, t_3^r), \ldots, \\ &(node_i, node_{i+1}, t_i^s, t_{i+1}^r), \ldots, \\ &(node_n, node_{n+1}, t_n^s, t_{n+1}^r)\rangle \end{aligned} \tag{2}$$

where t_i^s represents the time when the node $node_i$ starts to send m to $node_{i+1}$; t_{i+1}^r represents the time when $node_{i+1}$ successfully receives m from $node_i$. Meanwhile, $t_{i+1}^r = t_i^s + t_{trans}$, where t_{trans} denotes the duration to transmit m from $node_i$ to $node_{i+1}$ successfully. For convenience, in this paper, we assume that $t_{trans} = 1s$. For example, in Fig. 2, at the beginning (i.e., $0s$), UAV u_1 generates a data packet m and sends it to the ground station g_0. According to the pre-planned trajectory information and encounter situation, we can deduce that there is a transmission path: $\langle(u_1, u_2, 5, 6), (u_2, u_3, 20, 21), (u_3, g_0, 43, 44)\rangle$, which means u_1 encounters u_2 and transmits m to u_2 at position e_1 between $5s$ and $6s$; and then u_2 will store and carry m until it encounters u_3 and transmits m to u_3 at position e_2 between $20s$ and $21s$; finally u_3 encounters g_0 and transmits m to g_0 at position e_4 between $43s$ and $44s$.

TABLE I
COVERAGE OF CYBERATTACKS AND DETECTIONS IN EXISTING SURVEYS

	drop attacks	flood attacks	tamper attacks	replay attacks	hybrid attacks	time delay attacks
UAV networks	[9], [20]–[22]	[11]	[20]–[22]	[20]–[22]	[23], [24]	✗
CPSes	[27]	[28]	[29]	[30]	[31]	[12], [14], [15], [26]
PTP	[35]	[36]	[37], [38]	[39]	[40]	[13], [33], [34]

B. Time Delay Attack Model

We assume that attackers have the ability to intercept UAVs to launch attacks. In this paper, we consider a covert attack, called *time delay attack*. The time delay attack maliciously delays the transmission of data packets without tampering with packets, and it can be easily carried out by a malicious UAV.

The time delay attack is formally described as follows. In time delay attack, the transmission of the data packets is maliciously delayed by τs. Let $t_i^{s\prime}$ denote the delayed time when the malicious node $node_i$ starts to transmit m to $node_{i+1}$, and $t_{i+1}^{r\prime}$ denote the delayed time when $node_{i+1}$ successfully receives m from the malicious node $node_i$. In conventional wired networks and static WSNs [13], [14], [15], the time delay attack model can be formalized as:

$$t_i^{s\prime} = t_i^s + \tau \tag{3}$$

$$t_{i+1}^{r\prime} = t_{i+1}^r + \tau \tag{4}$$

However, these two equations are not always true in UAV networks. For example, in Fig. 2, at the beginning, UAV u_1 generates a data packet m and sends it to g_0. For convenience, we assume that each message-holder UAV transmits m to the first UAV it will encounter. According to the pre-planned trajectory information and encounter situation, when there is no malicious nodes, the transmission path is $\langle(u_1, u_2, 5, 6), (u_2, u_3, 20, 21), (u_3, g_0, 43, 44)\rangle$.

Then, we assume that u_2 is a malicious node and carries out a time delay attack, namely $u_2.P_{TDA} = 1$. When the duration of the attack $\tau = 1s$, the transmission path becomes $\langle(u_1, u_2, 5, 6), (u_2, u_3, 21, 22), (u_3, g_0, 43, 44)\rangle$.

When $\tau = 3s$, according to (3), $t_2^{s\prime}$ should be $23s$, however, as Fig. 2 shows, there is no UAV which can communicate with u_2 at $23s$. UAV u_2 has to store and carry the packet m until it encounters u_4 at $33s$, and then transmits it to u_4 after another time delay attack. Therefore, the transmission path will be $\langle(u_1, u_2, 5, 6), (u_2, u_4, 36, 37), (u_4, g_0, 55, 56)\rangle$. Here, $t_2^{s\prime} = 36s \gg 23s$. In this situation, it is worth noting that the time delay attack also changes the original transmission path, which causes greater damage. Although the duration of the attack is $3s$, the delivery delay of m is increased by $12s \gg 3s$. As the above example shows, time delay attack in UAV networks is more destructive.

When $\tau = 10s$, we found that there is no transmission path along which u_1 can deliver m to g_0. In this situation, time delay attack is equivalent to packet drop attack. However, different from packet drop attack, whose target is to drop the received data packets randomly with specific malicious goals, time delay attack is to prevent the timely delivery of data packets, thereby degrading the UAV network performance and even causing damage to the network. Moreover, compared with packet drop attack which can be easily detected [10], [21], time delay attack is stealthier.

To sum up, different from that in conventional wired networks and static WSNs, the time delay attack model in UAV networks can be formalized as

$$t_i^{s\prime} = \begin{cases} t_i^s + \tau, & \text{if } \tau + t_{trans} \le t_{dur(i,i+1)} \\ t'_{ste(i,i+1)} + \tau, & \text{otherwise.} \end{cases} \tag{5}$$

$$t_{i+1}^{r\prime} = t_i^{s\prime} + t_{trans} \tag{6}$$

where $t_{dur(i,i+1)}$ denotes the duration of the encounter between $node_i$ and $node_{i+1}$, $t'_{ste(i,i+1)}$ represents the start time of the encounter between $node_i$ and $node'_{i+1}$, where $node'_{i+1}$ is the suitable next-hop node of $node_i$ according to the routing protocol [48] and $\tau + t_{trans} \le t'_{dur(i,i+1)}$.

IV. ETD

A. Overview

Fig. 3 shows the overview of ETD. It consists of four phases: information collection, feature selection, model training and malicious node detection.

1) Information Collection (Section. IV-B): The transmitted messages are used for information collection. Specifically, while forwarding messages, UAVs will attach the delay-related information, such as the receiving time and the sending time, to the messages. The information will be finally received and collected at the ground station along with the messages for further processing.

2) Feature Selection (Section. IV-C): After receiving messages at the ground station, we perform a comprehensive analysis on these messages, then select appropriate delay-related features from four different dimensions (i.e., delay, node, message, and connection features). In addition, we further utilize the pre-planned trajectory information of UAVs to precisely calculate the real forwarding delay of nodes.

3) Model Training (Section. IV-D): Based on these delay-related features and the real forwarding delay, one-class classification is utilized for model training.

4) Malicious Node Detection (Section. IV-E): For each communication round, we use the trained detection model to evaluate each forwarding behavior of the node, based on which the trust value of each node can be calculated. Then, the K-Means clustering method is further used to distinguish malicious nodes from benign ones according to their trust values.

B. Information Collection

In this paper, for the collection of training samples and the identification of malicious nodes, UAVs will attach the

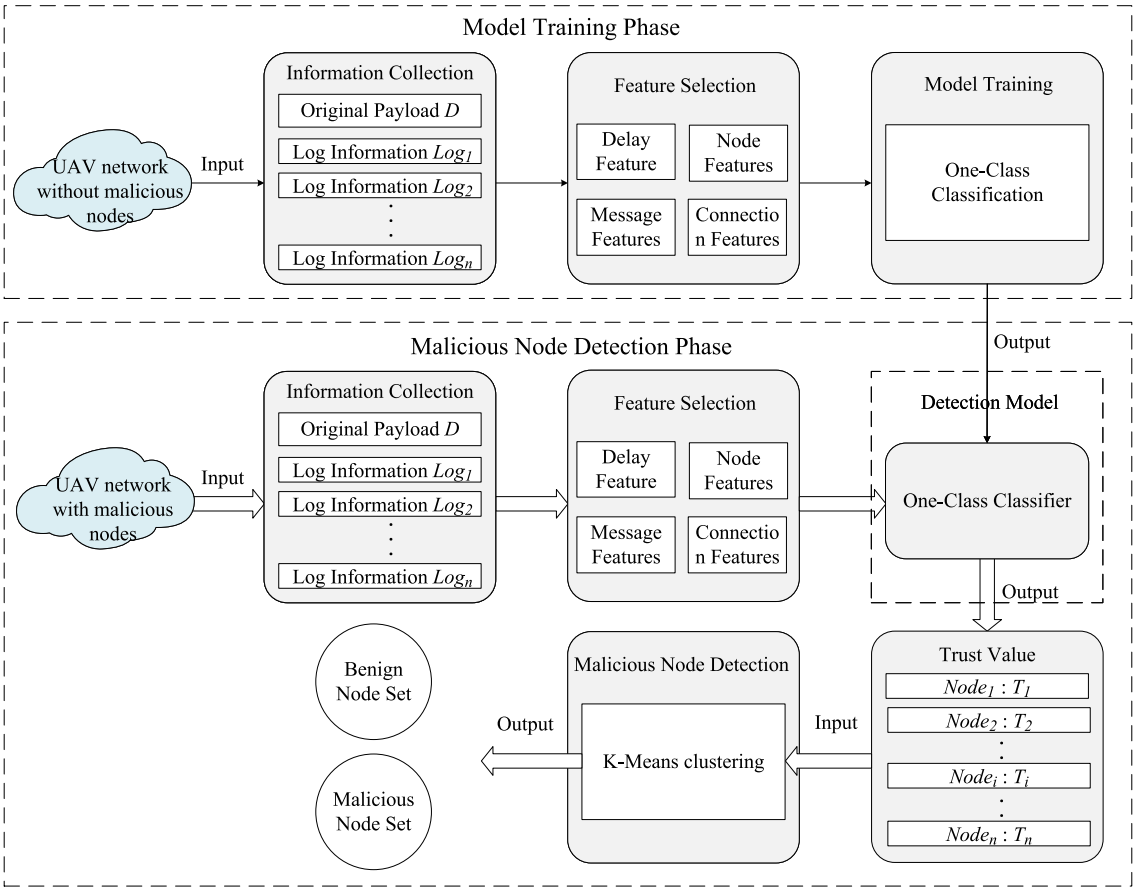


Fig. 3. The overview of ETD.

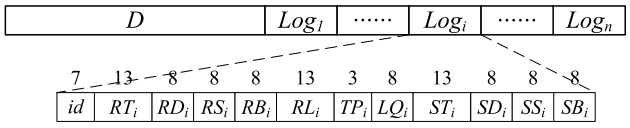


Fig. 4. The format of transmitted messages.

delay-related information to messages while forwarding the messages. Finally, these messages are received and analyzed at the ground station. The format of transmitted messages is shown in Fig. 4, where D is the original payload of the message m, Log_i denotes the transmission log information that $node_i$ attaches to m. Log_i records the delay-related information of $node_i$, where id is the unique identifier of $node_i$, RT_i is the time when $node_i$ receives m from $node_{i-1}$, RD_i represents the distance between $node_i$ and $node_{i-1}$ when m is received by $node_i$, RS_i and RB_i are the flight speed and buffer occupancy of $node_i$ when it receives m from $node_{i-1}$, and RS_i is a vector, indicating the flight direction and speed of $node_i$, RL_i is the remaining time to live (TTL) of m when it is received by $node_i$, TP_i and LQ_i indicate the transmission power and link quality of $node_i$. Similarly, ST_i, SD_i, SS_i and SB_i indicate the time, distance, flight speed and buffer occupancy of $node_i$ when it sends m to $node_{i+1}$.

It is worth noting that although some delay-related information is attached to the message, it is light-weight in saving. The following is an example of implementing the message. We use 7 bits to encode id, so the network can support 2^7 UAVs, then we assume that the network performs missions for at most two hours [49], so RT_i, RL_i and ST_i whose unit is second can be encoded by 13 bits. Next, we assume that the UAV has seven adjustable power levels, so TP_i can be encoded by 3 bits. Finally, in order to precisely reflect the distance, flight speed, buffer occupancy and link quality of communication parties, RD_i, RS_i, RB_i, LQ_i, SD_i, SS_i and SB_i are all encoded by 8 bits. Therefore, the transmission log information that each forwarding node attaches to the message can be encoded by $7 + 13 \times 3 + 3 + 8 \times 7 = 105$ bits ≈ 13 Bytes.

Meanwhile, the experimental results (see in Section. V-G) confirm the light-weight of our collection approach. The extra overhead introduced to the UAV network does not exceed 2.5% in all situations. In addition, the cost of storage and transmission can be further reduced if some efficient schemes are adopted. For instance, excepted for the source node to record the complete time stamp, other nodes on the transmission path can only record the relative time stamp to further reduce the extra cost.

C. Feature Selection

Due to the characteristics and unique SCF mechanism of UAV networks, time delay attack in UAV networks is covert and difficult to detect. Therefore, we need to explore the measures that possibly reveal the misbehavior of attackers in

the UAV network. Moreover, there are many factors which can influence the forwarding delay of messages. In addition, some information in the packet is not efficient enough to be directly used as the features, such as the sending time and the receiving time of each forwarding node.

Therefore, in this section, we perform a comprehensive analysis of the collected information, and then select delay-related features from four different dimensions, namely delay, node, message and connection features. Meanwhile, we further utilize the trajectory information of UAVs to accurately calculate the real forwarding delay of nodes.

1) Delay Feature: If attackers continue to maliciously delay received messages, the forwarding delay of the attackers will be abnormally long. Therefore, we can identify the forwarding delay of the attackers for received messages as the potential metric to distinguish them from benign nodes. However, due to the SCF mechanism, the forwarding delay we can directly collect from the network (e.g., $t_4' - t_1$, as shown in Fig. 1) includes the duration that UAVs store and carry the packets (e.g., $t_3 - t_2$ in Fig. 1). If we directly use the collected forwarding delay as a feature for the model training, this imprecise delay will affect the performance of the trained detection model.

To overcome this issue, in this section, we utilize the pre-planned trajectory information of UAVs to eliminate the adverse impact of storage and carry in order to accurately calculate the real forwarding delay, and then show how this feature can distinguish the forwarding behaviors of attackers from normal nodes.

In order to accurately evaluate the behavior of each forwarding node on the transmission path, for each data packet m that received at the ground station, we traverse the transmission path of m and extract all the two-hop sub-paths, which can be formalized as:

$$Path \Rightarrow \bigcup_{i=2}^{n} \langle (node_{i-1}, node_i, t_{i-1}^s, t_i^r), (node_i, node_{i+1}, t_i^s, t_{i+1}^r) \rangle \quad (7)$$

For example, in Fig. 2, at the beginning, UAV u_1 generates a data packet m and sends it to the ground station g_0. As mentioned in Section. III-B, when there is no malicious nodes in the network, the transmission path is $\langle (u_1, u_2, 5, 6), (u_2, u_3, 20, 21), (u_3, g_0, 43, 44) \rangle$. It can be split into two two-hop sub-paths: $\langle (u_1, u_2, 5, 6), (u_2, u_3, 20, 21) \rangle$ and $\langle (u_2, u_3, 20, 21), (u_3, g_0, 43, 44) \rangle$.

For each two-hop sub-path of the data packet m, denoted as $\langle (node_{i-1}, node_i, t_{i-1}^s, t_i^r), (node_i, node_{i+1}, t_i^s, t_{i+1}^r) \rangle$, we can utilize the trajectory information of $node_{i-1}$, $node_i$ and $node_{i+1}$ to predict encounters between them, and then compute the real forwarding delay of node $node_i$ to message m, denoted as t_{rfd}^i. The t_{rfd}^i is proposed to evaluate the forwarding behaviors of $node_i$ and can be formalized as

$$t_{rfd}^i = (t_{i+1}^r - t_{i-1}^s) - (t_{ste(i,i+1)} - t_i^r) \quad (8)$$

where $t_{ste(i,i+1)}$ represents the start time of the encounter between $node_i$ and $node_{i+1}$. For example, as mentioned above, for the two-hop sub-path $\langle (u_1, u_2, 5, 6), (u_2, u_3, 20, 21) \rangle$, u_2 is a benign node. As Fig. 2 shows, the start time of the encounter between u_2 and u_3 is $20s$. Therefore, $t_{rfd}^2 = (21s - 5s) - (20s - 6s) = 2s$.

Then, when $node_i$ is a malicious node and launches a time delay attack, the two-hop sub-path will become $\langle (node_{i-1}, node_i, t_{i-1}^s, t_i^r), (node_i, node_{i+1}', t_i^{s'}, t_{i+1}^{r'}) \rangle$. And the $t_{rfd}^{i'}$ can be formalized as

$$t_{rfd}^{i'} = (t_{i+1}^{r'} - t_{i-1}^s) - (t_{ste(i,i+1)} - t_i^r) \quad (9)$$

For instance, we assume that u_2 is a malicious node and launches a time delay attack. When the duration of the time delay attack $\tau = 1s$, the transmission path of m will become $\langle (u_1, u_2, 5, 6), (u_2, u_3, 21, 22), (u_3, g_0, 43, 44) \rangle$. Correspondingly, the extracted two-hop sub-path will become $\langle (u_1, u_2, 5, 6), (u_2, u_3, 21, 22) \rangle$ and $t_{rfd}^{2\,\prime} = (22s - 5s) - (20s - 6s) = 3s$. In addition, when the duration of the time delay attack τ is $3s$, the transmission path will become $\langle (u_1, u_2, 5, 6), (u_2, u_4, 36, 37), (u_4, g_0, 55, 56) \rangle$. In this situation, the corresponding two-hop sub-path will become $\langle (u_1, u_2, 5, 6), (u_2, u_4, 36, 37) \rangle$, which is different from the original path, and $t_{rfd}^{2\,\prime} = (37s - 5s) - (20s - 6s) = 18s \gg t_{rfd}^2 = 2s$.

The effect of launching a time delay attack τ on the real forwarding delay can be shown in (10).

$$\begin{aligned} t_{rfd}^{i'} &= (t_{i+1}^{r'} - t_{i-1}^s) - (t_{ste(i,i+1)} - t_i^r) \\ &\geq ((t_{i+1}^r + \tau) - t_{i-1}^s) - (t_{ste(i,i+1)} - t_i^r) \\ &\geq (t_{i+1}^r - t_{i-1}^s) - (t_{ste(i,i+1)} - t_i^r) + \tau \\ &\geq t_{rfd}^i + \tau \end{aligned} \quad (10)$$

If $node_i$ is an attacker and launches the time delay attack on the data packet m, the real forwarding delay of $node_i$ to m will reflect the abnormal forwarding behavior of $node_i$: t_{rfd}^i should be abnormally high, especially when the time delay attack changes the original transmission path.

In addition to the delay feature mentioned above, we also characterize the current network status in order to efficiently deal with the time delay attack in different environments. Therefore, we further analyze, extract and select delay-related features from three other dimensions: node, message and connection features.

2) Node Features: Due to the high mobility of nodes and high dynamic of topology in UAV networks, the misbehavior of malicious nodes will lead to the abnormal performance of themselves. Therefore, comprehensive analysis of node features can help the detection model better identify the abnormal performance of the nodes, and then use it as evidence to accuse the abnormal behaviors of malicious nodes.

For instance, attackers tend to delay rather than immediately forward the packets when encountering with other nodes, regardless of the occupancy of their buffers. This misbehavior makes the buffer occupancy of the attackers often higher than normal nodes. This phenomenon is more serious when the overhead of the UAV network is heavy. Meanwhile, due to the intermittent connectivity characteristic and SCF mechanism of UAV networks, time delay attack may lead to the malicious nodes storing and carrying more packets, which will increase

their load. Therefore, we can utilize the buffer size and buffer occupancy of the node to assist in distinguishing malicious nodes from benign ones.

Moreover, the transmission power of the node reflects the link quality and the communication range of the node, which is related to the propagation delay and can help eliminate the adverse impact of unstable link quality on the detection model. Based on the transmission power, we can further utilize the flight speed and direction of the UAV to estimate the link quality and the communication range in the future for a period of time [48].

As shown in Table II, the final accepted node features are $RxBufOcc$, $SndBufOcc$, $BufSize$, $RxDir$, $RxSpd$, $SndDir$, $SndSpd$ and $TxPwr$. It can be represented as

$$NFS = (RxBufOcc, SndBufOcc, BufSize, RxDir, RxSpd, SndDir, SndSpd, TxPwr) \quad (11)$$

3) Message Features: In UAV networks, UAVs continuously generate and forward messages as needed. Messages are the main entities of the network transmission and are closely related to the forwarding delay. The selection of delay-related features in the messages can efficiently and accurately evaluate the forwarding delay and identify malicious nodes.

For example, the transmission delay, which is the duration from the first digit to the last digit of the message leaving the sending node, depends on the packet size of the message. Moreover, selecting the source node, destination node and type of the message as features can better resist and identify various time delay attacks, such as intelligent attacks against specific source nodes [24], specific destination nodes [23] and specific messages types. Combining these features with other delay-related features can more accurately identify the abnormal behaviors of malicious nodes.

At the same time, the utilization of TTL of the message can further assist the real forwarding delay t^i_{rfd} to evaluate the forwarding delay of the node, thereby distinguishing the malicious nodes from normal ones.

Therefore, as shown in Table II, in ETD, we select the following message features: $MsgSize$, $RemTTL$, $MsgSrc$, $MsgDst$ and $MsgType$, which can be represented as

$$MFS = (MsgSize, RemTTL, MsgSrc, MsgDst, MsgType) \quad (12)$$

4) Connection Features: Due to the high dynamic of topology and intermittent connectivity of communications, the transmission of data packets in UAV networks depends on the dynamic connections between nodes. Therefore, we need to analyze the connection features between UAVs, which can reflect the current overall topology of the network and the trend of future topology. Meanwhile, due to the unstable link quality of UAV networks, using connection features can better evaluate the channel state between nodes and the forwarding behaviors of nodes, which will make the detection model well resist in the influence of packet loss and retransmissions on the detection accuracy of time delay attack.

For instance, the combination of the transmission distance between communication parties and the transmission power of the sending node can better reflect the current channel state and the propagation delay, which is closely related to the forwarding delay of the packets. Moreover, the collected signal-to-interference-plus-noise (SINR) and other link quality parameters play an important role in some scenarios as they consider signal strength as well as interference and noise.

Therefore, as shown in Table II, the final selected connection features are $RxDist$, $SndDist$ and LQ, and it can be represented as

$$CFS = (RxDist, SndDist, LQ) \quad (13)$$

Meanwhile, in order to eliminate the dimensional influence between features and further improve the performance of the detection model, we perform the feature standardization (i.e., Z-score normalization) [50]:

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma} \quad (14)$$

where $\mathbf{x}$ is the original feature vector, $\bar{\mathbf{x}}$ represents the mean of the feature vectors, and σ is the standard deviation. Feature standardization is beneficial to avoid the problems of outliers, and can increase the difference between samples and the discrimination between features.

TABLE II
DESIGNED FEATURES

Feature	Description
t^i_{rfd}	The real forwarding delay of $node_i$ to m
$RxBufOcc$	The buffer occupancy of $node_i$ when it receives m from $node_{i-1}$
$SndBufOcc$	The buffer occupancy of $node_i$ when it sends m to $node_{i+1}$
$BufSize$	The buffer size of $node_i$
$RxSpd$	The speed of $node_i$ when it receives m from $node_{i-1}$
$RxDir$	The direction of $node_i$ when it receives m from $node_{i-1}$
$SndSpd$	The speed of $node_i$ when it sends m to $node_{i+1}$
$SndDir$	The direction of $node_i$ when it sends m to $node_{i+1}$
$TxPwr$	The transmission power of $node_i$
$MsgSize$	The data packet size of m
$RemTTL$	The remaining TTL of m when it is received by $node_i$
$MsgSrc$	The source node of m
$MsgDst$	The destination node of m
$MsgType$	The message type of m
$RxDist$	The distance between $node_{i-1}$ and $node_i$ when $node_i$ receives m from $node_{i-1}$
$SndDist$	The distance between $node_i$ and $node_{i+1}$ when $node_i$ sends m to $node_{i+1}$
LQ	The parameters of link quality between $node_i$ and $node_{i+1}$ when $node_i$ transmits m to $node_{i+1}$

D. Model Training

In UAV networks, although the benign training samples from normal nodes are cheap and relatively easy to obtain, the collection of labeled malicious samples for model training is not a trivial task. Meanwhile, the correctness and generality of the obtained malicious samples can not be guaranteed, that is, whether these samples correctly characterize time delay attack in UAV networks. In addition, if most of the training samples are benign, there will be sample bias, which will seriously affect the performance of the trained detection model.

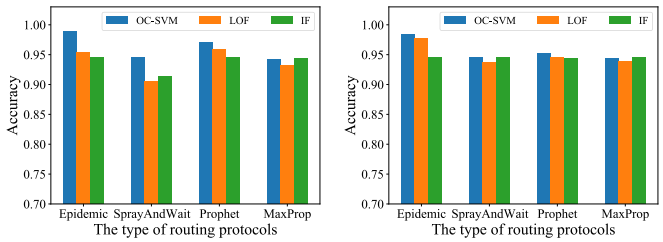


Fig. 5. The impact of different one-class classifiers on detection accuracy. (a) Scenario 1. (b) Scenario 2.

To overcome the aforementioned issues, we explore how to train the detection model and identify malicious nodes efficiently in the absence of labeled malicious samples. In this paper, one-class classification [51] is introduced and utilized for model training to further improve the versatility and practicability of ETD, where the training set only contains normal samples of single class. One-class classification has received considerable attention for anomaly detection in recent years [52], [53], [54], [55]. Different from binary or multiple classification, where the classifiers require samples of two or more classes to find differences between them, one-class classifiers only require samples of the normal class in order to learn representations and features of them, and then can identify if a new instance belongs to that class or not.

For each data packet m, we traverse its transmission path and extract all the two-hop sub-paths, in order to analyze and identify each forwarding behavior of each relay node on the transmission path accurately and efficiently. For each two-hop sub-path, we select the delay-related features of the forwarding node $node_i$ to obtain a training sample z, which can be expressed as $z = (\mathbf{x}, y)$, where $\mathbf{x}$ represents the feature vector of the forwarding node $node_i$, namely $\mathbf{x} = (NFS, MFS, CFS, t^i_{rfd})$, and y is the classification label of $\mathbf{x}$: "0" ("1") represents the forwarding behavior of $node_i$ is benign (malicious). After a period of data sampling, we get the training sample set with the same y (i.e., $y = 0$). Then based on the training sample set, one-class classification is used to train our detection model. There are three common one-class classification algorithms:

- One-class support vector machine (OC-SVM) [56]
- Local outlier factor (LOF) [57]
- Isolation forest (IF) [58]

Fig. 5 shows the detection accuracy of three one-class classifiers on time delay attack for UAV networks in two scenarios and four routing protocols. We choose OC-SVM as a representative for experiments and performance demonstrations due to the superior performance and space limitations.

OC-SVM [56], [59] is an extended classification algorithm of SVM, however, different from the traditional supervised learning-based SVM, it is an unsupervised learning algorithm. It assumes that all training samples belong to the same class, namely benign samples in this paper, and uses the kernel function to map the data points of the training set to the high-dimensional feature space. Then, it tries to find a minimal hypersphere that contains the data points of the training samples as many as possible. Meanwhile, it uses slack variables to control the influence of abnormal data points (i.e., correctly characterizes the benign forwarding behaviors). The process can be formalized as the following optimization problem [56]:

$$\begin{aligned} &\min_{R,a} \ R^2 + C\sum_{i=1}^{n}\zeta_i \\ &s.t. \ \ \|\mathbf{x}_i - \mathbf{a}\|^2 \le R^2 + \zeta_i \\ &\qquad \zeta_i \ge 0, \quad \forall i = 1, 2, 3 \ldots, n. \end{aligned} \tag{15}$$

where a is the center of the hypersphere, $R > 0$ is the radius of the hypersphere, C is the penalty parameter which controls the trade-off between the volume and the errors, ζ_i is the slack variables, which are utilized to create a soft margin with the penalty parameter C.

When identifying and classifying a new instance, if the data point of the new instance falls within the hypersphere in the high-dimensional feature space, the new instance belongs to this class (i.e., the forwarding behavior of $node_i$ is benign), otherwise it does not belong to this class (i.e., it is malicious). Gaussian Kernel is used as a distance function over two data points, and only if the new instance z satisfies the following inequality, it belongs to this class:

$$\|z - \mathbf{x}\|^2 = \sum_{i=1}^{n}\alpha_i \exp\left(\frac{-\|z - x_i\|^2}{s^2}\right) \ge -R^2/2 + C_R \tag{16}$$

where C_R depends only on the Support Vectors $\mathbf{x}_i$ and not on z.

E. Malicious Node Detection

After training the detection model, it can be used to evaluate the forwarding behaviors of nodes and identify time delay attack. For each node $node_i$ in the network, we evaluate each forwarding behavior of the node, based on which its trust value can be calculated. The trust value of $node_i$ can be formalized as

$$T_i = \frac{bf_i}{bf_i + mf_i} \tag{17}$$

where bf_i is the number of benign forwarding behaviors of $node_i$, mf_i is the number of malicious forwarding behaviors. The trust value of each node will be initiated as 0.5, with $bf_i = 1$ and $mf_i = 1$, indicating complete ignorance in the initial phase.

The basic process of malicious node detection is as follows. First, for each communication round, we analyze each data packet received during this communication round in the same way as the model training phase. However, at this time, the unlabeled samples we get include benign and malicious samples. Then, for each unlabeled sample (i.e., a two-hop sub-path), we think it as a classification problem, that is, classifying the given sample and judging whether the behavior of the current forwarding node (i.e., $node_i$) is benign or malicious. Therefore, the trained detection model is used for classification. If the sample is marked as malicious, the corresponding mf_i of $node_i$ is increased by 1, otherwise, it is

TABLE III
ENVIRONMENT SETTINGS

Item	Description
CPU	16-Core AMD Ryzen 7 4800U with Radeon Graphics @ 1.80 GHz
Memory	Kingston DDR4 8 GB*2
OS	Ubuntu 22.04.1 LTS
ONE	1.6.0
Scikit-learn	0.23.1

benign and bf_i is increased by 1. Finally, the trust value of each node in the network can be calculated.

Then, the K-Means clustering method is utilized to distinguish malicious nodes from benign ones according to their trust values. The output is the benign node set and the malicious node set.

V. PERFORMANCE EVALUATION

In this section, we provide a comprehensive experimental design and performance analysis of ETD on the Opportunistic Network Environment (ONE) simulator [16]. In order to evaluate the efficiency and accuracy of ETD, we extend the ONE simulator to support the time delay attack. All experiments are simulated in the Lenovo XiaoXin - 15ARE 2020 (16-Core AMD Ryzen 7 4800U with Radeon Graphics CPU @ 1.80 GHz, 16 GB RAM, 512 GB SSD). Table III shows the detailed environment settings.

A. Scenarios

Referring to the simulation scenarios in [47] and [48], we extend and design two typical simulation scenarios inspired by search and rescue missions. The simulation area, the number of UAVs and the deployment location of the two scenarios are all different.

In these two scenarios, one stationary ground station is placed together with different number of search UAVs and ferry UAVs. Each search UAV is assigned a $200m \times 200m$ region, in which the UAV should patrol and collect data. In order to efficiently cover the mission region, each search UAV adopts a typical search zigzag motion pattern, and each ferry UAV moves back and forth along specified trajectory to assist search UAVs to transmit data packets. Note that the flight trajectory of each UAV is planned in advance, and Fig. 6 shows the trajectories of UAVs in two scenarios. Each UAV in the network can be the source node of messages, which will generate data packets as needed. Meanwhile, there are some malicious UAVs in the network, which will carry out time delay attack with a certain probability. The ground station is a trusted authority where the detection model is deployed. Table IV summarizes the detailed default experimental parameters of the two scenarios.

B. Simulation Setup

In order to conduct extensive experiments, we implement time delay attack on four classic routing protocols based on the ONE simulator: Epidemic routing [2], Spray and Wait routing [3], Probabilistic routing [4] and MaxProp routing [5].

TABLE IV
DEFAULT SIMULATION SETTINGS

	Scenario 1	*Scenario 2*
Simulation area(m^2)	800×800	1200×1200
Number of UAVs	13	24
Mobility model	MapRouteMovement	
Communication range(m)	200	
UAV speed(m/s)	6	
Message size($Byte$)	1400	
Link throughput(KB/s)	14	
Link quality	1.0	
Interval of message creation in each UAV(s)	5	
Delay constraint of messages(s)	50	
Probability of attack	0.3	
Duration of time delay attack(s)	3	
Percentage of malicious nodes	0.3	
Communication Round(s)	480	

Meanwhile, as far as we know, there is no research on time delay attack detection in UAV networks, so we compare the proposed ETD with the state-of-the-art detection methods for time delay attack in CPSes [26] and PTP [33].

Based on their prior works [14], [15], [60], the authors in [26] propose a deep-learning based approach to characterize and detect time delay attack in CPSes. First, a Hierarchical Long Short-Term Memory (HLSTM) model is designed to process the continuous data sequences, and extract the relevant temporal features. Then, the classification module uses a deep-learning model to characterize and detect time delay attack. Since the method is independent of the location of the attack, we detect and identify each node individually. In addition, the approach is an online detection scheme, so we set the communication round to be the same as ETD to achieve the same detection delay and ensure fairness. All other parameters use the default values from the article [26].

Based on their prior work [13], the authors in [33] comprehensively summarize and analyze time delay attack in PTP, and then model the attack and quantify its impact. Based on the assumption of the symmetry of the communication path between the primary clock and the secondary clock in PTP, the method characterizes and detects time delay attack by observing and calculating the time offset between the primary and secondary clocks.

C. Metric

In order to comprehensively evaluate the detection performance, we use the accuracy (ACC), false positive ratio (FPR), false negative ratio (FNR), and F1-score as metrics, and based on Table V, they can be defined as $ACC = (TP + TN)/(TP + FP + FN + TN)$, $FPR = FP/(FP + TN)$, $FNR = FN/(FN + TP)$, and $F1 - score = (2 * TP)/(2 * TP + FP + FN)$. Meanwhile, to avoid bias, we run our simulation for each experiment of each routing protocol in each scenario with 100 rounds and calculate the average value as the final experimental result.

D. Parameter Experiment

As mentioned earlier, the detection model is used to detect malicious nodes, and we need to set the communication rounds reasonably to evaluate the behavior of nodes. Therefore, we conduct experiments to study the impact of different communication rounds on detection accuracy. The communication

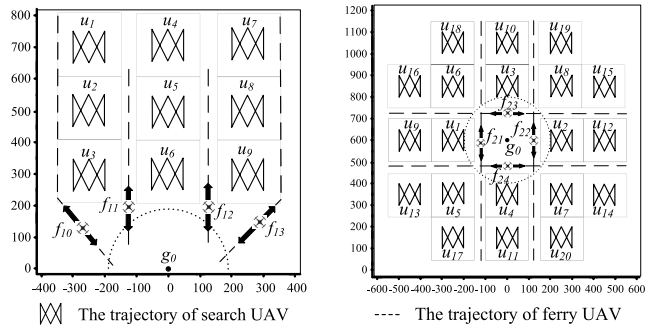


Fig. 6. The trajectories of UAVs in two scenarios. (a) Scenario 1. (b) Scenario 2.

TABLE V
EXPERIMENTAL EVALUATION

		Predicted Condition	
		Malicious	Benign
True Condition	Malicious	True Positive (TP)	False Negative (FN)
	Benign	False Positive (FP)	True Negative (TN)

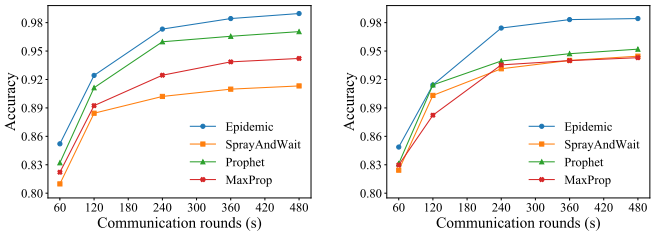


Fig. 7. The impact of the communication rounds on detection accuracy. (a) Scenario 1. (b) Scenario 2.

rounds are set to 60, 120, 240, 360, and 480s, and the experimental results are presented in Fig. 7.

The detection performance of ETD improves with the increase of communication rounds, because the samples used for evaluation also increase, which can well alleviate the impact of data distribution and single evaluation bias on the overall results of the trust values. As a result, the calculated trust values of nodes are more accurate. Meanwhile, we can find that even when the communication round is only 60s, the detection accuracy of ETD is still higher than 80% in all cases.

In addition, when the communication round increases from 60s to 240s, the detection accuracy of ETD increases rapidly. Subsequently, the detection accuracy gradually tends to be stable, and reaches the highest when the communication round is 480s. Owing to the space limitations of this study, we choose 240s as a representative for the following experiments. The trend of the experimental results of other parameters is roughly the same as that of the experimental results with a communication round of 240s.

E. Detection Performance Comparison

In this section, we compare the proposed ETD with the state-of-the-art detection schemes for time delay attack in CPSes [26] and PTP [33], and the experimental results are shown in Table VI.

Compared with the detection methods in CPSes and PTP, ETD always achieves the best accuracy and F1-score (higher than 90%) with the lowest FPR and FNR (lower than 10%) in four routing protocols and two scenarios. The reasons are as follows. First, ETD performs a comprehensive analysis and extraction of delay-related features from four different dimensions, thereby achieving an accurate characterization of time delay attack in dynamic and complex UAV networks. Second, one-class classification is utilized to effectively evaluate the forwarding behavior of nodes. Third, the trust value of each node can be calculated based on the forwarding behavior evaluation of each node. ETD then utilizes the K-means clustering method for malicious node classification to reduce the adverse impact of single evaluation bias on the overall evaluation results.

The detection method in CPSes [26] performs well in scenario 1; however, its performance drops significantly in scenario 2, which indicates the poor scalability of the method. It cannot adapt to large-scale and complex UAV networks in real-world environments. In addition, it achieves a high FPR, which causes numerous false alarms that are very troublesome in practical applications. The method utilizes the HLSTM model to capture the time-series dependencies of data in CPSes because the data in CPSes are continuous, and the transmission path is fixed. However, due to the dynamic and distributed nature of UAV networks, the data in UAV networks do not have time continuity and correlation. Meanwhile, the transmission paths of two consecutive packets forwarded by the same node may be different. Therefore, the proposed HLSTM model cannot accurately identify the behavior patterns of nodes in UAV networks.

The accuracy of the detection scheme in PTP [33] maintains approximately 50% in all cases, and its F1-score is lower than 50% in most cases. Additionally, its FPR and FNR are as high as 50%, which means that the scheme is infeasible and cannot accurately distinguish time delay attack in UAV networks. The reasons are as follows. First, the scheme relies on the assumption that the communication paths between the two parties are symmetric. However, the assumption does not hold in UAV networks because of the highly dynamic topology. Second, since PTP is only related to time features, the method only needs to model time delay attack from the perspective of the delay without considering other factors. However, due to the high dynamics and complexity of UAV networks, many factors can influence the forwarding delay, and it is difficult to model the relationship.

F. The Influence of Different Features

To study the features, we run the simulation to investigate the impact of different features on detection accuracy in two scenarios, four routing protocols and three network overheads. The experimental setup is shown in Table IV. Owing to space limitations, in this section, we present the detection accuracy of five key feature combinations in different situations:

1) Combination 1: All features in four different dimensions, as shown in Table II.

TABLE VI
RESULTS OF DIFFERENT DETECTION SCHEMES

		Scenario 1 (Router)				*Scenario 2 (Router)*			
		Epidemic	*SprayAndWait*	*Prophet*	*MaxProp*	*Epidemic*	*SprayAndWait*	*Prophet*	*MaxProp*
ETD (proposed)	*ACC*	0.9732	0.9021	0.9598	0.9245	0.9743	0.9313	0.9395	0.9354
	F1-score	0.9665	0.8972	0.9352	0.9147	0.9730	0.9142	0.9165	0.9334
	FPR	0.0253	0.0931	0.0380	0.0754	0.0257	0.0647	0.0526	0.0577
	FNR	0.0265	0.1155	0.0415	0.0775	0.0275	0.0656	0.0703	0.0689
Ganesh et al. [26]	*ACC*	0.9561	0.8912	0.8059	0.8346	0.6549	0.7417	0.5385	0.6022
	F1-score	0.9542	0.8887	0.7565	0.8338	0.6373	0.7386	0.5311	0.5721
	FPR	0.0830	0.0728	0.2096	0.3190	0.5792	0.4459	0.6641	0.6853
	FNR	0.0497	0.1319	0.3742	0.0243	0.1229	0.0193	0.1762	0.0751
Moussa et al. [33]	*ACC*	0.5352	0.5035	0.5773	0.5100	0.5047	0.5651	0.5758	0.4994
	F1-score	0.3710	0.4108	0.3050	0.4323	0.5082	0.3156	0.2985	0.5124
	FPR	0.3857	0.4509	0.3150	0.4718	0.5139	0.3381	0.3111	0.5381
	FNR	0.6124	0.5654	0.6908	0.5181	0.4755	0.6563	0.6825	0.4602

2) Combination 2: Delay, message and connection features, except node features.
3) Combination 3: Node, message and connection features, except delay feature.
4) Combination 4: Only message and connection features, except both delay and node features.
5) Combination 5: t^i_{rfd} in the delay feature; $RxBufOcc$, $SndBufOcc$, $BufSize$ in the node features; $MsgSize$, $MsgSrc$, $MsgDst$, $MsgType$ in the message features; and LQ in the connection features.

1) Complementarities and Synergies: In this part, we aim to explore the contribution of different features to the detection accuracy. The key results are shown in Table IV, namely the first four feature combinations.

First, in most situations, a certain detection accuracy can be achieved when only utilizing message and connection features, that is, using combination 4 to detect malicious nodes. Then, by comparing combination 2, 3 and 4, we can find that, on the basis of the message and connection features, the utilization of delay and node features can further improve the detection accuracy. Moreover, in different scenarios, routing protocols and network overheads, the delay feature and node features have their own pros and cons, and have different contributions to the detection accuracy: the delay feature is more conducive to malicious node detection when the network overhead is light, while the node features can better help the model detect time delay attack when the network overhead is heavy.

In addition, it is worth noting that taking all features in different dimensions into consideration, namely, utilizing combination 1, can always achieve the best detection accuracy.

To sum up, features in different dimensions have their own pros and cons, and all have contributions to the detection accuracy. Through the utilization of all features from four different dimensions, ETD achieves the complementarities and synergies between features, thereby being able to deal with time delay attack in different environments in UAV networks with high detection accuracy.

2) Tradeoff Between Overhead and Accuracy: In the last section, we investigate the influence of different feature combinations on the detection accuracy, and it is well-known that the best detection accuracy can be achieved by utilizing all features in four different dimensions. However, the approach needs to collect features through attaching delay-related information to the messages. Therefore, the extra cost it introduces to the network is also the largest. In some time-sensitive or energy-sensitive UAV application scenarios, this additional overhead is often not negligible. They would rather sacrifice some detection accuracy to reduce the extra overhead caused by the attached delay-related information. Therefore, in this section, we further study, analyze and select the delay-related features in order to achieve the tradeoff between the extra overhead and detection accuracy.

In addition to the experimental results shown above, in this part, we prefer to reduce the extra overhead of the network as much as possible while sacrificing a small amount of detection accuracy in order to explore a good tradeoff between the extra overhead and detection accuracy. To this end, we conduct extensive experiments, due to space limitations, we only show the key experimental results here.

As shown in Table VII, we compare the combination 1 and combination 5 in two scenarios, four routing protocols and three network overheads. It is worth noting that, compared with combination 1, combination 5 can also achieve good detection accuracy, while greatly reducing the introduced additional overhead: On the one hand, the delay-related information that each node attaches to the messages by combination 1 is 105 bits, while the extra overhead introduced by combination 5 is $7 + 13 + 13 + 8 + 8 + 8 = 57$ bits, which is only 54% of combination 1. On the other hand, in most situations, compared with combination 1, the decrease in the detection accuracy of combination 5 is within 6%. The experimental results reveal that ETD can effectively achieve a good tradeoff between the extra overhead and detection accuracy.

G. Overhead Analysis

In UAV networks, the storage and computing resources are relatively sufficient, however, there are tight communication resources [48]. Therefore, after exploring the tradeoff between the extra overhead and detection accuracy, in this part, we further conduct experiments on the extra overhead ratio introduced by the transmission of collected information.

TABLE VII
RESULTS SUMMARY

Router	Combination	Scenario 1			Scenario 2		
		Light	*Moderate*	*Heavy*	*Light*	*Moderate*	*Heavy*
Epidemic	*1*	0.9759	0.9732	0.9695	0.9758	0.9764	0.9787
	2	0.9724	0.9703	0.8712	0.9589	0.9128	0.9000
	3	0.9634	0.9582	0.9380	0.9561	0.9479	0.9478
	4	0.7459	0.8888	0.8215	0.8746	0.7697	0.7416
	5	0.9684	0.9728	0.9401	0.9560	0.9467	0.9388
SprayAndWait	*1*	0.9152	0.8946	0.8977	0.9287	0.9277	0.9272
	2	0.9100	0.8876	0.8666	0.9249	0.9208	0.8455
	3	0.8149	0.8251	0.8819	0.8534	0.8677	0.9222
	4	0.6204	0.7445	0.8048	0.7725	0.7875	0.7764
	5	0.8945	0.8718	0.8529	0.9123	0.9217	0.9028
Prophet	*1*	0.9524	0.9510	0.9224	0.9492	0.9481	0.9393
	2	0.9467	0.9432	0.8711	0.9477	0.9324	0.8908
	3	0.9007	0.9219	0.9096	0.9293	0.9054	0.9263
	4	0.7485	0.7978	0.8066	0.7708	0.6783	0.8360
	5	0.9490	0.9380	0.8815	0.9244	0.9343	0.9272
MaxProp	*1*	0.9405	0.8943	0.8911	0.9344	0.9339	0.9500
	2	0.9375	0.8606	0.8607	0.8964	0.8490	0.8992
	3	0.8981	0.8617	0.8735	0.8872	0.9057	0.9409
	4	0.7240	0.7808	0.7582	0.7338	0.7610	0.8851
	5	0.9327	0.8602	0.8548	0.9155	0.8779	0.9248

TABLE VIII
EXTRA OVERHEAD RATIO

	EpidemicRouter	*SprayAndWaitRouter*	*ProphetRouter*	*MaxPropRouter*
Scenario 1	2.28%	1.80%	1.86%	2.30%
Scenario 2	2.21%	1.78%	1.82%	2.17%

The extra overhead ratio can be defined as

$$EOR = \frac{\sum_{i=1}^{M} \sum_{j=1}^{H_i} j \times A_i}{\sum_{i=1}^{M} D_i \times H_i} \tag{18}$$

where M is the number of transmitted messages, H_i is the hop count to deliver m_i to the destination, D_i is the size of the original payload of m_i and A_i is the size of the information that each forwarding node attaches to m_i. In this paper, as mentioned in Section. IV-B, $A_i = 105$ bits, and as shown in Table IV, $D_i = 1400$ Bytes.

As shown in Table VIII, in two scenarios and four routing protocols, the extra overhead ratio introduced by ETD does not exceed 2.5%. The experimental results confirm the light-weight of the collection approach and indicate the versatility and practicability of ETD.

H. The Impact of Different Variables

In the following experiments, we mainly investigate the impact of some variables on the detection performance.

1) Impact of the Duration of Time Delay Attack: In this evaluation, we aim to explore the impact of the duration of time delay attack on the detection performance of ETD, including absolute time delay attack and relative time delay attack. The performance results are depicted in Fig. 8 and 9.

It is observed that in most situations, the detection accuracy of ETD for the absolute time delay attack is greater than 90%. As shown in Fig. 8, with the increasing duration of absolute time delay attack, the detection accuracy of ETD gains an obvious improvement. This is because in UAV networks, when the maliciously delayed time is too long, the attack will cause obvious abnormalities in the forwarding behaviors of the nodes and abnormal fluctuations in the forwarding delay, resulting in the exposure of malicious nodes themselves.

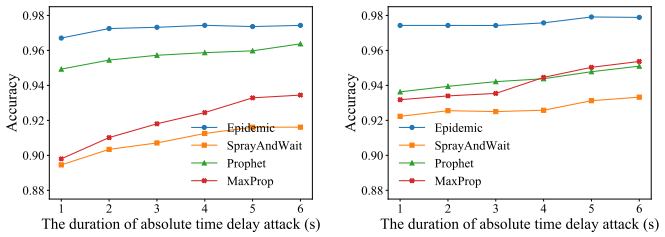


Fig. 8. The impact of the duration of absolute time delay attack on detection accuracy. (a) Scenario 1. (b) Scenario 2.

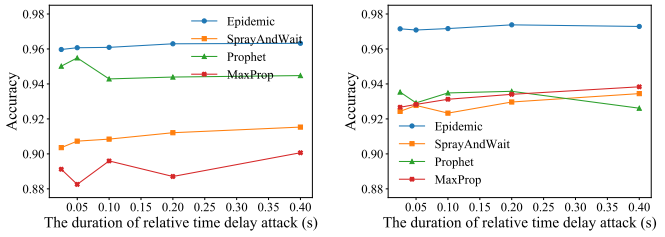


Fig. 9. The impact of the duration of relative time delay attack on detection accuracy. (a) Scenario 1. (b) Scenario 2.

Meanwhile, due to the flooding nature of Epidemic routing, compared with the other three routing protocols, in Epidemic routing, the time delay attack will cause the behaviors and performances of malicious nodes to deviate from normal nodes more seriously. Therefore, the one-class classifier can more accurately identify malicious behaviors based on the normal training samples.

Moreover, in order to further study the detection performance of ETD in different situations, we design the relative time delay attack, which can dynamically adjust the duration of the attack based on the transmission delay of the messages (e.g., one-quarter, one-half, one-time, two-times, four-times of the transmission delay). Experimental results show that on all four routing protocols of the two scenarios, ETD can achieve good detection accuracy, that is, higher than 88%, as shown in Fig. 9. This is because ETD conducts a comprehensive and in-depth analysis and feature extraction from four different dimensions, it can adapt to time delay attack in different environments.

2) Impact of the Probability of Time Delay Attack: As Fig. 10 shows, the detection accuracy of ETD maintains greater than 80% in all situations. Meanwhile, the detection accuracy of ETD decreases as the probability of time delay attack increases. This is because frequent time delay attack will greatly increase the overhead and complexity of the UAV network, which will adversely affect neighboring nodes and make it difficult to retrieve accurate information.

Furthermore, as the complexity of the scenarios increases, the detection accuracy of ETD does not fluctuate significantly, indicating the practicability and versatility of ETD. This is because the training set of ETD only contains normal samples. They can be used in different scenarios to detect attacks that show deviations with respect to legitimate behaviors of benign

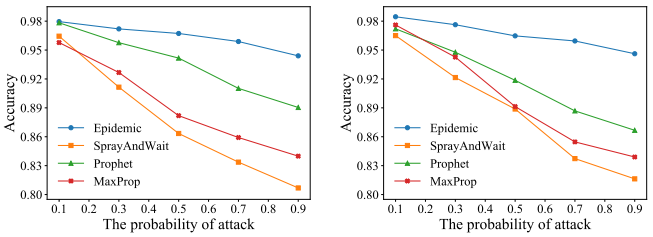


Fig. 10. The impact of the attack probability on detection accuracy. (a) Scenario 1. (b) Scenario 2.

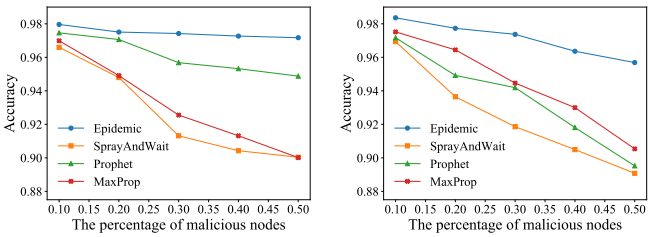


Fig. 11. The impact of the percentage of malicious nodes on detection accuracy. (a) Scenario 1. (b) Scenario 2.

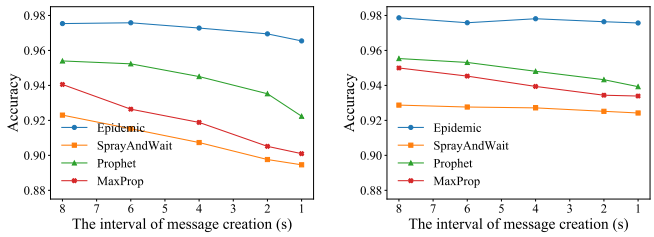


Fig. 12. The impact of the interval of message creation on detection accuracy. (a) Scenario 1. (b) Scenario 2.

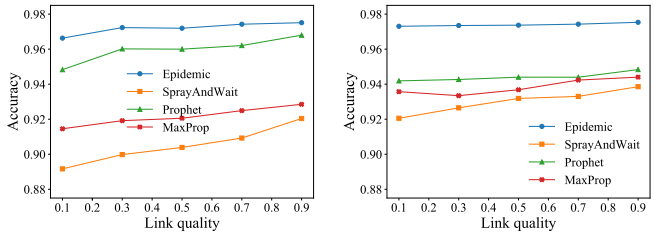


Fig. 13. The impact of the link quality on detection accuracy. (a) Scenario 1. (b) Scenario 2.

nodes in UAV networks and similarly good detection accuracy can be achieved.

3) Impact of the Percentage of Malicious Nodes: Fig. 11 shows the impact of the percentage of malicious nodes on the detection performance of ETD. It is found that the detection accuracy of ETD is greater than 90% in most situations. Meanwhile, as the percentage of malicious nodes in the UAV network increases, so does the decline in detection accuracy. In addition, the detection accuracy of ETD decreases as the complexity of the scenarios increase. The overall situation presented by the experimental results is similar to that of the probability of the attack.

4) Impact of the Interval of Message Creation: In this part, we explore the impact of the interval of message creation on the performance of ETD. We keep the total number of injected packets the same at different intervals of message creation. The experimental results are depicted in Fig. 12.

It is observed that the detection accuracy of ETD is greater than 90% in all situations, and the detection accuracy of ETD is decreasing as the interval of message creation decreases. This is because when the rate of message creation is fast, there will be numerous messages to be sent in the buffer of nodes in the UAV network, even if benign nodes, resulting in the increasing queuing delay of the messages. If so, ETD may misjudge it as a malicious node, resulting in a decrease in detection accuracy. In this case, although the node is benign, due to its heavy load, it does have an impact on the delivery of the message similar to time delay attack.

5) Impact of Link Quality: As depicted in Fig. 13, with the improvement of the link quality, the detection accuracy of ETD is generally on the rise. This is because when the link quality is poor, for the packet transmission between nodes, there will be numerous data packet loss and retransmissions, which will waste a lot of time and cause an increase in the load of the network. This poor network environment cause obvious abnormalities in the forwarding behaviors of the nodes and abnormal fluctuations in the delay of the messages, which increases the difficulty of time delay attack detection. However, it is noting that the detection accuracy of ETD is greater than 90% in most situations.

VI. Conclusion and Future Work

With the widespread application of UAV networks, UAV networks also face many security problems and attack threats. Time delay attack is a covert and threatening attack that is easy to implement and difficult to detect. Meanwhile, the unique characteristic and SCF mechanism of UAV networks greatly increase the concealment and destructiveness of time delay attack. However, to the best of our knowledge, there is no research on time delay attack in UAV networks.

In this paper, we model time delay attack in UAV networks and propose an Efficient Time Delay Attack Detection Framework (ETD). First, we perform a comprehensive selection of delay-related features from four different dimensions, namely delay, node, message and connection. Meanwhile, we utilize the pre-planned trajectory information of UAVs to accurately calculate the real forwarding delay of nodes. Then, one-class classification is used for model training to deal with the detection of time delay attack. With the trained model, each forwarding behavior of the node will be evaluated, based on which the trust value of each node can be calculated. Finally, the K-Means clustering method is further utilized to distinguish malicious nodes from benign ones according to their trust values. Extensive experimental results show that ETD performs well in terms of the detection accuracy and extra overhead.

Our detection method is a centralized scheme, and its detection performance depends on the collection and processing of

the data by the ground station, which introduces an inevitable communication overhead to UAV networks. Although we explore the tradeoff between the detection performance and extra overhead, it is still a challenge how to further reduce the communication and storage overhead without sacrificing the detection performance. As future work, we will explore to use federated learning and distributed learning to overcome this challenge.

REFERENCES

[1] D. Mishra and E. Natalizio, "A survey on cellular-connected UAVs: Design challenges, enabling 5G/B5G innovations, and experimental advancements," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107451.

[2] A. Vahdat et al., "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, 2000.

[3] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 252–259.

[4] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2012.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–11.

[6] X. Liu, M. Abdelhakim, P. Krishnamurthy, and D. Tipper, "Identifying malicious nodes in multihop IoT networks using diversity and unsupervised learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[7] S. Sun, Z. Ma, L. Liu, H. Gao, and J. Peng, "Detection of malicious nodes in drone ad-hoc network based on supervised learning and clustering algorithms," in *Proc. 16th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2020, pp. 145–152.

[8] P. Kaliyar, W. B. Jaballah, M. Conti, and C. Lal, "LiDL: Localization with early detection of Sybil and wormhole attacks in IoT networks," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101849.

[9] T. N. D. Pham and C. K. Yeo, "Detecting colluding blackhole and greyhole attacks in delay tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1116–1129, May 2016.

[10] Z. Ma, L. Liu, and W. Meng, "DCONST: Detection of multiple-mix-attack malicious nodes using consensus-based trust in IoT networks," in *Information Security and Privacy—25th Australasian Conference, ACISP 2020, Perth, WA, Australia, November 30–December 2, 2020, Proceedings*. Springer, 2020, pp. 247–267.

[11] S. Aneja, P. Nagrath, and G. N. Purohit, "Energy efficient reputation mechanism for defending different types of flooding attack," *Wireless Netw.*, vol. 25, no. 7, pp. 3933–3951, Oct. 2019.

[12] K. S. Xiahou, Y. Liu, and Q. H. Wu, "Robust load frequency control of power systems against random time-delay attacks," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 909–911, Jan. 2021.

[13] B. Moussa, M. Debbabi, and C. Assi, "A detection and mitigation model for PTP delay attack in an IEC 61850 substation," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 3954–3965, Sep. 2018.

[14] X. Lou et al., "Assessing and mitigating impact of time delay attack: Case studies for power grid controls," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 141–155, Jan. 2020.

[15] X. Lou et al., "Learning-based time delay attack characterization for cyber-physical systems," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2019, pp. 1–6.

[16] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. ICST Conf. Simulation Tools Techn.*, 2009, pp. 1–10.

[17] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar, and A. Rashid, "Data exfiltration: A review of external attack vectors and countermeasures," *J. Netw. Comput. Appl.*, vol. 101, pp. 18–54, Jan. 2018.

[18] K. Prathapchandran and T. Janani, "A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest—RFTRUST," *Comput. Netw.*, vol. 198, Oct. 2021, Art. no. 108413.

[19] W. Khalid et al., "A taxonomy on misbehaving nodes in delay tolerant networks," *Comput. Secur.*, vol. 77, pp. 442–471, Aug. 2018.

[20] Z. Ma, L. Liu, and W. Meng, "ELD: Adaptive detection of malicious nodes under mix-energy-depleting-attacks using edge learning in IoT networks," in *Information Security—23rd International Conference, ISC 2020, Bali, Indonesia, December 16–18, 2020, Proceedings*. Springer, 2020, pp. 255–273.

[21] L. Liu, Z. Ma, and W. Meng, "Detection of multiple-mix-attack malicious nodes using perceptron-based trust in IoT networks," *Future Gener. Comput. Syst.*, vol. 101, pp. 865–879, Dec. 2019.

[22] Z. Ma, L. Liu, and W. Meng, "Towards multiple-mix-attack detection via consensus-based trust management in IoT networks," *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101898.

[23] L. Yang, L. Liu, Z. Ma, and Y. Ding, "Detection of selective-edge packet attack based on edge reputation in IoT networks," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107842.

[24] L. Liu, X. Xu, Y. Liu, Z. Ma, and J. Peng, "A detection framework against CPMA attack based on trust evaluation and machine learning in IoT network," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15249–15258, Oct. 2021.

[25] J. Jamaludin and J. M. Rohani, "Cyber-physical system (CPS): State of the art," in *Proc. Int. Conf. Comput., Electron. Electr. Eng. (ICE Cube)*, Nov. 2018, pp. 1–5.

[26] P. Ganesh et al., "Learning-based simultaneous detection and characterization of time delay attack in cyber-physical systems," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3581–3593, Jul. 2021.

[27] M. R. Aliabadi, M. V. Asl, and R. Ghavamizadeh, "ARTINALI++: Multi-dimensional specification mining for complex cyber-physical system security," *J. Syst. Softw.*, vol. 180, Oct. 2021, Art. no. 111016.

[28] L. Zhou, H. Guo, and G. Deng, "A fog computing based approach to DDoS mitigation in IIoT systems," *Comput. Secur.*, vol. 85, pp. 51–62, Aug. 2019.

[29] Z. Lv, Y. Han, A. K. Singh, G. Manogaran, and H. Lv, "Trustworthiness in industrial IoT systems based on artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1496–1504, Feb. 2021.

[30] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7059–7067, Oct. 2022.

[31] K. Pan, D. Gusain, and P. Palensky, "Modelica-supported attack impact evaluation in cyber physical energy system," in *Proc. IEEE 19th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2019, pp. 228–233.

[32] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102481.

[33] B. Moussa, M. Kassouf, R. Hadjidj, M. Debbabi, and C. Assi, "An extension to the precision time protocol (PTP) to enable the detection of cyber attacks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 18–27, Jan. 2020.

[34] J. Neyer, L. Gassner, and C. Marinescu, "Redundant schemes or how to counter the delay attack on time synchronization protocols," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization for Meas., Control, Commun. (ISPCS)*, Sep. 2019, pp. 1–6.

[35] D. G. Berbecaru and A. Lioy, "Attack strategies and countermeasures in transport-based time synchronization solutions," in *Intelligent Distributed Computing XIV, 14th International Symposium on Intelligent Distributed Computing, IDC 2021, Virtual Event, 16–18 September 2021*. Springer, 2022, pp. 203–213.

[36] C. DeCusatis, R. M. Lynch, W. Kluge, J. Houston, P. A. Wojciak, and S. Guendert, "Impact of cyberattacks on precision time protocol," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2172–2181, May 2020.

[37] T. He, Y. Zheng, and Z. Ma, "Study of network time synchronisation security strategy based on polar coding," *Comput. Secur.*, vol. 104, May 2021, Art. no. 102214.

[38] E. Itkin and A. Wool, "A security analysis and revised security extension for the precision time protocol," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 1, pp. 22–34, Jan. 2020.

[39] L. Narula and T. E. Humphreys, "Requirements for secure clock synchronization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 749–762, Aug. 2018.

[40] M. Langer and R. Bermbach, "NTS4PTP—A comprehensive key management solution for PTP networks," *Comput. Netw.*, vol. 213, Aug. 2022, Art. no. 109075.

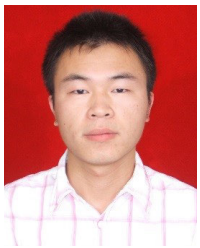
[41] K. Schneider, B. Zhang, and L. Benmohamed, "Hop-by-hop multipath routing: Choosing the right nexthop set," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Jul. 2020, pp. 2273–2282.

[42] Z. Zhou et al., "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.

[43] X. Li, L. Liu, L. Wang, J. Xi, J. Peng, and J. Meng, "Trajectory-aware spatio-temporal range query processing for unmanned aerial vehicle networks," *Comput. Commun.*, vol. 178, pp. 271–285, Oct. 2021.
[44] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing UAV communications via joint trajectory and power control," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1376–1389, Feb. 2019.
[45] J. Peng, H. Gao, L. Liu, N. Li, and X. Xu, "TBM: An efficient trajectory-based multicast routing protocol for sparse UAV networks," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 867–872.
[46] J. P. Jeong et al., "TPD: Travel prediction-based data forwarding for light-traffic vehicular networks," *Comput. Netw.*, vol. 93, pp. 166–182, Dec. 2015.
[47] J. Peng, H. Gao, L. Liu, Y. Wu, and X. Xu, "FNTAR: A future network topology-aware routing protocol in UAV networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.
[48] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 843–856, Mar. 2017.
[49] Y. Qin, M. A. Kishk, and M.-S. Alouini, "Performance evaluation of UAV-enabled cellular networks with battery-limited drones," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2664–2668, Dec. 2020.
[50] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105524.
[51] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.
[52] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks," in *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 2219–2236.
[53] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, "A one-class classification approach for bot detection on Twitter," *Comput. Secur.*, vol. 91, Apr. 2020, Art. no. 101715.
[54] S. R. Arashloo, "Matrix-regularized one-class multiple kernel learning for unseen face presentation attack detection," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4635–4647, 2021.
[55] A. George and S. Marcel, "Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 361–375, 2021.
[56] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in UAV-aided wireless systems: Unsupervised learning with one-class SVM and K-means clustering," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 139–142, Feb. 2020.
[57] W. A. Yousef, I. Traoré, and W. Briguglio, "UN-AVOIDS: Unsupervised and nonparametric approach for visualizing outliers and invariant detection scoring," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 5195–5210, 2021.
[58] S. Ahmed, Y. Lee, H. Seung-Ho, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2765–2777, Mar. 2019.
[59] J. Yi and S. Yoon, "Patch SVDD: Patch-level SVDD for anomaly detection and segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–16.
[60] X. Lou, C. Tran, R. Tan, D. K. Y. Yau, and Z. T. Kalbarczyk, "Assessing and mitigating impact of time delay attack: A case study for power grid frequency control," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Apr. 2019, pp. 207–216.

Wenbin Zhai received the B.S. degree from the Nanjing University of Chinese Medicine, Nanjing, Jiangsu, China, in 2020. He is currently pursuing the M.S. degree with the Nanjing University of Aeronautics and Astronautics, Nanjing. His research interests include wireless sensor networks.

Liang Liu received the B.S. degree in computer science from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2005, and the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China, in 2012. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include distributed systems, big data, and system security.



Youwei Ding received the B.S. and M.S. degrees in computer science from Yangzhou University, Yangzhou, Jiangsu, China, in 2007 and 2010, respectively, and the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, in 2016. He is currently an Associate Professor with the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing. His research interests include energy efficient data management, big data analysis, and data security.

Shanshan Sun received the B.S. degree from the China University of Geosciences, Beijing, China, in 2016. She is currently pursuing the M.S. degree with the Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China. Her research interests include drone network security.

Ying Gu received the B.E. degree in software engineering from Fudan University in 2019. She is currently pursuing a degree with the School of Engineering and Applied Sciences, Columbia University, New York, NY, USA. Her research interests include distributed computing.