Journal of Parallel and Distributed Computing 177 (2023) 117–130

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

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

HOTD: A holistic cross-layer time-delay attack detection framework for unmanned aerial vehicle networks

Wenbin Zhai [a], Shanshan Sun [a], Liang Liu [a,*], Youwei Ding [b], Wanying Lu [a]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[b] School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing, China

ARTICLE INFO

Article history:
Received 27 April 2022
Received in revised form 2 November 2022
Accepted 4 March 2023
Available online 11 March 2023

Keywords:
Time-delay attack
Unmanned aerial vehicle networks
Cross-layer
Supervised learning
Clustering

ABSTRACT

Recently, unmanned aerial vehicle (UAV) networks have been widely used in military and civilian scenarios; however, they suffer various attacks. Time-delay attacks maliciously delay the transmission of packets without tampering with the contents or significantly affecting the transmission pattern, making detection difficult. In this paper, a holistic cross-layer time-delay attack detection framework (HOTD) is proposed for UAV networks. A holistic selection of the delay-related features available at all layers is performed, before adopting supervised learning to build a consistency model between these features and the corresponding forwarding delay to calculate the degree of consistency of each node. Finally, the clustering method is used to distinguish malicious from benign nodes according to their degree of consistency. Experimental results show that the performance of HOTD is superior to that of state-of-the-art detection methods, and it achieves a detection accuracy higher than 85% with less than 2.5% additional overhead.

1. Introduction

With the development of sensors, navigation systems, and wireless communication technologies, unmanned aerial vehicle (UAV) networks have achieved significant performance improvements and attracted considerable attention from researchers. In UAV networks, numerous UAVs cooperate in clusters to deliver messages. UAV networks have many advantages such as flexible deployment, low manufacturing cost, and good scalability. Owing to their excellent performance, UAV networks have been widely used in military and civilian applications; for example, border patrols, disaster response, and farmland monitoring [28].

UAV networks are flexible; however, owing to their distributed nature, they are vulnerable to many security threats, including external and internal attacks [21]. Internal attacks launched by malicious nodes inside the network are more harmful than external attacks carried out by unauthorised UAVs. For instance, attackers can invade legitimate UAVs and carry out various types of attacks (e.g. packet drop, flood, replay, or tamper attacks) for specific malicious purposes [42]. Unfortunately, internal attacks cannot be resisted by traditional encryption and authentication schemes alone [15].

Time-delay attacks (TDAs) are a type of internal attack in which malicious nodes deliberately delay the transmission of received data packets before forwarding them to the destination. TDAs are more challenging to deal with and pose a greater threat to the network than other types of attacks. TDAs are characterised by being easy to implement and difficult to detect. Unlike traditional data-oriented attacks, which need to break cryptographic protection and tamper with data packets, TDAs only delay packet transmission, without manipulating the contents [23]. Additionally, unlike packet drop, flood, and replay attacks, carefully launched TDAs may not obviously affect the pattern of packet transmission [12].

Simultaneously, TDAs are ubiquitous and can cause significant damage. Many time-sensitive applications for UAV networks, such as forest surveillance [49], traffic monitoring [10], video conferencing [11], disaster rescue [27], task coordination [55], and battlefield networks [52], involve stringent requirements for the transmission delay of data. Data must be delivered to the destination on time; otherwise, their value will be significantly reduced or invalidated. For example, in forest fire monitoring, if fire alarm information is maliciously delayed, the fire may spread rapidly, resulting in huge losses [13]. Additionally, the real-time collaboration of a UAV network depends on the exchange of formation control and route maintenance information between UAVs [8]. If this information is maliciously delayed, it may lead to confusion and failure in formation control (e.g. UAV collisions), outdated and invalid route paths, and possible loss of control over the UAV swarm [56].

* Corresponding author.
E-mail address: liangliu@nuaa.edu.cn (L. Liu).

https://doi.org/10.1016/j.jpdc.2023.03.001
0743-7315/

Owing to the threat of TDAs, effective detection mechanisms must be developed. However, most existing works concentrate on packet drop, flood, replay, and tamper attacks [2,24,35], and there has been little research on TDAs. Unfortunately, research on TDAs focuses on wired and static wireless sensor networks (WSNs) [22, 29,50], rather than UAV networks.

Compared with conventional WSNs and mobile ad hoc networks, UAV networks have many unique characteristics, such as high mobility, sparse distribution, intermittent connectivity, and unstable link quality. These characteristics may result in a lack of instant and stable end-to-end paths. Therefore, many UAV networks deliver packets based on the store-carry-forward (SCF) mechanism [3,18,33]: when there is no suitable next-hop node within the communication range, the message-holder UAV stores and carries the message until it encounters a suitable forwarding UAV. These characteristics make existing TDA detection approaches unsuitable for UAV networks.

To the best of our knowledge, there has been no research on TDA detection in UAV networks. The challenges in achieving detection are manifold: (1) Owing to the high topology dynamics and intermittent communication connectivity, the transmission path and delivery delay of packets change rapidly. Therefore, malicious TDAs cannot be detected by significant fluctuations in the delivery delay. (2) Owing to the SCF mechanism, a relatively short malicious delay injected by the attacker is likely to be misjudged as normal UAV SCF behaviour. (3) Owing to its complex architecture and high dynamics, many factors influence the forwarding delay, resulting in difficulty constructing mathematical or relational models.

To overcome these issues, we propose a holistic cross-layer TDA detection framework (HOTD). To detect TDAs efficiently and accurately, we evaluate the forwarding delay of nodes rather than the delivery delay of messages. First, because the forwarding delay is related to each layer of the UAV network protocol (i.e. physical, medium access control (MAC), network, and application layers), we perform a holistic collection of the information available at these layers and then select the delay-related features from a cross-layer perspective. Subsequently, we utilise supervised learning to build a *consistency* model between the selected features and the corresponding forwarding delay to calculate the degree of consistency of each node in the network. Lastly, we use the clustering method to distinguish malicious from benign nodes according to their degree of consistency. In summary, we make the following key contributions:

- We construct a mathematical model of TDAs in UAV networks. We believe that this is the first attempt to detect these attacks in UAV networks.
- We propose HOTD by comprehensively and systematically collecting and selecting delay-related features at each layer in the UAV network protocol from a cross-layer perspective. The supervised learning algorithm is used to construct a consistency model between these features and the corresponding forwarding delay, and the clustering method is utilised to identify malicious nodes.
- We implement extensive simulations on the Opportunistic Network Environment (ONE) simulator [16]. The experimental results show that the performance of HOTD is superior to that of state-of-the-art detection methods [12,30]. Simultaneously, HOTD achieves over 85% detection accuracy with less than 2.5% extra overhead.

The remainder of this paper is organised as follows. In Section 2, we review and summarise existing works on malicious node detection. Section 3 formalises the model, including the network and attack models. The proposed HOTD is described in detail in Section 4. In Section 5, the performance of HOTD is evaluated through extensive simulations. Finally, we finish with Section 6.

2. Related works

2.1. Malicious node detection in UAV networks

In [35], researchers focused on colluding attacks, in which attackers cooperate to launch packet drop attacks and compensate for their misconduct. Recorded encounter information and the forwarding ratio of nodes were used to suspect and confirm malicious nodes in the network. In [2], three types of flood attacks were discussed and a trust-based approach was proposed to detect malicious nodes. The behaviour of malicious nodes was manifested and led to the loss of their reputation metrics.

Additionally, machining learning (ML) algorithms have been used for malicious node detection in WSNs [9]. A hybrid attack was considered in which attackers launched packet drop, tamper, and replay attacks simultaneously [25]. The information exchange between nodes was used to evaluate the node trustworthiness, before K-means clustering was used to distinguish benign and malicious nodes. An advanced attack was considered where malicious nodes only attacked data packets sent to specific neighbour nodes [53]. The reputation model of all nodes and edges was reduced to a multiple linear regression problem, and the support vector machine (SVM) algorithm was adopted to identify malicious edges and confirm malicious nodes. An intelligent attack was proposed, in which adversaries attacked only data packets that satisfied certain conditions [20]. Regression and clustering algorithms were used to evaluate the node trustworthiness and distinguish malicious from benign nodes. However, TDAs are easier to implement and more difficult to detect than these types of attacks.

Although the main security threats capable of invading and manipulating UAV networks [44] are active interfering attacks (13%) [6] and jamming attacks (12%) [36], both are consistent with TDAs in terms of the attack purpose and intent [5]; however, internal TDAs are more covert. Furthermore, existing detection methods are based on the channel and radio frequency characteristics of the physical and MAC layers [51]. These methods cannot handle internal TDAs in UAV networks because the attacks do not trigger obvious alarms in any layer of the protocol stack.

2.2. TDA detection

Recently, TDAs have attracted significant attention from researchers in various fields owing to their concealment and destructiveness, such as cyber-physical systems (CPSs) [57] and the precision time protocol (PTP) [30].

CPSs are classic time-sensitive systems that are vulnerable to TDAs and are typically in the form of wired networks and static WSNs. Researchers proposed a perturbation term to estimate the measurement deviation of the load and frequency, and used it to detect a TDA [50]. ML has been used to evaluate the impact of TDAs on system stability and security, and two-tiered mitigation measures have been developed to detect and defend against attacks [22]. In [23], recurrent neural networks (RNNs) were used to assess the effect of a TDA, and then detect and characterise the attack. A deep learning model was used to efficiently process the long-term sequence data. Based on this work, the authors of [12] improved the practicability of the system through real-time processing and online analysis of data from CPS sensors. Moreover, different detection model strategies were proposed which could be adjusted dynamically based on different objectives.

The PTP is a synchronisation protocol introduced in IEEE Std. 1588. It can achieve sub-microsecond accuracy, which makes it vulnerable to TDAs. Quantitative analysis of a TDA was conducted

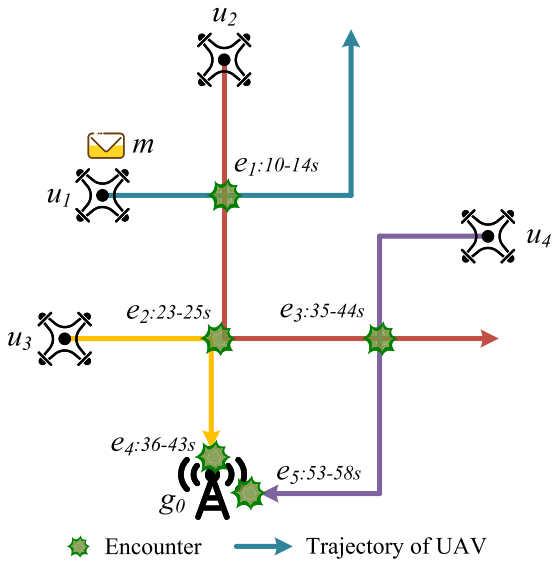


Fig. 1. Example of unmanned aerial vehicle network.

to show the vulnerability of the PTP to this attack [29]. A new type of PTP clock was utilised to respond to and mitigate TDAs. Subsequently, the authors of [30] further analysed and summarised the TDA surface in the PTP and proposed a protocol extension to enable the detection of TDAs against the PTP. Redundant paths and participants between the primary and secondary clocks were used to calculate the relative offset rate and time of the secondary clocks [31]. Clocks that drifted faster and further were suspected to be under attack.

However, the unique characteristics of TDAs in UAV networks cause existing detection approaches to be inefficient and inapplicable. Therefore, it is important to study TDA detection in UAV networks.

3. System model

In this section, we formulate a system model. We first describe a UAV network model, and then illustrate the TDA model in UAV networks, which is different from that in conventional wired networks and static WSNs.

3.1. Network model

Many UAVs patrol and search an area and send data packets to the ground station as needed. Without loss of generality, we abstract the three-dimensional space into a Euclidean space, ignoring the vertical space [39]. The trajectories of UAVs are pre-planned and can be obtained in advance through mission and path planning [59]. Even if UAVs re-plan their trajectories during a mission, these trajectories can be obtained by the ground station in advance [14]. Based on the pre-planned trajectories, the ground station calculates the encounters between UAVs [18]. For ease of representation, we abstract communication between UAVs as an encounter point [33]. As depicted in Fig. 1, there is a ground station g_0 and four UAVs u_1, u_2, u_3, and u_4, which fly along their trajectories. For example, UAV u_1 encounters u_2 at position e_1 between 10 and 14 s, which means u_1 and u_2 can communicate between 10 and 14 s.

3.1.1. Node model

We assume that there are malicious UAVs in the network which can carry out TDAs with a certain probability and the ground station is a trusted authority that collects data packets from the UAVs [25]. For convenience, in this paper both "UAV" and "node" represent a UAV in the network. A node can be represented as:

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

where id represents the unique identifier of the node (for example, u_1, u_2 in Fig. 1) and P_{TDA} is the probability of the node launching a TDA. For a benign node, $P_{TDA} = 0$, whereas $0 < P_{TDA} \leq 1$ for a malicious node.

3.1.2. Path model

The transmission path of 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 $node_1$ and $node_{n+1}$ denote the source and destination, respectively; t_i^s represents the time that $node_i$ starts sending m to $node_{i+1}$; t_{i+1}^r represents the time that $node_{i+1}$ successfully receives m from $node_i$; and $t_{i+1}^r - t_i^s = t_{trans}$, where t_{trans} is the time taken to successfully transmit m from $node_i$ to $node_{i+1}$.

For example, as illustrated in Fig. 1, at the start (0 s), u_1 generates m and wants to send it to g_0. For convenience, we assume that $t_{trans} = 1$ s. According to the pre-planned trajectory information, we deduce that a transmission path exists, such that $\langle (u_1, u_2, 10, 11), (u_2, u_3, 23, 24), (u_3, g_0, 36, 37) \rangle$, which means that u_1 encounters u_2 and transmits m to u_2 at position e_1 between 10 and 11 s. Then, u_2 stores and carries m until it encounters u_3 and transmits m to u_3 at position e_2 between 23 and 24 s, Finally, u_3 encounters g_0 and transmits m to g_0 at position e_4 between 36 and 37 s.

3.2. TDA model

Similar to prior studies [20,25,26,42,53], we assume that adversaries invade UAVs and use them to launch *time-delay attacks*, maliciously delaying data packet transmission by τ s. Let $t_i^{s'}$ denote the delayed time when malicious node $node_i$ starts transmitting m to $node_{i+1}$ and $t_{i+1}^{r}{}'$ denote the delayed time at which $node_{i+1}$ successfully receives m from $node_i$. In conventional wired networks and static WSNs, the TDA model can be formalised as:

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

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

However, the above model is not always true for UAV networks because of the SCF mechanism. For convenience, we assume that each message-holder UAV transmits m to the first UAV it encounters. Therefore, with no malicious nodes in the network, the transmission path is $\langle (u_1, u_2, 10, 11), (u_2, u_3, 23, 24), (u_3, g_0, 36, 37) \rangle$.

Then, we assume that u_2 is a malicious node and performs a TDA. When $\tau = 1$ s, the transmission path becomes $\langle (u_1, u_2, 10, 11), (u_2, u_3, 24, 25), (u_3, g_0, 36, 37) \rangle$, which is consistent with the attack model above. Subsequently, when $\tau = 3$ s, according to (3), $t_2^{s'}$ should be 26 s. However, as illustrated in Fig. 1, no UAV can communicate with u_2 at 26 s; u_2 must store and carry m until it encounters u_4 at position e_3 and transmit m to u_4 after another TDA. Therefore, the transmission path becomes $\langle (u_1, u_2, 10, 11), (u_2, u_4, 38, 39), (u_4, g_0, 53, 54) \rangle$. Here, $t_2^{s'} = 38$ s $\gg 26$ s. In this case, the TDA changes the original transmission path. Moreover, although the duration of the attack is 3 s, the delivery delay of m increases by 17 s $\gg 3$ s. Therefore, the unique characteristics and SCF mechanism of UAV networks make TDAs more destructive.

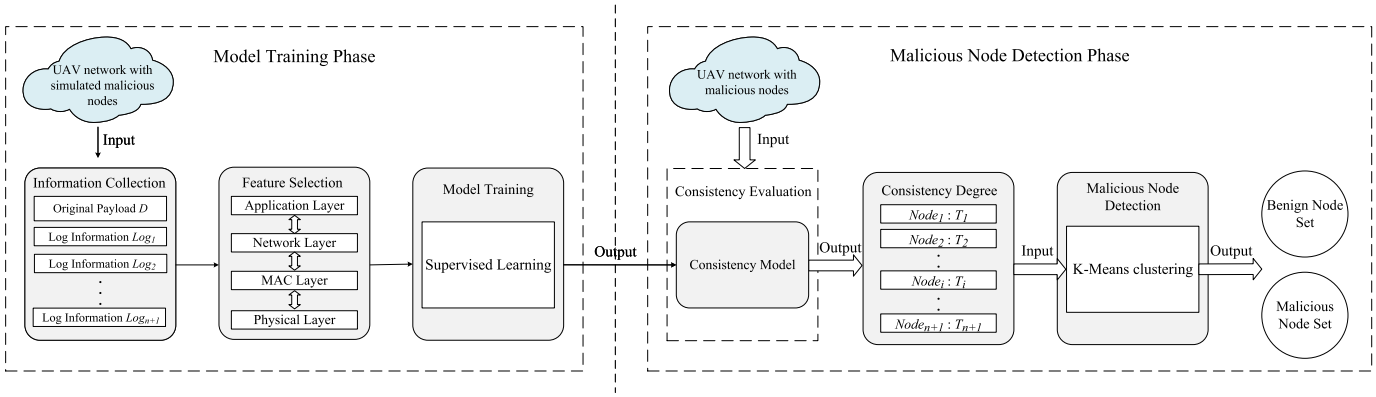


Fig. 2. Main workflow of our framework (HOTD).

When $\tau = 10$ s, there is no transmission path which can transmit m from u_1 to g_0. In this situation, a TDA is equivalent to a packet-drop attack. However, unlike packet drop attacks, which drop the received data packets randomly, TDAs prevent the timely delivery of packets. Moreover, TDAs are more covert than packet drop attacks, which can be easily detected [24,25].

To summarise, in contrast to conventional wired networks and static WSNs, TDA models in UAV networks can be formalised as:

$$t_i^{s\prime} = \begin{cases} t_i^s + \tau, & \text{if } \tau + t_{trans} \le t_{dur(i,i+1)}, \\ t_{ste(i,i+1)}{}' + \tau, & \text{otherwise,} \end{cases} \tag{5}$$

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

where $t_{dur(i,i+1)}$ denotes the duration of encounters between $node_i$ and $node_{i+1}$ and $t_{ste(i,i+1)}{}'$ represents the start time of the encounter between $node_i$ and $node_{i+1}{}'$, where $node_{i+1}{}'$ is the most suitable next-hop node of $node_i$ according to the routing protocol [3] and $\tau + t_{trans} \le t_{dur(i,i+1)}{}'$.

4. HOTD

4.1. Main workflow

Fig. 2 shows the main workflow of HOTD, consisting of information collection, feature selection, model training, and malicious node detection.

4.2. Information collection

To collect delay-related information efficiently in each layer of the network, the nodes attach their delay-related information to messages while forwarding them. The transmitted message can be formalised as:

$$M = < D, Log_1, Log_2, \ldots, Log_i, \ldots, Log_{n+1} >, \tag{7}$$

$$Log_i = <id, RT_i, RD_i, RS_i, RB_i, RL_i, TP_i, LQ_i, ST_i, SD_i, SS_i, SB_i >, \tag{8}$$

where D is the original payload of message M; Log_i denotes the transmission log information that $node_i$ attaches to M, including the delay-related information of $node_i$; RT_i (ST_i) is the receiving (sending) time of $node_i$; RD_i (SD_i) represents the distance between $node_i$ and $node_{i-1}$ ($node_{i+1}$) when $node_i$ receives (sends) M; RS_i (SS_i) is a vector representing the flight speed of $node_i$ when it receives (sends) M; RB_i (SB_i) is the buffer occupancy of $node_i$ when it receives (sends) M; RL_i is the remaining time to live (TTL) of M when received by $node_i$; and TP_i and LQ_i denote the transmission power and link quality, respectively, of $node_i$.

Although some delay-related information is attached to the message, it remains lightweight in terms of saving. We implement a message prototype to further analyse the additional overhead introduced by the message attachment. We use seven bits to encode id so that the network can support 2^7 UAVs. Then, we assume that the network performs missions for at most 2 h [37], so RT_i, RL_i, and ST_i (all units s), can be encoded by 13 bits. Next, we assume that the UAV has seven adjustable power levels, and thus TP_i is encoded by three bits. Finally, to precisely reflect the distance, flight speed, buffer occupancy, and link quality of the communication parties, RD_i, RS_i, RB_i, LQ_i, SD_i, SS_i, and SB_i are encoded by eight 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 B.

Meanwhile, the experimental results (see Section 5.7) confirm the lightweight nature of our collection approach. The extra overhead does not exceed 2.5% in any situation. Additionally, the costs of storage and transmission can be further reduced if efficient schemes are adopted. For example, except for the source node that records the complete time stamp, other forwarding nodes on the transmission path could record only the relative timestamp to reduce the extra overhead [32].

4.3. Feature selection

Owing to the complex unique characteristics of UAV networks, we must perform a holistic and systematic analysis of delay-related information and explore measures that may reveal the misbehaviour of attackers.

First, to accurately and efficiently evaluate and identify the behaviours of each node in the network, we traverse the transmission path of each received m and extract all two-hop sub-paths, which can be formalised as

$$Path \Rightarrow \bigcup_{i=2}^{n} \langle (node_{i-1}, node_i, t_{i-1}^s, t_i^r), (node_i, node_{i+1}, t_i^s, t_{i+1}^r) \rangle. \tag{9}$$

For example, at 0 s, u_1 generates m and wants to send it to g_0 (see Fig. 1). When there are no malicious nodes in the network, the transmission path is $\langle (u_1, u_2, 10, 11), (u_2, u_3, 23, 24), (u_3, g_0, 36, 37) \rangle$, as mentioned in Section 3.2. The path can be split into 2 two-hop sub-paths: $\langle (u_1, u_2, 10, 11), (u_2, u_3, 23, 24) \rangle$ and $\langle (u_2, u_3, 23, 24), (u_3, g_0, 36, 37) \rangle$. Then, for each two-hop sub-path of m, which can be 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 select the delay-related features at each layer from a cross-layer perspective to evaluate the behaviour and performance of $node_i$.

4.3.1. Physical layer

The physical layer is primarily responsible for providing wireless communication channels for data transmission. However, owing to the high dynamics of topology and intermittent connectivity of communications, transmitting data packets in UAV networks depends on the dynamic connections between nodes. Moreover, unstable link quality significantly affects the forwarding delay, and the wireless channel parameters available in this layer reflect the link quality. Therefore, the utilisation of information at the physical layer is beneficial for assessing the current channel state and resisting the influence of packet loss and retransmissions on the delay.

We chose the signal-to-interference-plus-noise ratio as a representation because it considers signal strength as well as interference and noise. Moreover, the combination of the transmission distance between communication parties and other communication-related information can better reflect the channel state and propagation delay, which is closely related to packet forwarding delay.

Therefore, as listed in Table 1, the final selected features in the physical layer are $RxDist$, $SndDist$, and LQ, which can be represented as follows:

$$PFS = (RxDist, SndDist, LQ). \tag{10}$$

4.3.2. MAC layer

The MAC layer is primarily responsible for data error and congestion control. In UAV networks, information related to nodes can be obtained at this layer. The selection of delay-related information at the MAC layer can effectively resist the influence of congestion on the forwarding delay and accurately evaluate the behaviour of nodes.

The forwarding delay can be reflected to a certain extent by the information at the MAC layer. For example, the buffer occupancy of UAVs reflects their traffic load and the message queuing delay. The transmission power of the node reflects the link quality and communication range of the node, which are related to the propagation delay.

Additionally, the utilisation of delay-related information at the MAC layer can evaluate node behaviour and help detect TDAs. For example, attackers tend to delay rather than immediately forward packets when encountering other nodes, regardless of the occupancy of their buffers. This often makes the buffer occupancy of attackers higher than that of normal nodes, resulting in inconsistencies with the normal forwarding delay.

As shown in Table 1, the final selected features in the MAC layer are $RxBufOcc$, $SndBufOcc$, $BufSize$, and $TxPwr$, which can be represented as follows:

$$MFS = (RxBufOcc, SndBufOcc, BufSize, TxPwr). \tag{11}$$

4.3.3. Network layer

The target of the network layer is to provide stable data communication to the nodes. Information used to characterise the message and end-to-end transmission path can be obtained here. Extracting delay-related features at the network layer helps evaluate forwarding delays and identify malicious nodes. For example, the transmission delay (the duration between the first and last digits of the message leaving the sending node) depends on the message packet size. The utilisation of the message TTL further assists in evaluating the forwarding delay of the node.

Moreover, selecting the source node [20], destination node [53], and type of message as features, can better resist and identify various TDAs, such as intelligent attacks against these specific features. Combining these features with other delay-related features, more accurately identifies abnormal behaviours of malicious nodes.

Table 1
Design features.

Feature	Description
t^i_{sc}	Estimated duration that $node_i$ stores and carries m
$RxSpd$	Speed of $node_i$ when it receives m from $node_{i-1}$
$RxDir$	Direction of $node_i$ when it receives m from $node_{i-1}$
$SndSpd$	Speed of $node_i$ when it sends m to $node_{i+1}$
$SndDir$	Direction of $node_i$ when it sends m to $node_{i+1}$
$MsgSize$	Data packet size of m
$RemTTL$	Remaining time to live of m when $node_i$ receives m from $node_{i-1}$
$MsgSrc$	Source node of m
$MsgDst$	Destination node of m
$MsgType$	Message type of m
$RxBufOcc$	Buffer occupancy of $node_i$ when it receives m from $node_{i-1}$
$SndBufOcc$	Buffer occupancy of $node_i$ when it sends m to $node_{i+1}$
$BufSize$	Buffer size of $node_i$
$TxPwr$	Transmission power of $node_i$
$RxDist$	Distance between $node_{i-1}$ and $node_i$ when $node_i$ receives m from $node_{i-1}$
$SndDist$	Distance between $node_i$ and $node_{i+1}$ when $node_i$ sends m to $node_{i+1}$
LQ	Parameters of link quality between $node_i$ and $node_{i+1}$ when $node_i$ transmits m to $node_{i+1}$

Therefore, as shown in Table 1, HOTD selects the final features $MsgSize$, $RemTTL$, $MsgSrc$, $MsgDst$, and $MsgType$, which can be represented as:

$$NFS = (MsgSize, RemTTL, MsgSrc, MsgDst, MsgType). \tag{12}$$

4.3.4. Application layer

The application layer provides services to users and characterises the objective entities of applications. In UAV networks, this layer performs analysis and utilisation based on data management and processing.

For each two-hop sub-path of m, denoted as $\langle(node_{i-1}, node_i, t^s_{i-1}, t^r_i), (node_i, node_{i+1}, t^s_i, t^r_{i+1})\rangle$, forwarding delay t^i_{fd} of $node_i$ to m can be formalised as

$$t^i_{fd} = t^r_{i+1} - t^s_{i-1}. \tag{13}$$

However, owing to the unique SCF mechanism of UAV networks, t^i_{fd} includes the transmission delay and duration for which UAVs store and carry the packet. For example, as illustrated in Fig. 1, for the two-hop sub-path $\langle(u_1, u_2, 10, 11), (u_2, u_3, 23, 24)\rangle$, u_2 is a benign node, $t^2_{fd} = 24$ s -10 s $= 14$ s with a 2 s transmission delay, and u_2 stores and carries the packet for 12 s (from 11 to 23 s).

If we directly use forwarding delay t^i_{fd} as an input for model training, the duration for which UAVs store and carry the packets, t^i_{sc}, greatly affects the performance of the trained consistency model. Therefore, to eliminate the adverse impact of this duration, we utilise the pre-planned trajectory information obtained at the application layer for estimation, and then use it as a feature to construct a better consistency model with forwarding delay t^i_{fd}. t^i_{sc} is proposed to evaluate the duration that UAVs store and carry packets, which is formalised as

$$t^i_{sc} = t_{ste(i,i+1)} - t^s_{i-1}, \tag{14}$$

where $t_{ste(i,i+1)}$ represents the start time of an encounter between $node_i$ and $node_{i+1}$. For example, as Fig. 1 shows, the start time of the encounter between u_2 and u_3 is 23 s. Therefore, $t^2_{sc} = 23$ s -10 s $= 13$ s. We then assume that u_2 is a malicious node and launches a TDA. When $\tau = 1$ s, the corresponding two-hop sub-path is $\langle(u_1, u_2, 10, 11), (u_2, u_3, 24, 25)\rangle$, $t^2_{fd} = 25$ s -10 s $= 15$ s, and $t^2_{sc} = 23$ s -10 s $= 13$ s. Additionally, when $\tau = 3$ s, the transmission path becomes $\langle(u_1, u_2, 10, 11), (u_2, u_4, 38, 39), (u_4, g_0, 53,$

54)⟩. The corresponding two-hop sub-path becomes ⟨$(u_1, u_2, 10, 11), (u_2, u_4, 38, 39)$⟩, which is different from the original path. Here, $t^2_{fd} = 39\ \text{s} - 10\ \text{s} = 29\ \text{s} \gg t^2_{sc}$. If $node_i$ is an attacker and launches a TDA, there are inconsistencies between t^i_{fd} and t^i_{sc}, particularly when the TDA changes the original path.

Based on the transmission power, we can further utilise the flight speed and direction of the UAV to estimate the link quality and communication range over a period of time [3].

As shown in Table 1, the final selected features in the application layer are t^i_{sc}, $RxDir$, $RxSpd$, $SndDir$, and $SndSpd$, which can be represented as:

$$AFS = (t^i_{sc}, RxDir, RxSpd, SndDir, SndSpd). \tag{15}$$

Meanwhile, to eliminate the dimensional influence between features and further improve the performance of the consistency model, we perform feature standardisation (i.e. Z-score normalisation) [40]:

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma}, \tag{16}$$

where $\mathbf{x}$ is the original feature vector, $\bar{\mathbf{x}}$ is the mean feature vector, and σ is the standard deviation. Feature standardisation is beneficial for avoiding outliers and can increase the difference between samples and discrimination between features.

4.4. Model training

We utilised supervised learning to build a consistency model to detect TDAs in UAV networks. To obtain sufficient labelled benign and malicious samples to train the model before the model training phase, we injected data packets into the network. Meanwhile, some benign nodes were driven to simulate the attack behaviours of malicious nodes, which launched a TDA with a set probability. Based on the analysis of these injected data packets, we collected benign and malicious samples for consistency model training and construction.

We traversed the transmission path of each data packet received at the ground station and extracted all two-hop sub-paths to analyse and identify forwarding behaviour. Then, for each two-hop sub-path, we selected the delay-related features of forwarding node $node_i$ to obtain a training sample z, which can be expressed as $z = (\mathbf{x}, y)$, where $\mathbf{x} = (PFS, MFS, NFS, AFS, t_{fd})$ and y denotes the classification label of $\mathbf{x}$. The forwarding behaviour of $node_i$ was benign (0) or malicious (1). After a period of data sampling, we obtained the labelled training dataset, including benign and malicious samples, which were used with supervised learning to train our consistency model.

4.5. Malicious node detection

After obtaining the trained consistency model, it was used to identify malicious nodes and detect TDAs. For each node in the network, the forwarding behaviour was evaluated to calculate and obtain their degree of consistency. The degree of consistency of $node_i$ was formalised as

$$C_i = \frac{bf_i}{bf_i + mf_i}, \tag{17}$$

where bf_i and mf_i indicate the number of benign and malicious forwarding behaviours of $node_i$, respectively. C_i must be a real number between 0 and 1; this was initially set as 0.5, with $bf_i = mf_i = 1$, indicating complete ignorance in the initial phase.

The process of malicious node detection was as follows: First, we analysed each received data packet in the same manner as in the model training phase. However, all the samples obtained were unlabelled. For each unlabelled sample, we determined whether the delay-related features of the current forwarding node were consistent with the corresponding forwarding delay. If the sample was marked as consistent, the current forwarding behaviour of $node_i$ was benign, and bf_i was increased by one; otherwise, it was inconsistent and mf_i was increased by one. Finally, the degree of consistency for each node was obtained and used by the clustering method to distinguish malicious from benign nodes. The outputs were the benign and malicious node sets.

Table 2
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	

Note: UAV, unmanned aerial vehicle.

5. Performance evaluation

In this section, we evaluate and analyse the performance of HOTD in an ONE simulator [16].

5.1. Scenarios

We designed two simulation scenarios for UAV networks inspired by forest surveillance missions and battlefield networks [3,34]. The simulation area, number of UAVs, and deployment location of the two scenarios were different. Each UAV was responsible for a 200 × 200 m area and used a typical zigzag movement pattern to efficiently cover the region. The UAV flight trajectories were planned in advance. UAVs generated data packets as required and sent them back to the ground station. The ground station was a trusted authority, and there were malicious nodes which launched TDAs. Table 2 summarises the default simulation settings.

5.2. Simulation setup

To evaluate the performance of HOTD, we conducted extensive experiments on four classical routing protocols for UAV networks: epidemic [46], spray-and-wait [41], probabilistic [19], and MaxProp routing [4]. Due to the lack of research on TDA detection in UAV networks, we compared HOTD with current state-of-the-art schemes for TDA detection in CPSs [12] and the PTP [30] (i.e. static networks) to demonstrate the uniqueness of TDAs in UAV networks and the efficiency of HOTD.

We used the advanced deep learning-based method to characterise and detect TDAs in CPSs [12], as discussed in Section 2.2. The hierarchical long short-term memory (LSTM) model is a data-driven approach for processing a continuous stream of data and characterising an attack. Subsequently, a deep learning model was utilised as the classification module to detect attacks. Because the method was independent of the position of the attack, we individually identified each UAV node. Meanwhile, because the method was an online detection method, we set the reaction latency as the end time of the experiment to ensure fairness; that is, the method

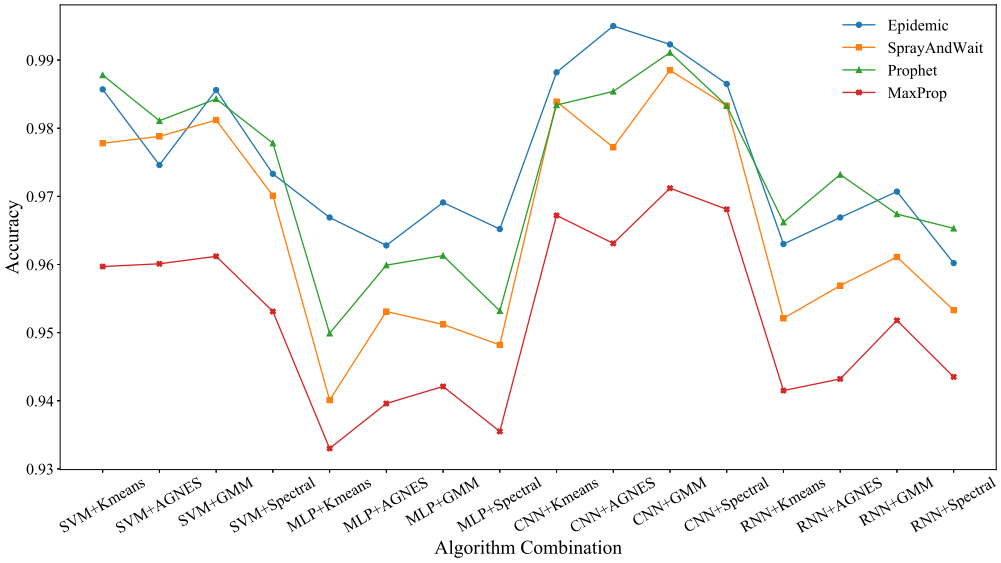


Fig. 3. Impact of different algorithm combinations on detection accuracy in Scenario 1.

only needed to identify malicious nodes by the end of the experiment. All other parameters used were default values from [12].

Based on their prior work [29], the authors in [30] performed a comprehensive analysis and summary of the TDA surface in the PTP, and then modelled and quantified the effect of TDAs. Based on the assumption of symmetry in the communication path between the primary and secondary clocks in the PTP, the method accurately modelled the delay characteristics and detects TDAs by observing and calculating the offset between the primary and secondary clocks. We used this method to characterise and detect TDAs in the PTP.

5.3. Metrics

To measure and compare the detection performance, an error matrix was used, in which true positive (TP) and true negative (TN) represent correct detections of a TDA, and false negative (FN) and false positive (FP) are incorrect detections. The sensitivity and specificity of the method correspond to avoiding the false-positive rate (FPR) and false-negative rate (FNR), respectively. We define the accuracy rate as $ACC = (TP + TN)/(TP + FP + FN + TN)$, FPR as $FPR = FP/(FP + TN)$, and FNR as $FNR = FN/(FN + TP)$. To avoid bias, each experiment was simulated for 100 rounds and the average value was used as the final result.

5.4. Algorithm combination comparison

The performance of HOTD relies on a combination of the supervised learning and clustering algorithms. Therefore, we conducted experiments to study the detection accuracy of different algorithm combinations under HOTD on TDAs in UAV networks. We chose four typical supervised learning algorithms: SVM [48], multilayer perceptron (MLP) [1], convolutional neural network (CNN) [7], and RNN [45]; and four classical clustering algorithms: K-means [38], agglomerative nesting hierarchical (AGNES) [43], Gaussian mixed model (GMM) [58], and spectral clustering [54].

The MLP model included one input, one fully connected layer (with 10 hidden neurones), and one output layer. The CNN model consisted of two 1D-CNNs (filters = 64 and 128, kernel size = 4), one flattened layer, one fully connected layer (with 256 hidden neurones, activation function = rectified linear unit), one dropout layer (dropout rate = 0.5), and one fully connected layer (activation function = SoftMax). The RNN model was composed of one gated recurrent unit and one fully connected layer.

The experimental results are presented in Figs. 3 and 4. We found that the performance of the CNN + GMM algorithm was better than the other algorithm combinations. Owing to the shared convolution kernel, CNN can handle high-dimensional data, and has been proven to effectively detect malicious nodes in WSNs [17,47]. The convolutional layer of CNN can automatically perform feature extraction, which makes CNN suitable for UAV networks with complex architectures and many delay-related features. Meanwhile, weight sharing between the convolutional layers decreases the number of training parameters, which greatly reduces the complexity of the network structure and makes CNN more applicable. Additionally, the fully connected layer of CNN alleviates overfitting, while decreasing the loss of feature information. Simultaneously, GMM clustering considers both the mean and variance of the data and uses the expectation-maximisation (EM) algorithm to iteratively update the model parameters, thereby achieving higher accuracy. Furthermore, GMM adopts a probabilistic model (soft classification), which provides more flexibility than other clustering methods.

All the algorithm combinations achieved a good detection performance (over 90%) in two scenarios and four routing protocols, which indicates the wide applicability of HOTD. Owing to the space limitations of this study, we chose algorithm CNN + GMM to represent HOTD for the following experiments.

5.5. Detection performance comparison

We evaluated our proposed HOTD against current state-of-the-art schemes for TDA detection in CPSs [12] and the PTP [30], and the results are listed in Table 3. HOTD always achieved good accuracy (over 95%) while maintaining low FPR and low FNR (below 10%). In TDA scenarios for UAV networks, HOTD's performance is far superior to that of detection methods for CPSs and the PTP. The reasons for this are as follows. First, because of the complex architecture of UAV networks, HOTD performs a holistic analysis of the information available at each layer of the UAV network protocol, and then extracts the delay-related features from a cross-layer perspective, enabling a comprehensive and accurate characterisation of TDAs. Second, a consistency model between these features and the corresponding forwarding delay is constructed to effectively evaluate the forwarding behaviour of each node in the network.

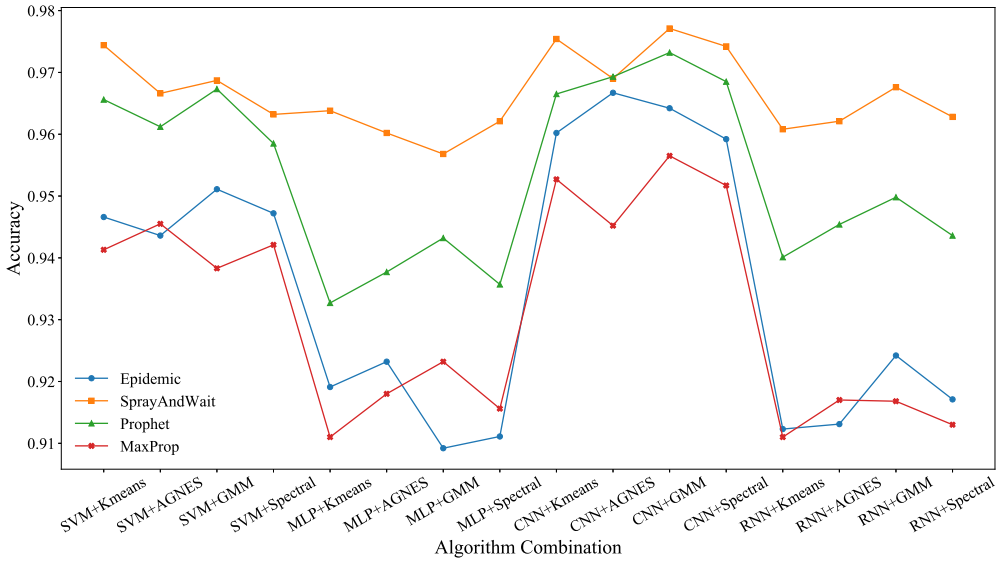


Fig. 4. Impact of different algorithm combinations on detection accuracy in Scenario 2.

Table 3
Results of different detection schemes.

		Scenario 1 (Router)				*Scenario 2 (Router)*			
		Epidemic	*Spray and Wait*	*Prophet*	*MaxProp*	*Epidemic*	*Spray and Wait*	*Prophet*	*MaxProp*
HOTD (proposed)	*ACC*	0.9923	0.9885	0.9911	0.9712	0.9642	0.9771	0.9732	0.9565
	FPR	0.0094	0.0211	0.0132	0.0277	0.0357	0.0206	0.0223	0.0613
	FNR	0.0076	0.0393	0.0178	0.0293	0.0368	0.0329	0.0532	0.0542
Ganesh et al. [12]	*ACC*	0.9724	0.9224	0.8149	0.8506	0.6616	0.7523	0.5457	0.6071
	FPR	0.0322	0.0557	0.1538	0.2442	0.5782	0.3596	0.6513	0.6663
	FNR	0.0193	0.1081	0.2746	0.0186	0.0751	0.0156	0.1398	0.0663
Moussa et al. [30]	*ACC*	0.5352	0.5035	0.5773	0.5100	0.5047	0.5651	0.5758	0.4994
	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

Note: ACC, accuracy; FPR, false positive ratio; FNR, false negative ratio.

Third, the degree of consistency of each node can be calculated based on the forwarding behaviour evaluation of each node. HOTD then utilises a clustering algorithm to classify nodes, which mitigates the impact of evaluation bias on the overall results.

As shown in Table 3, the method for CPSs [12] performs well in scenario 1, but the performance drops sharply in scenario 2. Therefore, the CPS TDA detection method has poor scalability under highly dynamic and complex UAV networks. Moreover, it cannot adapt to large-scale and increasingly complex UAV networks in real-world environments. We found that the FNR of this method was low and FPR was very high, showing that although the method rarely misses malicious nodes, it simultaneously misjudges numerous benign nodes. This causes many false alarms, which are very troublesome in practical applications. In CPSs, the transmission paths of packets are fixed and the data of each node are sequential and continuous. Therefore, the method uses LSTM, which accurately captures the dependencies and features in the time series to detect TDAs. However, owing to the characteristics of UAV networks, the transmission paths of packets change dynamically. Meanwhile, the transmitted packets for each node are discontinuous, the time interval of each packet transmission is not fixed, and there is no close correlation between adjacent packets. Therefore, LSTM cannot learn an effective pattern for identifying malicious nodes in UAV networks.

The detection method for TDAs in the PTP is inapplicable in UAV networks, as shown in Table 3. Under the four routing protocols in the two scenarios, the detection accuracy of the method in [30] was approximately 50%. Meanwhile, the FPR and FNR were as high as 50% in most cases. Therefore, the method cannot discriminate TDAs in UAV networks because the PTP assumes that the communication path between the primary and secondary nodes is symmetrical. The existing methods all rely on this assumption, but it does not hold in UAV networks because of the highly dynamic topology. Additionally, because the structure of the PTP is relatively simple and only relates to time characteristics, the method extracts and processes information related to the delay without considering other factors. However, owing to the complex architecture of UAV networks, many factors can influence the forwarding delay, and it is difficult to model this relationship.

5.6. Influence of different features

We then investigated the influence of different features on detection accuracy. We conducted extensive experiments on different feature combinations in two scenarios, four routing protocols, and three network overheads. Owing to space limitations, we only show the accuracy results for the following five key combinations in Table 4.

1. Combination 1: All features at different layers, as shown in Table 1.
2. Combination 2: Features at the physical, network, and application layers.

Table 4
Accuracy results of different feature combinations.

		Scenario 1			Scenario 2		
Router	Combination	Light	Moderate	Heavy	Light	Moderate	Heavy
Epidemic	1	0.9976	0.9795	0.9113	0.9697	0.9579	0.9166
	2	0.9943	0.9770	0.8188	0.9528	0.8963	0.8435
	3	0.9849	0.9013	0.8817	0.9499	0.9304	0.8877
	4	0.7626	0.8127	0.7721	0.8691	0.7554	0.6945
	5	0.9912	0.9768	0.8832	0.9554	0.9222	0.8782
Spray and Wait	1	0.9955	0.9810	0.9431	0.9828	0.9664	0.9487
	2	0.9898	0.9732	0.9106	0.9775	0.9594	0.8655
	3	0.8860	0.9051	0.9266	0.9013	0.9040	0.9440
	4	0.6748	0.8161	0.8456	0.8157	0.8204	0.8072
	5	0.9787	0.9676	0.9097	0.9754	0.9565	0.9305
Prophet	1	0.9912	0.9911	0.9245	0.9740	0.9732	0.9570
	2	0.9854	0.9830	0.8730	0.9705	0.9573	0.9077
	3	0.9374	0.9608	0.9116	0.9501	0.9293	0.9438
	4	0.7789	0.8314	0.8085	0.7881	0.6963	0.8518
	5	0.9864	0.9737	0.8886	0.9487	0.9627	0.9466
MaxProp	1	0.9920	0.9364	0.9059	0.9678	0.9565	0.9605
	2	0.9889	0.9011	0.8751	0.9286	0.8677	0.9091
	3	0.9472	0.9023	0.8885	0.9193	0.9257	0.9517
	4	0.7636	0.8215	0.7707	0.7602	0.7777	0.8949
	5	0.9889	0.9193	0.8703	0.9452	0.9007	0.9394

3. Combination 3: Features at the physical, MAC, and network layers.
4. Combination 4: Features at the physical and network layers.
5. Combination 5: LQ at the physical layer; $RxBufOcc$, $SndBufOcc$, and $BufSize$ at the MAC layer; $MsgSize$, $MsgSrc$, $MsgDst$, and $MsgType$ at the network layer; and t_{sc}^{i} at the application layer.

5.6.1. Complementarities and synergies

In this section, we focus on the contribution of different features to detection accuracy, and combinations 1-4 are chosen as representative results.

The physical and network layers form the basis for UAV network communication; therefore, the utilisation of features in these layers (combination 4) achieves a certain detection accuracy. We found that features at the MAC and application layers are beneficial for detecting TDAs in different environments. The MAC layer is primarily responsible for data error and congestion control, and the application layer performs a further process and utilisation of data. Therefore, on the basis of the physical and network layers, further utilisation of features at the MAC layer (combination 3) can better handle large-scale or heavy-load UAV networks. We utilised the pre-planned trajectory information, which can be obtained at the application layer, to estimate the duration that UAVs store and carry the packet and eliminate its adverse impact. When the network load is light, trajectory information can be analysed and utilised more accurately. Thus, the consistency model is better established and the detection accuracy is improved (combination 2). Finally, considering the features at all layers (combination 1) achieves the best detection accuracy in all situations.

In summary, features at different layers have their own advantages and disadvantages, and all contribute to the detection accuracy. Through holistic collection, selection, and utilisation of information at all layers from a cross-layer perspective, HOTD achieves complementarities and synergies between features and is thereby able to deal with TDAs in different environments.

5.6.2. Trade-off between overhead and accuracy

HOTD uses message attachment for information collection, which inevitably increases the network overhead. Therefore, we explored the trade-off between the extra overhead and detection accuracy. We chose to minimise the extra overhead while sacrificing a small amount of detection accuracy. To this end, we conducted extensive experiments, and because of space limitations, we only present the final experimental results.

As shown in Table 4, combination 5 achieved a better detection accuracy than combination 1, while greatly reducing the additional overhead. The extra overhead introduced by combination 5 was $7 + 13 + 13 + 8 + 8 + 8 = 57$ bits, which was only 54% of that of combination 1 (105 bits). However, the decrease in the detection accuracy of combination 5 was within 6% of combination 1. Therefore, HOTD effectively achieves a good trade-off between extra overhead and detection accuracy.

5.7. Overhead analysis

In UAV networks, storage and computing resources are relatively sufficient; however, there are low communication resources [3]. Therefore, we conducted experiments to investigate the extra overhead ratio introduced by the transmission of collected information, which can be defined as

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

where N is the number of transmitted messages, D_i is the size of the original payload of message M_i, H_i is the hop count to deliver M_i to the destination, and A_i is the size of the information that each forwarding node attaches to M_i. We set, $A_i = 105$ bits and $D_i = 1400$ B (see Table 2). Table 5 lists the results for the two scenarios and four routing protocols. The extra overhead ratio introduced by HOTD was very small (under 2.5%) in all cases, indicating feasibility and practicability.

5.8. Impact of different variables

In the following experiments, we studied the impact of different variables on detection accuracy.

5.8.1. Impact of TDA duration

We studied the impact of TDA duration on detection accuracy, including absolute and relative TDAs. The experimental results are presented in Figs. 5 and 6.

Table 5
Extra overhead ratio.

	Epidemic Router	*Spray-and-Wait Router*	*Prophet Router*	*MaxProp Router*
Scenario 1	2.28%	1.80%	1.86%	2.30%
Scenario 2	2.21%	1.78%	1.82%	2.17%

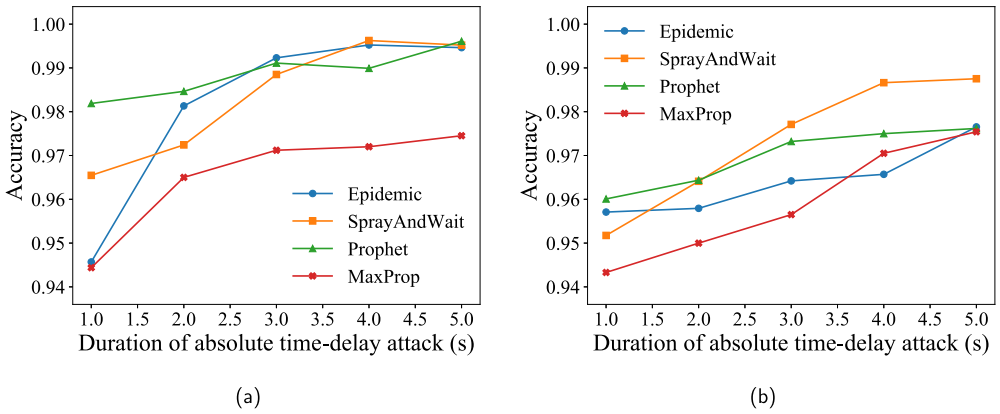


Fig. 5. Impact of duration of absolute time-delay attack on detection accuracy. (a) Scenario 1. (b) Scenario 2.

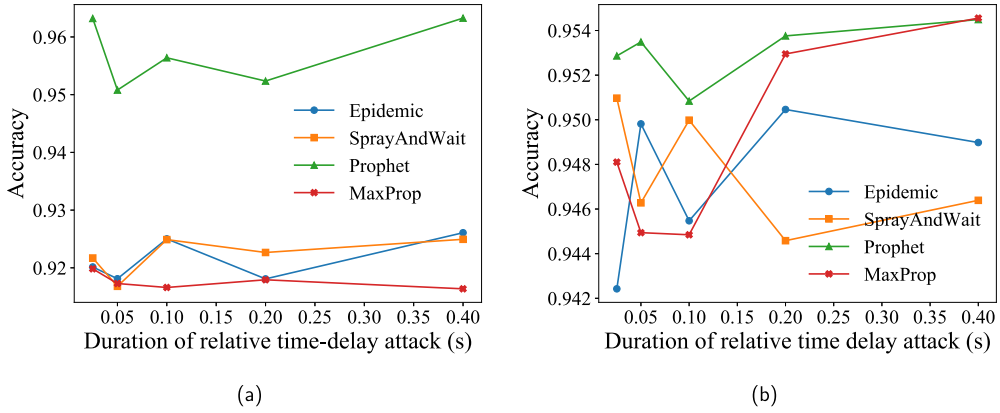


Fig. 6. Impact of duration of relative time-delay attack on detection accuracy. (a) Scenario 1. (b) Scenario 2.

The detection accuracy improved as the absolute TDA duration increased (see Fig. 5) due to the increased TDA duration leading to obvious inconsistencies between the delay-related features and corresponding forwarding delay based on the trained consistency model, thereby exposing the malicious behaviour of the node.

Moreover, to further study the performance of HOTD, we designed a more covert relative TDA. The relative TDA duration depends on the transmission duration of messages (i.e. one-quarter, one-half, one-time, two-times and four-times). The experimental results showed that the detection accuracy of HOTD was higher than 91% in the two scenarios and four routing protocols, as shown in Fig. 6. This is because we performed a holistic collection, extraction, and selection of delay-related information at different layers from a cross-layer perspective, and the trained consistency model could handle TDAs in different environments.

5.8.2. Impact of TDA probability

As shown in Fig. 7, the detection accuracy dropped slightly as the attack probability increased because more TDAs led to more complex network conditions, making it difficult to accurately extract delay-related information. However, the detection accuracy of HOTD remained above 92% in all situations.

Furthermore, the detection accuracies of spray-and-wait and probabilistic routing were better than those of epidemic and MaxProp routing in most situations because both epidemic and MaxProp routing are based on natural flooding. The load of all nodes on these two routing protocols increased, whether they were malicious or benign, resulting in a decreased influence of the MAC layer features on accuracy.

5.8.3. Impact of percentage of malicious nodes

The detection accuracy of HOTD was over 90% in most situations (see Fig. 8), and as the percentage of malicious nodes increased, the detection accuracy decreased. Overall, the experimental results were similar to those of TDA probability. However, the percentage of malicious nodes had a greater impact on detection accuracy than TDA probability because increasing the percentage of malicious nodes in the UAV network adversely affected the neighbouring nodes more widely than increasing the probability of a TDA, thereby quickly and comprehensively degrading the overall performance of the network.

5.8.4. Impact of link quality

As depicted in Fig. 9, when the link quality improved, the detection accuracy of HOTD generally increased, because when the link quality is poor, there are numerous data packet losses and retransmissions, which wastes a significant amount of time and causes an increased network load. This poor network environment causes

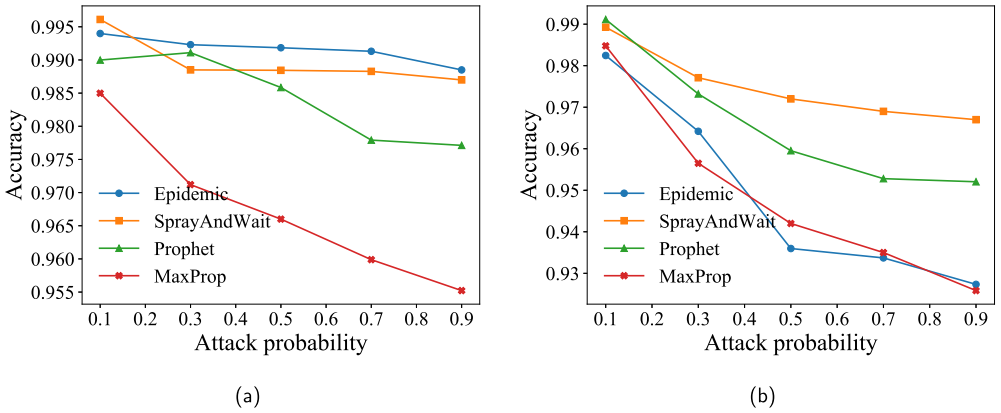


Fig. 7. Impact of attack probability on detection accuracy. (a) Scenario 1. (b) Scenario 2.

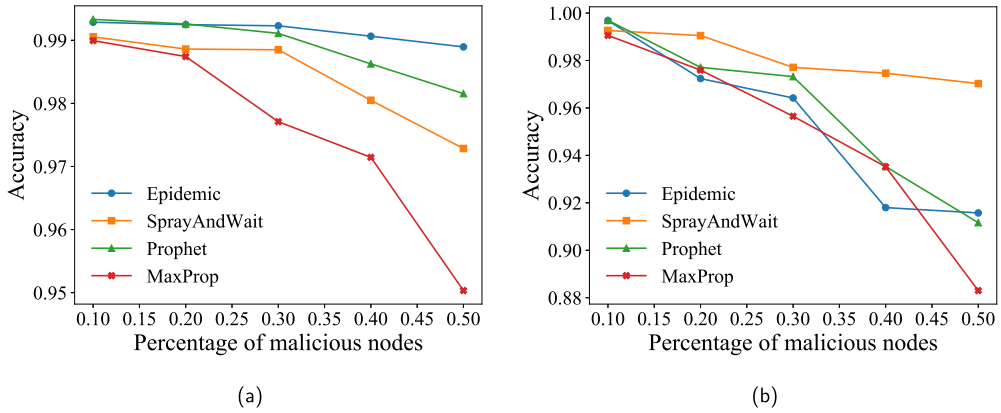


Fig. 8. Impact of percentage of malicious nodes on detection accuracy. (a) Scenario 1. (b) Scenario 2.

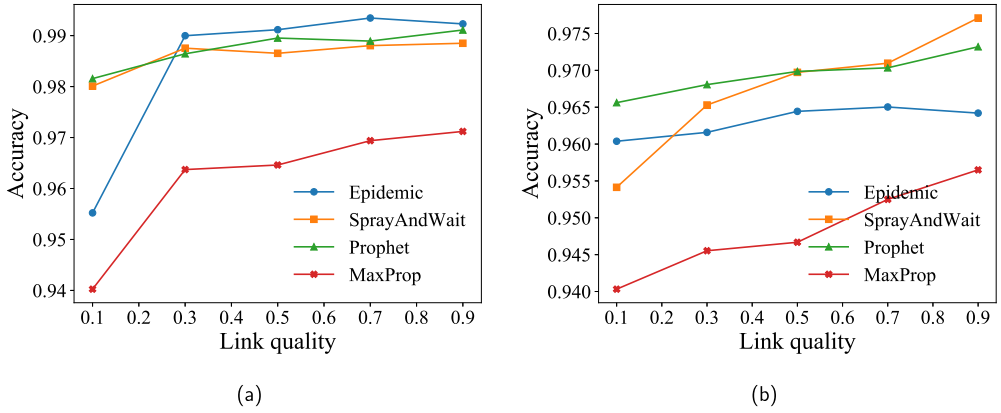


Fig. 9. Impact of link quality on detection accuracy. (a) Scenario 1. (b) Scenario 2.

abnormalities in the forwarding behaviours of the nodes and abnormal fluctuations in the delay of messages, which increases the difficulty of TDA detection. However, the detection accuracy of HOTD was above 94% in all situations.

5.8.5. Impact of message creation interval

We investigated the impact of the message creation interval on the detection accuracy by keeping the total number of injected data packets the same at different intervals. The performance results of HOTD are shown in Fig. 10. The detection accuracy of HOTD was over 91% in all situations. Moreover, as the message creation interval increased, so did the detection accuracy, because network load decreases when the message creation interval increases. Therefore, accurately collecting the delay-related information of each forwarding node improves the detection accuracy.

6. Conclusion

The rapid development and widespread application of UAV networks has resulted in them being vulnerable to internal attacks, such as TDAs, which are easy to implement and difficult to detect. Moreover, the unique characteristics of UAV networks greatly increase the concealment and destructiveness of TDAs. Therefore, there is an urgent need to design efficient mechanisms for accurate detection. However, no research has been conducted on TDA detection in UAV networks.

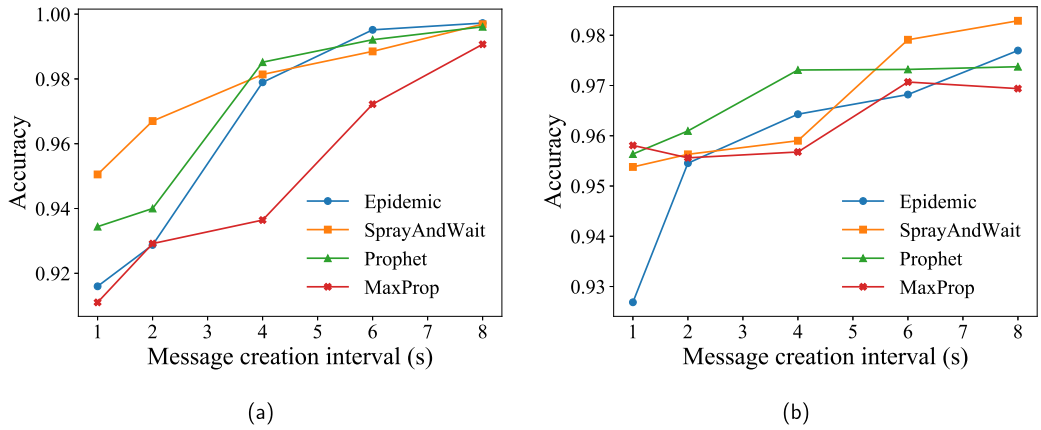


Fig. 10. Impact of message creation interval on detection accuracy. (a) Scenario 1. (b) Scenario 2.

In this paper, we provided a comprehensive and in-depth analysis of TDAs in UAV networks and proposed HOTD. First, we performed a holistic selection of the delay-related features at each layer of the UAV network. Supervised learning was then used to construct a consistency model between these selected features and the corresponding forwarding delay; based on which, the degree of consistency of each node was calculated. Finally, the clustering method was utilised to distinguish malicious from benign nodes according to their degree of consistency. Through extensive experiments, we demonstrated that the performance of HOTD was far superior to that of state-of-the-art detection methods. HOTD achieved a detection accuracy above 85%, with less than 2.5% extra overhead in various UAV network settings and different routing protocols.

CRediT authorship contribution statement

Wenbin Zhai: Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Shanshan Sun:** Investigation, Methodology, Software, Writing – original draft. **Liang Liu:** Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. **Youwei Ding:** Formal analysis, Methodology, Supervision, Writing – review & editing. **Wanying Lu:** Writing – review & editing.

Declaration of competing interest

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

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgment

This work is supported by the National Key R&D Program of China under No. 2020YFB1005902, 2021YFB2700500 and 2021YFB2700502, the Open Fund of Key Laboratory of Civil Aviation Smart Airport Theory and System, Civil Aviation University of China under No. SATS202206, the National Natural Science Foundation of China under No. 82004499, 62076125, 62032025, U20B2049, U20B2050, U21A20467, 61702236 and 6207020639, the Key R&D Program of Guangdong Province under No. 2020B0101090002, the Natural Science Foundation of Jiangsu Province under No. BK20200418, BE2020106, the Guangdong Basic and Applied Basic Research Foundation under No. 2021A1515012650, the Shenzhen Science and Technology Program under No. JCYJ20210324134810028, Public Service Platform for Basic Software and Hardware Supply Chain Guarantee under No. TC210804A.

References

[1] A.S. Almogren, Intrusion detection in edge-of-things computing, J. Parallel Distrib. Comput. 137 (2020) 259–265, https://doi.org/10.1016/j.jpdc.2019.12.008.

[2] S. Aneja, P. Nagrath, G. Purohit, Energy efficient reputation mechanism for defending different types of flooding attack, Wirel. Netw. 25 (7) (2019) 3933–3951, https://doi.org/10.1007/s11276-018-01928-x.

[3] M. Asadpour, K.A. Hummel, D. Giustiniano, S. Draskovic, Route or carry: motion-driven packet forwarding in micro aerial vehicle networks, IEEE Trans. Mob. Comput. 16 (3) (2016) 843–856, https://doi.org/10.1109/TMC.2016.2561291.

[4] J. Burgess, B. Gallagher, D.D. Jensen, B.N. Levine, et al., Maxprop: routing for vehicle-based disruption-tolerant networks, in: Infocom, vol. 6, Barcelona, Spain, 2006.

[5] H. Chi, C. Fu, Q. Zeng, X. Du, Delay wreaks havoc on your smart home: delay-based automation interference attacks, in: 43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022, IEEE, 2022, pp. 285–302.

[6] A. Chriki, H. Touati, H. Snoussi, F. Kamoun, FANET: communication, mobility models and security issues, Comput. Netw. 163 (2019), https://doi.org/10.1016/j.comnet.2019.106877.

[7] Z. Cui, L. Du, P. Wang, X. Cai, W. Zhang, Malicious code detection based on cnns and multi-objective algorithm, J. Parallel Distrib. Comput. 129 (2019) 50–58, https://doi.org/10.1016/j.jpdc.2019.03.010.

[8] X. Dong, Y. Li, C. Lu, G. Hu, Q. Li, Z. Ren, Time-varying formation tracking for uav swarm systems with switching directed topologies, IEEE Trans. Neural Netw. Learn. Syst. 30 (12) (2018) 3674–3685, https://doi.org/10.1109/TNNLS.2018.2873063.

[9] U. Farooq, N. Tariq, M. Asim, T. Baker, A. Al-Shamma'a, Machine learning and the Internet of things security: solutions and open challenges, J. Parallel Distrib. Comput. 162 (2022) 89–104, https://doi.org/10.1016/j.jpdc.2022.01.015.

[10] L. Fu, X. Fu, Z. Zhang, Z. Xu, X. Wu, X. Wang, S. Lu, Joint optimization of multicast energy in delay-constrained mobile wireless networks, IEEE/ACM Trans. Netw. 26 (1) (2018) 633–646, https://doi.org/10.1109/TNET.2018.2790639.

[11] X. Fu, Z. Xu, Q. Peng, J. You, L. Fu, X. Wang, S. Lu, Conmap: a novel framework for optimizing multicast energy in delay-constrained mobile wireless networks, in: Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2017, pp. 1–10.

[12] P. Ganesh, X. Lou, Y. Chen, R. Tan, D.K. Yau, D. Chen, M. Winslett, Learning-based simultaneous detection and characterization of time delay attack in cyber-physical systems, IEEE Trans. Smart Grid (2021), https://doi.org/10.1109/TSG.2021.3058682.

[13] J. Hu, H. Zhang, L. Song, R. Schober, H.V. Poor, Cooperative Internet of uavs: distributed trajectory design by multi-agent deep reinforcement learning, IEEE Trans. Commun. 68 (11) (2020) 6807–6821, https://doi.org/10.1109/TCOMM.2020.3013599.

[14] J.P. Jeong, J. Kim, T. Hwang, F. Xu, S. Guo, Y.J. Gu, Q. Cao, M. Liu, T. He, Tpd: travel prediction-based data forwarding for light-traffic vehicular networks, Comput. Netw. 93 (2015) 166–182, https://doi.org/10.1016/j.comnet.2015.10.016.

[15] P. Kaliyar, W.B. Jaballah, M. Conti, C. Lal Lidl, Localization with early detection of sybil and wormhole attacks in iot networks, Comput. Secur. 94 (2020) 101849, https://doi.org/10.1016/j.cose.2020.101849.

[16] A. Keränen, J. Ott, T. Kärkkäinen, The one simulator for dtn protocol evaluation, in: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, 2009, pp. 1–10.

[17] D. Kwon, K. Natarajan, S.C. Suh, H. Kim, J. Kim, An empirical study on network anomaly detection using convolutional neural networks, in: 38th IEEE International Conference on Distributed Computing Systems, ICDCS 2018, Vienna, Austria, July 2-6, 2018, IEEE Computer Society, 2018, pp. 1595–1598.

[18] X. Li, L. Liu, L. Wang, J. Xi, J. Peng, J. Meng, Trajectory-aware spatio-temporal range query processing for unmanned aerial vehicle networks, Comput. Commun. 178 (2021) 271–285, https://doi.org/10.1016/j.comcom.2021.08.008.

[19] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, ACM SIGMOBILE Mobile Comput. Commun. Rev. 7 (3) (2003) 19–20, https://doi.org/10.1145/961268.961272.

[20] L. Liu, X. Xu, Y. Liu, Z. Ma, J. Peng, A detection framework against cpma attack based on trust evaluation and machine learning in iot network, IEEE Int. Things J. (2021), https://doi.org/10.1109/JIOT.2020.3047642.

[21] X. Liu, M. Abdelhakim, P. Krishnamurthy, D. Tipper, Identifying malicious nodes in multihop iot networks using diversity and unsupervised learning, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.

[22] X. Lou, C. Tran, R. Tan, D.K. Yau, Z.T. Kalbarczyk, A.K. Banerjee, P. Ganesh, Assessing and mitigating impact of time delay attack: case studies for power grid controls, IEEE J. Sel. Areas Commun. 38 (1) (2019) 141–155, https://doi.org/10.1109/JSAC.2019.2951982.

[23] X. Lou, C. Tran, D.K. Yau, R. Tan, H. Ng, T.Z. Fu, M. Winslett, Learning-based time delay attack characterization for cyber-physical systems, in: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2019, pp. 1–6.

[24] Z. Ma, L. Liu, W. Meng, Dconst: detection of multiple-mix-attack malicious nodes using consensus-based trust in iot networks, in: Australasian Conference on Information Security and Privacy, Springer, 2020, pp. 247–267.

[25] Z. Ma, L. Liu, W. Meng, Towards multiple-mix-attack detection via consensus-based trust management in iot networks, Comput. Secur. 96 (2020) 101898, https://doi.org/10.1016/j.cose.2020.101898.

[26] Z. Ma, L. Liu, W. Meng, Eld: adaptive detection of malicious nodes under mix-energy-depleting-attacks using edge learning in iot networks, in: International Conference on Information Security, Springer, 2020, pp. 255–273.

[27] K. Meng, D. Li, X. He, M. Liu, Space pruning based time minimization in delay constrained multi-task uav-based sensing, IEEE Trans. Veh. Technol. 70 (3) (2021) 2836–2849, https://doi.org/10.1109/TVT.2021.3061243.

[28] D. Mishra, E. Natalizio, A survey on cellular-connected UAVs: design challenges, enabling 5g/b5g innovations, and experimental advancements, Comput. Netw. 182 (2020) 107451, https://doi.org/10.1016/j.comnet.2020.107451.

[29] B. Moussa, M. Debbabi, C. Assi, A detection and mitigation model for ptp delay attack in an IEC 61850 substation, IEEE Trans. Smart Grid 9 (5) (2016) 3954–3965, https://doi.org/10.1109/TSG.2016.2644618.

[30] B. Moussa, M. Kassouf, R. Hadjidj, M. Debbabi, C. Assi, An extension to the precision time protocol (PTP) to enable the detection of cyber attacks, IEEE Trans. Ind. Inform. 16 (1) (2019) 18–27, https://doi.org/10.1109/TII.2019.2943913.

[31] J. Neyer, L. Gassner, C. Marinescu, Redundant schemes or how to counter the delay attack on time synchronization protocols, in: 2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), IEEE, 2019, pp. 1–6.

[32] T. Pelkonen, S. Franklin, J. Teller, P. Cavallaro, Q. Huang, J. Meza, K. Veeraraghavan, Gorilla: a fast, scalable, in-memory time series database, Proc. VLDB Endow. 8 (12) (2015) 1816–1827, https://doi.org/10.14778/2824032.2824078, http://www.vldb.org/pvldb/vol8/p1816-teller.pdf.

[33] J. Peng, H. Gao, L. Liu, N. Li, X. Xu, Tbm: an efficient trajectory-based multicast routing protocol for sparse UAV networks, in: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2020, pp. 867–872.

[34] J. Peng, H. Gao, L. Liu, Y. Wu, X. Xu, Fntar: a future network topology-aware routing protocol in UAV networks, in: 2020 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2020, pp. 1–6.

[35] T.N.D. Pham, C.K. Yeo, Detecting colluding blackhole and greyhole attacks in delay tolerant networks, IEEE Trans. Mob. Comput. 15 (5) (2015) 1116–1129, https://doi.org/10.1109/TMC.2015.2456895.

[36] H. Pirayesh, H. Zeng, Jamming attacks and anti-jamming strategies in wireless networks: a comprehensive survey, IEEE Commun. Surv. Tutor. 24 (2) (2022) 767–809, https://doi.org/10.1109/COMST.2022.3159185.

[37] Y. Qin, M.A. Kishk, M.-S. Alouini, Performance evaluation of UAV-enabled cellular networks with battery-limited drones, IEEE Commun. Lett. 24 (12) (2020) 2664–2668, https://doi.org/10.1109/LCOMM.2020.3013286.

[38] T. Rose, K. Kifayat, S. Abbas, M. Asim, A hybrid anomaly-based intrusion detection system to improve time complexity in the Internet of energy environment, J. Parallel Distrib. Comput. 145 (2020) 124–139, https://doi.org/10.1016/j.jpdc.2020.06.012.

[39] K. Schneider, B. Zhang, L. Benmohamed, Hop-by-hop multipath routing: choosing the right nexthop set, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 2273–2282.

[40] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, Appl. Soft Comput. 97 (2020) 105524, https://doi.org/10.1016/j.asoc.2019.105524.

[41] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005, pp. 252–259.

[42] S. Sun, Z. Ma, L. Liu, H. Gao, J. Peng, Detection of malicious nodes in drone ad-hoc network based on supervised learning and clustering algorithms, in: 2020 16th International Conference on Mobility, Sensing and Networking (MSN), IEEE, 2020, pp. 145–152.

[43] Y. Sun, S. Wang, D. Huang, Y. Sun, A. Hu, J. Sun, A multiple hierarchical clustering ensemble algorithm to recognize clusters arbitrarily shaped, Intell. Data Anal. 26 (5) (2022) 1211–1228, https://doi.org/10.3233/IDA-216112.

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

[45] S. ur Rehman, M. Khaliq, S.I. Imtiaz, A. Rasool, M. Shafiq, A.R. Javed, Z. Jalil, A.K. Bashir, DIDDOS: an approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (GRU), Future Gener. Comput. Syst. 118 (2021) 453–466, https://doi.org/10.1016/j.future.2021.01.022.

[46] A. Vahdat, D. Becker, et al., Epidemic Routing for Partially Connected Ad Hoc Networks, 2000.

[47] F. van Wyk, Y. Wang, A. Khojandi, N. Masoud, Real-time sensor anomaly detection and identification in automated vehicles, IEEE Trans. Intell. Transp. Syst. 21 (3) (2020) 1264–1276, https://doi.org/10.1109/TITS.2019.2906038.

[48] H. Wang, J. Gu, S. Wang, An effective intrusion detection framework based on svm with feature augmentation, Knowl.-Based Syst. 136 (2017) 130–139, https://doi.org/10.1016/j.knosys.2017.09.014.

[49] Q. Wu, J. Xu, Y. Zeng, D.W.K. Ng, N. Al-Dhahir, R. Schober, A.L. Swindlehurst, A comprehensive overview on 5g-and-beyond networks with UAVs: from communications to sensing and intelligence, IEEE J. Sel. Areas Commun. (2021), https://doi.org/10.1109/JSAC.2021.3088681.

[50] K. Xiahou, Y. Liu, Q. Wu, Robust load frequency control of power systems against random time-delay attacks, IEEE Trans. Smart Grid 12 (1) (2020) 909–911, https://doi.org/10.1109/TSG.2020.3018635.

[51] L. Xiao, Y. Ding, J. Huang, S. Liu, Y. Tang, H. Dai, UAV anti-jamming video transmissions with qoe guarantee: a reinforcement learning-based approach, IEEE Trans. Commun. 69 (9) (2021) 5933–5947, https://doi.org/10.1109/TCOMM.2021.3087787.

[52] F. Xiong, A. Li, H. Wang, L. Tang, An sdn-mqtt based communication system for battlefield uav swarms, IEEE Commun. Mag. 57 (8) (2019) 41–47, https://doi.org/10.1109/MCOM.2019.1900291.

[53] L. Yang, L. Liu, Z. Ma, Y. Ding, Detection of selective-edge packet attack based on edge reputation in iot networks, Comput. Netw. 188 (2021) 107842, https://doi.org/10.1016/j.comnet.2021.107842.

[54] X. Yang, C. Deng, F. Zheng, J. Yan, W. Liu, Deep spectral clustering using dual autoencoder network, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 4066–4075, http://openaccess.thecvf.com/content_CVPR_2019/html/Yang_Deep_Spectral_Clustering_Using_Dual_Autoencoder_Network_CVPR_2019_paper.html.

[55] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, C. Bettstetter, Drone networks: communications, coordination, and sensing, Ad Hoc Netw. 68 (2018) 1–15, https://doi.org/10.1016/j.adhoc.2017.09.001.

[56] N.R. Zema, D. Quadri, S. Martin, O. Shrit, Formation control of a mono-operated uav fleet through ad-hoc communications: a q-learning approach, in: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, 2019, pp. 1–6.

[57] Z. Zhang, R. Deng, P. Cheng, Q. Wei, On feasibility of coordinated time-delay and false data injection attacks on cyber-physical systems, IEEE Int. Things J. 9 (11) (2022) 8720–8736, https://doi.org/10.1109/JIOT.2021.3118065.

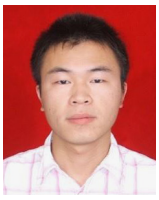
[58] Y. Zhao, A.K. Shrivastava, K.L. Tsui, Regularized gaussian mixture model for high-dimensional clustering, IEEE Trans. Cybern. 49 (10) (2019) 3677–3688, https://doi.org/10.1109/TCYB.2018.2846404.

[59] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, M. Guizani, When mobile crowd sensing meets uav: energy-efficient task assignment and route planning, IEEE Trans. Commun. 66 (11) (2018) 5526–5538, https://doi.org/10.1109/TCOMM.2018.2857461.

Wenbin Zhai received the B.S. degree from Nanjing University Of Chinese Medicine, Nanjing, Jiangsu Province, China in 2020. He is currently working toward the M.S. degree in Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China. His research interest includes wireless sensor networks.

Shanshan Sun received the B.S. degree from China University of Geosciences, Beijing, China in 2016. She is currently working towards the M.S. degree in Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China. Her research interest includes Drone Network Security.

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



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

Wanying Lu graduated from Henan Polytechnic University with a bachelor's degree in 2021. At present, she is studying for a master's degree in the collage of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. Her main research direction is time series big data storage and data mining.