Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.1, Feb. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.1.39

시퀀스 유사도 기반 무인 비행체 이상 탐지 시스템

서 강 욱,[1†] 김 휘 강[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수)

Sequence Based Anomaly Detection System for Unmanned Aerial Vehicle

Kang Uk Seo,[1†] Huy Kang Kim[2‡]
[1,2]School of Cybersecurity, Korea University (Graduate student, Professor)

요 약

본 논문에서는 무인 비행체 내부 네트워크의 이상 징후를 탐지하는 시퀀스 기반 이상 탐지 시스템을 제안한다. 제안하는 이상 탐지 시스템은 무인 비행체가 지상 통제 시스템에 주기적으로 전송하는 상태 메시지 시퀀스들 간의 유사도를 측정하여 이상 징후를 탐지한다. 본 연구에서는 무인 비행체 내부 네트워크에서 수행 가능한 악의적인 메시지 주입 공격 세 가지를 정의하고, 해당 공격 기법들을 Pixhawk4 쿼드콥터에서 시뮬레이션하였다. 결과적으로, 제안하는 이상 탐지 시스템은 96% 이상의 정확도로 비정상 시퀀스를 탐지할 수 있었다.

ABSTRACT

In this paper, we propose an anomaly detection system (ADS) to detect anomalies of the in-vehicle network for unmanned aerial vehicle (UAV). The proposed ADS detects the anomalies by measuring the similarity of status messages sequences periodically sent by the UAV to the ground control system. We defined three types of malicious message injection attacks that can be performed on the in-vehicle network of UAV and simulated those attack techniques in the Pixhawk4 quadcopter. The proposed ADS can detect abnormal sequences with accuracy of higher than 96%.

Keywords: Anomaly detection system, Sequence data, Unmanned aerial vehicle

I. 서 론

무인 이동체는 재해 지역 모니터링, 국경 감시, 물품 배달 등 다양한 영역에서 활용되고 있다[1]. 단일 무인 이동체가 제공하는 기능이 늘어나고 비행체 내부 구조가 복잡해짐에 따라 무인 비행체, 무인 잠수정, 무인 자동차 등과 같은 최신의 무인 이동체는 UAVCAN (Uncomplicated Application-layer Vehicular Computing and Networking) 프로토콜과 같은 효율적이고 신뢰성 높은 통신 프로토콜을 기반으로 내부 네트워크를 구성한다[2].

대부분의 무인 이동체들과 이들의 내부 네트워크에서 사용되는 통신 프로토콜은 정보 보호 기능보다는 고신뢰/저비용과 같은 성능을 중시하여 설계되었다[3]. 이에 따라, 트래픽 암호화나 심층 패킷 검사 등 이상 징후 탐지 매커니즘이 거의 적용되지 않은 군용 드론의 내/외부 네트워크에 공격자가 침투하여 드론을 해킹하고 제어권을 탈취하는 등의 문제가 있었다[4].

본 논문에서는 UAVCAN 프로토콜을 기반으로 구성된 무인 비행체 내부 네트워크에서 수행 가능한 악의적인 메시지 주입 공격 세 가지를 정의하고, 해당 공격 기법들로 인해 유발된 이상 징후를 탐지하는 이상 탐지 시스템을 제안한다. 제안하는 이상 탐지 시스템은 무인 비행체와 지상 통제 시스템 간 통신 트래픽의 패턴을 분석하여 이상 징후를 탐지한다.

Received(11. 22. 2021), Modified(1st: 12. 28. 2021, 2nd: 02. 08. 2022), Accepted(02. 08. 2022)
† 주저자, tjrkddnr@korea.ac.kr
‡ 교신저자, cenda@korea.ac.kr(Corresponding author)

본 논문의 기여는 다음과 같이 요약된다:

- ◉ 제안하는 공격 기법은 UAVCAN 프로토콜의 구조적 결함을 이용하므로 UAVCAN 프로토콜을 지원하는 모든 종류의 전자 제어 장치에 시도해볼 수 있다.
- ◉ 제안하는 이상 탐지 시스템은 무인 비행체 외부에 위치하므로 무인 비행체에 추가적인 부하를 일으키지 않는다.
- ◉ 제안하는 이상 탐지 시스템은 정상 운행 상태일 때 수집한 데이터만을 활용하여 이상 징후를 효과적으로 식별할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 기존 연구 사례를 소개한다. 3장에서는 제안하는 기법 및 이와 관련된 배경 지식을 소개한다. 4장에서는 제안하는 기법을 소개하고, 5장에서는 실제 무인 비행체에서 수집한 데이터를 이용하여 이상 탐지 실험을 수행한 결과를 보여준다. 마지막으로 6장에서는 결론과 한계점, 후속 연구 방향을 소개한다.

II. 관련 연구

이동체 네트워크에 대한 침입 또는 이상 징후 탐지 기법은 탐지에 사용한 데이터가 생성된 위치에 따라 내부 네트워크 데이터, 외부 네트워크 데이터, 그리고 센서 데이터의 3가지로 분류할 수 있다.

2.1 내부 네트워크 데이터

내부 네트워크란 이동체 내부 컴포넌트들이 데이터를 서로 주고받는 환경을 의미한다. CAN bus는 이동체 내부 네트워크의 전형적인 예이다.

CAN bus로 구성된 이동체 내부 네트워크에서 수행 가능한 공격 기법들에는 message injeciton attack, synthetic attack, spoofing attack, fuzzy, DoS attack 등이 존재한다[5]. 일반적으로, 내부 네트워크 데이터를 이용한 침입 또는 이상 탐지 연구의 목적은 이러한 공격들의 발생 여부를 탐지하는 것이다.

Olufowobi 등[6]은 CAN bus 상에서 CAN ID의 발생 주기 패턴을 분석하여 등의 공격을 탐지하였다. Sun 등[7]은 동적 CAN ID를 이용하여 replay attack, message forgery attack, bus off attack 등을 방지하는 방법론을 제안하였다. Tariq 등[8]은 CAN frame으로부터 timestamp, CAN ID, DLC, data 등의 정보를 추출하고, 이들을 RNN 기반 탐지 모델에 입력하여 공격을 탐지하였다. Wang 등[9]은 공격이 수행 중일 때 수집한 CAN ID들의 발생 빈도 엔트로피와 정상 주행 상태일 때 수집한 CAN ID들의 발생 빈도 엔트로피를 비교하여 공격을 탐지하였다.

2.2 외부 네트워크 데이터

외부 네트워크란 다수의 이동체 또는 지상 통제 시스템과 같은 외부 컴포넌트들이 서로 데이터를 주고받는 환경을 의미한다.

Arthur는 ONE 시뮬레이터를 기반으로 다수의 무인 이동체를 운용하는 가상 환경을 구축하고, multi-class SVM을 이용하여 false GPS attack, DoS attack 등을 탐지하였다. Birnbaum 등[10]은 RPE(Recursive Parameter Estimation)기법을 이용하여 무인 비행체와 지상 통제 시스템 간 통신 채널에서 발생 가능한 공격을 탐지하였다. Kim 등[11]은 SITL(Software In The Loop) 환경에서 다수의 가상 무인 비행체를 운용하는 환경을 구축하고, false data injection attack이 진행 중일 때 비행체로부터 수집한 데이터를 GAN(Generative Adversarial Network) 모델을 통해 증식시키는 연구를 수행하였다. Loukas 등[12]은 클라우드 기반으로 운행되는 무인 이동체로부터 네트워크 트래픽, CPU 사용량, 저장 공간 사용량, 배터리 사용량, 가속도계 값 등을 수집하고 이를 RNN 기반 딥러닝 모델에 입력하여 침입을 탐지하였다. Sedjelmaci 등[13]은 다수의 무인 비행체들이 서로 통신하는 환경에서 GPS spoofing, GPS jamming, gray hole/black hole attack 등을 탐지하는 룰 기반 침입 탐지 방법론을 제안하였다.

2.3 센서 데이터

센서 데이터는 이동체에 부착된 GPS, 자이로스코프, radar 등의 센서 장비들이 외부 환경을 관찰하여 수집한 데이터를 의미한다.

Meng 등[14]은 GPS 정보, 광학계 정보 등을 이용하는 서로 다른 여러 탐지 모델을 결합하여 sensor spoofing attack을 탐지하였다. Kapoor 등[15]은 FMCW 안테나로부터 수집한 정보를 이용

하여 RADER signal spoofing attack과 fake signal attack을 탐지하였다. Panice 등[16]은 MATLAB 기반의 시뮬레이션 환경에서 가상의 무인 비행체를 생성하고, SVM 모델을 이용하여 GPS jamming과 false GPS signal injection attack 등을 탐지하였다.

III. 무인 비동체 내/외부 네트워크 구조

Fig. 1.은 본 연구에서 구성한 무인 비행체 내/외부 네트워크의 구조를 보여준다. 무인 비행체 내부 네트워크는 비행 컨트롤러(Flight Controller), 변속기(Electric Speed Controller), GPS, 원격 송수신 장치 등으로 이루어진다. 비행 컨트롤러는 비행체 내부 컴포넌트들로부터 센서 데이터를 수집 및 분석하여 안정적인 비행을 위한 신호를 생성하고, 이를 변속기에 전달한다[17]. 이때 비행체 내부 컴포넌트들은 유선으로 연결되어 UAVCAN 프로토콜을 기반으로 데이터를 주고 받는다[3].

비행체 외부 네트워크는 무인 비행체, 지상 통제 시스템, 조종기 등으로 이루어진다. 이들은 Telemetry 장비나 RF 안테나와 같은 무선 송수신 장치를 통해 통신하며, 일반적으로 MAVLink (Micro Air Vehicle Link) 프로토콜을 기반으로 데이터를 주고 받는다[18].

3.1 UAVCAN 프로토콜

UAVCAN은 이동체 내부 컴포넌트들 간 데이터 통신에 사용되는 응용 계층 프로토콜이다. 많은 무인 비행체들이 전송 계층 프로토콜로 CAN을, 응용 계층 프로토콜로 UAVCAN을 사용한다[3].

UAVCAN 패킷은 송신 노드 ID, 수신 노드 ID, Message type ID, Service type ID, PAYLOAD, Tail byte 등의 필드로 이루어져있다.

수신자는 Message type ID 필드에 기록된 값에 따라 PAYLOAD 필드에 담긴 데이터의 의미를 해석한다. 각각의 Message type ID들의 해석 방식은 DSDL(Data Structure Description Language) 형태로 사전에 노드들에 기록되며, 노드들은 DSDL에 명시된 Message type ID만을 해석할 수 있다.

한번에 전송하고자 하는 데이터의 크기가 너무 클 경우, UAVCAN은 데이터를 여러 조각으로 분할하여 다수의 패킷에 나누어 보내는 기능을 제공한다. 이때 Tail byte에 어떤 값이 기록되어 있는지에 따라 해당 패킷이 데이터의 시작에 해당하는지, 중간에 해당하는지, 또는 마지막에 해당하는지 결정할 수 있다.

UAVCAN은 구독-발행 모델(publish-subscribe model)과 요청-응답 모델(request-response model)의 두 가지 통신 방식을 제공한다. 구독-발행 모델에서는 데이터 내에 송신 노드 ID만 명시되어 있고, 특정 노드가 네트워크 상에 데이터를 브로드캐스팅하면 다른 모든 노드들은 이 데이터를 수신한다. 요청-응답 모델에서는 데이터 내에 송신 노드 ID와 수신 노드 ID가 명시되어 있어 요청 노드와 응답 노드의 두 노드 사이에서만 통신이 이루어진다. 요청 노드가 특정 Service type ID를 갖는 메시지를 보내면 응답 노드는 이에 대한 적절한 작업을 수행하는 식이다.

3.2 MAVLink 프로토콜

MAVLink는 지상 통제 시스템, 무인 이동체 간의 효율적이고 신뢰할 수 있는 통신을 제공하기 위해 설계된 통신 프로토콜로, 무인 이동체의 Autopilot 시스템 구축에 필요한 다양한 기능들을 제공한다. MAVLink는 무인 비행체 뿐 아니라 무인 자동차, 무인 잠수정 등 다양한 무인 이동체들에서 사용된다. MAVLink는 흔히 드론으로 알려진 중소형 무인 비행체에 가장 일반적으로 채택되는 통신 프로토콜 중 하나이다[18].

MAVLink 패킷은 송신자 ID, 수신자 ID, MSG ID, PAYLOAD, CHECKSUM 등의 필드들로 이루어져있다. 수신자는 MSG ID 필드에 기록된 값에 따라 PAYLOAD 필드에 담긴 데이터의 의미를 해석한다.

IV. 제안하는 기법

4.1 공격자 모델

4.1.1 무인 비행체 내부 네트워크 메시지 주입 공격

본 연구에서는 Fig. 1.과 같이 UAVCAN 프로토콜을 기반으로 구성된 무인 비행체 내부 네트워크에 악의적인 노드(Malicious node)가 존재하는 상황을 가정하였다. 악의적인 노드의 제약 사항은 다음

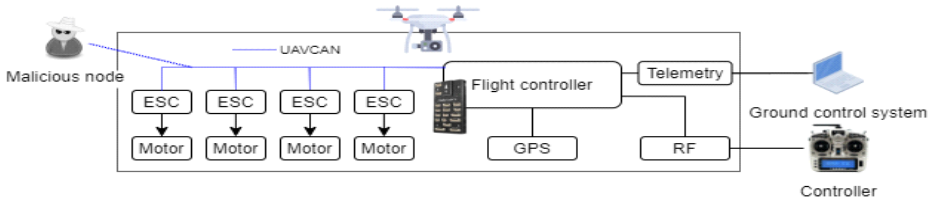

Fig. 1. UAV network topology

과 같다.

◉ 악의적인 노드는 무인 비행체 내부 네트워크에 물리적으로 연결되어 다른 노드들과 통신할 수 있다.

◉ 악의적인 노드는 UAVCAN 통신에 필요한 DSDL 정보를 알지 못한다.

위 제약 사항을 고려하여, 이동체 내부 컴포넌트들의 정상적인 작동을 방해하는 세 종류의 악의적인 메시지 주입 공격 기법들을 정의한다.

◉ Wrong end : 정상 노드들이 주고 받은 UAVCAN 패킷들을 미리 수집하고, 이를 무작위로 재전송한다. 이때, 분할되지 않고 전송되는 데이터를 분할 전송되는 것으로 Tail byte 필드를 조작한다. 해당 공격은 수신 노드 측이 의미 없는 분할 데이터를 계속 기다리게 만들어 저장 공간이 낭비되는 등의 오작동을 유발한다.

◉ Service request flooding : 정상 노드들이 주고 받은 UAVCAN 패킷들 중 요청-응답 모델에 해당하는 패킷을 수집하고, 특정 노드에 비정상적으로 많은 요청 메시지를 전송한다. 해당 공격은 공격 대상 노드에 서비스 거부(Denial of Service)를 유발한다.

◉ Fuzzing : 정상 노드들이 주고 받은 UAVCAN 패킷들을 미리 수집하고, PAYLOAD 필드의 값을 무작위로 변경하여 재전송한다. 해당 공격은 다양한 예측 불가능한 효과를 야기하여 노드들의 정상적인 작동을 방해한다.

4.1.2 공격 영향 관찰

Fig. 2.는 본 논문에서 정의한 악의적인 메시지 주입 공격을 무인 비행체 내부 네트워크에서 시뮬레이션했을 때 무인 비행체에 끼친 영향을 보여준다. 공격이 진행 중일 때 비행 컨트롤러의 CPU 부하율이 48.63% 에서 60.17%로 급격하게 상승하거나 간헐적으로 멸멸 변속기와 비행 컨트롤러 간 연결이

```
Processes: 26 total, 4 running, 22 sleeping, max FDs: 15
CPU usage: 48.63% tasks, 2.63% sched, 48.74% idle
DMA Memory: 5120 total, 1536 used 1536 peak
```

Fig. 2(a). CPU load rate of flight controller in normal state

```
Processes: 26 total, 4 running, 22 sleeping, max FDs: 15
CPU usage: 60.17% tasks, 3.12% sched, 36.71% idle
DMA Memory: 5120 total, 1536 used 1536 peak
```

Fig. 2(b). CPU load rate of flight controller in attack state

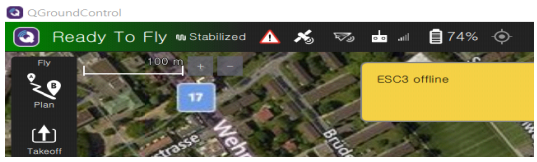

Fig. 2(c). ESC(Electric Speed Controller) connection lost alert shown in ground control system

Table 1. Impacts of each attack technique

Attack technique	Impacts of attack
Wrong end	◉ Some of the ESCs are intermittently lost their connection with the flight controller.
Service request flooding	◉ CPU load rate of flight controller increased.
Fuzzing	◉ Some of the ESCs are intermittently lost their connection with the flight controller. ◉ RPM of motors changed abnormally.

끊어지는 등 무인 비행체의 정상 작동을 저해하는 다양한 결과가 관측되었다. Table 1.은 각 공격 기법의 영향을 정리한 것이다.

4.2 시퀀스 기반 이상 탐지 시스템

2장에서 알 수 있듯이, 기존의 이동체 네트워크 공격 탐지 연구들은 이동체 내부에서 발생한 CAN 트래픽 혹은 이동체 내 외부에서 발생한 센서 데이터 등을 분석하여 공격을 탐지하였다. 하지만, 중소형 무인 비행체의 경우 내부 컴퓨팅 능력의 한계로 인해 이동체 내부에서 공격을 탐지 연산을 수행하기에 적합하지 않다.

한편, 무인 비행체는 주기적으로 지상 통제 시스템에 MAVLink 기반 상태 메시지를 전송한다. 배터리 잔량, GPS 위치 정보, 고도, 모터 RPM, 가속도계 정보 등 다양한 종류의 상태 메시지가 존재하며, 각각의 상태 메시지는 특정 MSG ID를 갖는 MAVLink 패킷으로 표현된다. Fig. 3.은 지상 통제 시스템이 무인 비행체로부터 수신한 상태 메시지들의 갱신 주기를 보여준다. 여러 상태 메시지들이 일정한 주기로 수신되고 있는 것을 알 수 있다. 본 연구에서는 악의적인 노드에 의해 비행체 내부 컴포넌트에 오작동이 발생할 시 비행체가 상태 메시지들을 제대로 생성하거나 송신할 수 없을 것이라 가정하였다. 이 가정을 토대로, 무인 비행체의 상태 메시지 시퀀스들 간의 유사도를 계산하여 무인 비행체의 이상 여부를 탐지하는 시스템을 제안한다. 제안하는 이상 탐지 시스템은 무인 비행체 외부에 위치하므로 비행체에 추가적인 부하를 일으키지 않는다.

Fig. 4.는 본 논문에서 제안하는 시퀀스 기반 이상 탐지 시스템의 동작 흐름도이다. 제안하는 시스템은 데이터 수집부(Data Collection Module), 피

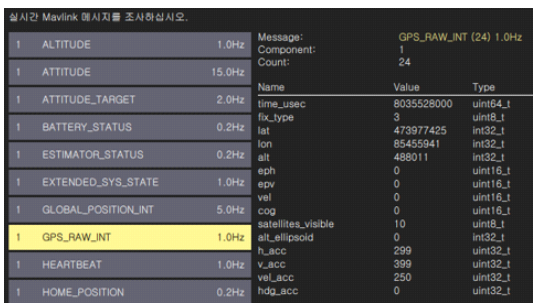

Fig. 3. Periodic updates of status messages

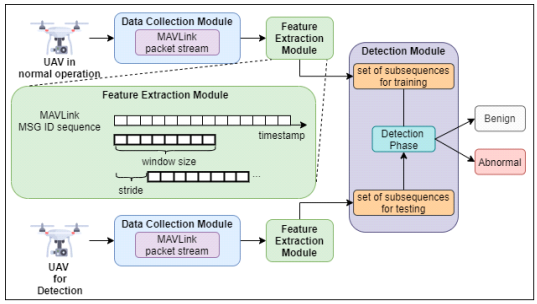

Fig. 4. Overall process of the proposed anomaly detection system

쳐 추출부(Feature Extraction Module), 탐지부(Detection Module)의 세 가지 모듈로 이루어져 있다.

4.2.1 데이터 수집부

데이터 수집부(Data Collection Module)는 무인 비행체가 지상 통제 시스템에 전송한 MAVLink 패킷들로 MSG ID 필드 값만 추출하고 이들을 수신 시간 순으로 나열한 시퀀스를 생성한다.

시퀀스란 유한한 크기를 갖는 심볼 집합에 속한 심볼들의 나열을 의미하며, 하나의 시퀀스는 유한한 길이를 갖는다. 예를 들어, MAVLink 프로토콜의 MSG ID 필드가 가질 수 있는 값들의 집합 $\{ID_1, ID_2, ID_3, ID_4, ID_5 \ldots ID_N\}$에 대해, MSG ID 필드 값들의 나열 $ID_2ID_3ID_5ID_1ID_4$을 하나의 시퀀스로 볼 수 있다.

4.2.2 피쳐 추출부

피쳐 추출부(Feature Extraction Module)는 데이터 수집부가 생성한 MSG ID 시퀀스를 슬라이딩 윈도우 방식으로 분할하여 여러 개의 부분 시퀀스(subsequence)들을 생성한다. 부분 시퀀스란 원본 시퀀스의 심볼 순서를 변경하지 않고 일부 심볼을 삭제하여 생성한 짧은 길이의 시퀀스를 의미한다. 예를 들어, 시퀀스 ID_2ID_3, ID_3ID_5, ID_5ID_1 등은 원본 시퀀스 $ID_2ID_3ID_5ID_1ID_4$의 부분 시퀀스로 볼 수 있다.

4.2.3 탐지부

탐지부(Detection Module)는 피쳐 추출부가

생성한 부분 시퀀스들 간의 유사도를 계산하여 이상 여부를 판단한다.

분석 대상 부분 시퀀스 t의 유사도 점수 $SimilarityScore(t)$는 t와 정상 운행 상태일 때 수집한 부분 시퀀스들 간 유사도 평균으로 주어지며, 수식 (1)과 같이 정의된다.

$$SimilarityScore(t) = \frac{\sum_{s \in S_{train}} similarity(t,s)}{n(S_{train})} \tag{1}$$

이때 S_{train}는 정상 운행 상태일 때 수집한 부분 시퀀스들의 집합이고, $n(S_{train})$은 집합 S_{train}의 크기이다. $similarity(t,s)$는 두 시퀀스 t와 s의 유사도를 나타내는 0과 1 사이의 값으로, 두 시퀀스가 유사할수록 유사도는 1에 가까운 값을 가진다. 따라서, Detection module은 분석 대상 시퀀스 t가 정상 운행 상태일 때 수집한 시퀀스와 다른 패턴을 띈다면 t의 유사도 점수는 낮은 값을 가진다.

유사도 점수가 $threshold$ 보다 작은 값을 가질 경우, 분석 대상 부분 시퀀스 t를 비정상 시퀀스로 탐지한다. $threshold$는 S_{train}에 속한 부분 시퀀스들의 유사도 점수 중 최솟값이며, 수식(2)와 같이 정의된다.

$$threshold = \min(SimilarityScore(s)), \quad where\ s \in S_{train} \tag{2}$$

본 연구에서는 잘 알려진 아래 네 종류의 시퀀스 유사도 측정 알고리즘[19] 중 하나를 선택하여 유사도 점수를 계산하였다.

4.2.3.1 nLCS

nLCS (normalized Longest Common Subsequence)는 두 시퀀스에 공통으로 존재하는 부분 시퀀스들의 길이를 기반으로 유사도를 계산하는 알고리즘이다[20].

$nLCS(s_i, s_j)$는 수식 (3)과 같이 주어진다.

$$similarity(s_i, s_j) = nLCS(s_i, s_j) = \frac{|LCS(s_i, s_j)|}{\sqrt{|s_i||s_j|}} \tag{3}$$

4.2.3.2 Levenshtein Distance

Levenshtein Distance는 편집 거리로도 알려져 있으며, 하나의 시퀀스를 다른 시퀀스로 바꾸기 위해 필요한 삽입, 삭제, 치환 연산의 횟수를 기반으로 유사도를 계산하는 알고리즘이다[21].

$Lev(s_i, s_j)$는 수식 (4)와 같이 주어진다.

$$Lev(s_i, s_j) = \begin{cases} |s_i| & \text{if } |s_j| = 0 \\ |s_j| & \text{if } |s_i| = 0 \\ Lev(tail(s_i), tail(s_j)) & \text{if } s_i[0] = s_j[0] \end{cases}$$
$$otherwise,$$
$$Lev(s_i, s_j) = 1 + \min \begin{cases} Lev(tail(s_i), s_j) \\ Lev(s_i, tail(s_j)) \\ Lev(tail(s_i), tail(s_j)) \end{cases} \tag{4}$$

이때, $tail(s)$는 시퀀스 s의 첫 번째 심볼을 제거하여 생성한 부분 시퀀스를 의미한다.

Levenshtein Distance는 두 시퀀스가 유사할수록 0에 가까운 값을 가진다. 0과 1 사이의 값을 가지면서 두 시퀀스가 유사할수록 1의 가까운 값을 이 되도록 계산식을 수정하면 수식 (5)와 같다.

$$similarity(s_i, s_j) = 1 - \frac{Lev(s_i, s_j)}{\max(|s_i|, |s_j|)} \tag{5}$$

4.2.3.3 Jaccard Similarity

Jaccard Similarity는 두 시퀀스 내에 존재하는 고유한 심볼들 중 공통된 심볼의 비율을 기반으로 유사도를 계산하는 알고리즘이다[22].

$Jaccard(s_i, s_j)$는 수식 (6)과 같이 주어진다.

$$similarity(s_i, s_j) = Jaccard(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|} \tag{6}$$

4.2.3.4 Jaro Similarity

Jaro Similarity는 하나의 시퀀스를 다른 시퀀스로 바꾸기 위해 필요한 전치 연산의 횟수를 기반으로 유사도를 계산하는 알고리즘이다[23].

$Jaro(s_i, s_j)$는 수식 (7)과 같이 주어진다.

Table 2. Dataset information

	Attack technique	Packet capture time (sec)	Number of MSG IDs
Training	Benign	181.43	5418
Test	Benign	122.23	3651
	Wrong end	121.08	3553
	Service request flooding	121.58	3570
	Fuzzing	120.97	3548

$$similarity(s_i, s_j) = Jaro(s_i, s_j) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_i|} + \frac{m}{|s_j|} + \frac{m-t}{m}\right) & otherwise \end{cases} \quad (7)$$

이때, m은 두 시퀀스 간 일치하는 심볼 수를 나타내며, t는 전치 연산 수를 의미한다.

V. 실험 결과

5.1 실험환경 및 데이터셋

본 연구에서는 Pixhawk4 펌웨어가 내장된 비행 컨트롤러에 UAVCAN을 지원하는 변속기 4개를 연결한 쿼드콥터 무인 비행체를 사용하여 제안하는 기법의 성능을 검증하였다. 무인 비행체 내부 네트워크에 악의적인 노드가 존재하는 상황을 구성하기 위해 CAN-BUS shield를 장착한 라즈베리 장비를 무인 비행체 내부 네트워크에 추가로 연결하였다[24]. 본 연구에서 정의한 공격 기법은 Python 라이브러리인 python-can을 이용하여 구현하였으며, 라즈베리 장비를 통해 무인 비행체 내부 네트워크에서 공격을 시뮬레이션하였다. 안전을 위해 무인 비행체의 프로펠러를 제거하고 비행체를 지면에 고정시킨 상태에서 비행체에 시동을 걸고 공격 메시지를 주입하였다. 공격 메시지가 과도하게 발생하여 내부 네트워크의 대역폭이 소진되는 등의 의도치 않은 이상 현상을 방지하기 위해, 정상 상태일 때 발생하는 데이터 양의 10% 수준이 되도록 공격 메시지 주입 빈도를 조절하였다.

Table 2.는 각각의 공격 기법을 시뮬레이션 할 때 수집한 데이터셋의 정보를 보여준다. 모든 데이터셋은 공통적으로 16 종류의 MSG ID들로 이루어져 있다.

5.2 성능 평가

각각의 유사도 계산 알고리즘과 Test 데이터셋에 대해 피쳐 추출부의 window size 값을 조정하며 탐지 성능을 측정하였다. stride 값은 window size와 같은 값을 갖도록 하였다. Fig. 5.는 Test 데이터셋 중 Benign 데이터셋과 Fuzzing 데이터셋에 대해 window size를 조정하며 nLCS 또는 Ja

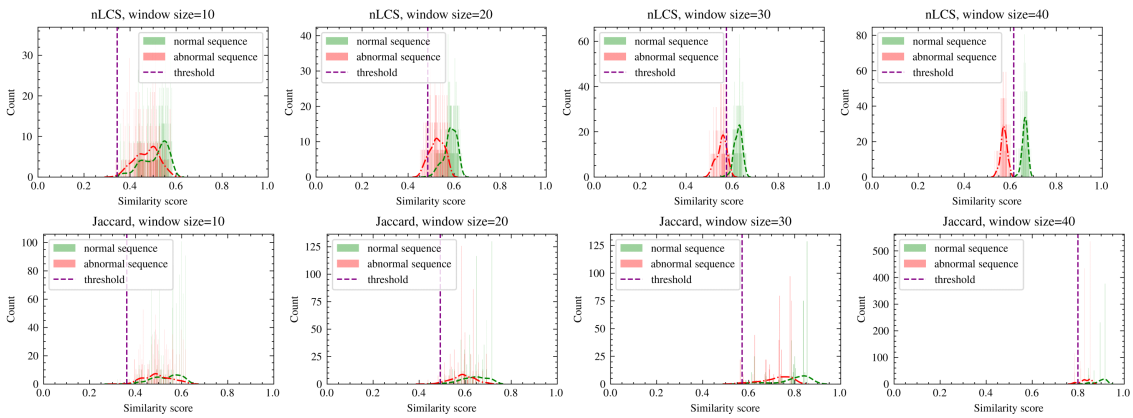

Fig. 5. Distribution anomaly scores calculated by nLCS or Jaccard Similarity per each window size

Table 3. Detection performances per each similarity measure algorithm and window size

window size		10		20		30		40	
Algorithm	**Dataset**	**Acc (%)**	**F1 (%)**	**Acc (%)**	**F1 (%)**	**Acc (%)**	**F1 (%)**	**Acc (%)**	**F1 (%)**
nLCS	Wrong end	50.97	1.12	57.10	22.99	92.05	91.32	**100.0**	**100.0**
	Service request flooding	51.31	2.21	52.5	7.56	74.89	66.29	**96.11**	**95.90**
	Fuzzing	55.21	16.58	57.38	23.88	79.97	74.73	**99.44**	**99.42**
Levenshtein Distance	Wrong end	51.25	2.22	52.92	8.65	92.05	91.24	100.0	100.0
	Service request flooding	51.31	2.77	51.94	5.46	69.03	54.32	77.77	71.10
	Fuzzing	53.54	10.69	55.71	18.46	81.17	76.43	90.50	89.30
Jaccard Similarity	Wrong end	50.83	1.11	51.81	4.41	52.71	8.13	56.98	22.22
	Service request flooding	51.45	3.84	52.77	8.60	55.23	17.05	57.77	25.49
	Fuzzing	54.65	15.10	51.25	2.23	56.06	19.84	56.42	20.40
Jaro Similarity	Wrong end	50.69	1.11	50.69	0.0	69.87	6.00	57.54	24.00
	Service request flooding	50.76	1.66	50.55	0.0	68.20	56.81	55.55	18.36
	Fuzzing	53.12	10.13	51.53	3.33	74.47	68.06	56.98	22.22

ccard Similarity로 계산한 유사도 점수 분포를 보여준다. nLCS의 경우 window size가 증가함에 따라 정상 시퀀스와 비정상 시퀀스의 유사도 점수 분포 간 차이가 극명하게 나타났고, window size가 40일 때 대부분의 정상 시퀀스와 비정상 시퀀스를 구분할 수 있었다. 하지만 Jaccard Similarity의 경우 window size에 관계없이 정상 시퀀스와 비정상 시퀀스를 거의 구분할 수 없었다.

Table 3.은 test 데이터셋에 속한 각각의 데이터셋들에 대해 유사도 계산 알고리즘과 window size에 따른 Accuracy, F1-score 성능을 보여준다. nLCS와 Levenshtein distance은 window size가 증가함에 따라 성능 수치도 증가한 반면, Jaccard Similarity와 Jaro Distance는 window size에 따른 유의미한 성능 변화가 관찰되지 않았다. nLCS를 사용하고 window size를 40으로 설정했을 때 가장 높은 성능을 기록했다.

nLCS와 Levenshtein Distance는 유사도 계산 시 시퀀스 내 심돌들의 순서 정보를 고려한다[20][21]. 반면, Jaccard Similarity와 Jaro Similarity는 심볼들의 등장 여부만을 고려하여 시퀀스들 간 유사도를 계산한다[22][23]. 이와 같은 유사도 계산 알고리즘의 성질과 Table 3.의 실험 결과를 종합해보면, 정상 운행 상태일 때와 비교했을 때 공격이 진행 중일 시 분석 대상 시퀀스 내에 존재하는 MSG ID들의 종류는 변화가 없으나 MSG ID의 순서 정보가 변경된다는 사실을 알 수 있다. 정상 및 비정상 시퀀스 내 MSG ID들의 종류에 차이가 없으므로 Jaccard Similarity 또는 Jaro Similarity를 사용할 시 정상 시퀀스와 비정상 시퀀스를 거의 구분해낼 수 없는 것이다. 즉, 무인 비행체 내부 네트워크에 이상 징후가 발생할 시 무인 비행체가 생성하는 상태 메시지들의 종류에는 큰 변화가 없으나 무인 비행체가 일정한 주기로 상태 메시지를 생성하는 능력이 저하된다는 것을 알 수 있다.

VI. 결 론

본 논문에서는 무인 비행체 내부 네트워크에서 수행 가능한 세 종류의 악의적인 메시지 주입 공격 기법을 정의하고, 해당 공격 기법들이 비행체의 정상 작동을 저해한다는 것을 실제 무인 비행체를 이용한 실험을 통해 입증하였다. 또한 본 논문에서는 MAVLink MSG ID 시퀀스들 간 유사도를 기반으로 무인 비행체의 이상 징후를 탐지하는 시스템을 제안하였다. 잘 알려진 네 종류의 유사도 측정 알고리즘을 이용하여 시퀀스의 유사도 점수를 계산하였고, 정상 시퀀스와 비정상 시퀀스를 효과적으로 분류해냈다. 유사도 측정 알고리즘의 성질을 바탕으로 시퀀스 내 MSG ID들의 순서 정보가 무인 비행체의 정상 및 비정상 여부를 식별하는 중요한 정보임을 확인하였다.

본 연구의 한계점은 다음과 같다. 본 논문에서 사용한 무인 비행체는 내부 네트워크 내에 변속기와 비행 컨트롤러만 존재하는 간단한 형태의 소형 드론이다. 복잡한 내부 네트워크를 갖고 더욱 다양한 종류의 상태 메시지를 생성하는 중대형 무인 비행체를 대상으로 제안하는 이상 탐지 시스템의 효용성을 검증해야한다. 또한, 탐지 성능을 높이기 위해 단순히 w

indow size를 증가시키는 것은 바람직하지 않다. window size가 커질수록 시퀀스를 관측하는 데 걸리는 시간이 길어지고 유사도 계산을 위한 연산량이 늘어나므로 이상 징후에 대한 대응이 늦어지기 때문이다.

향후 복잡한 내부 네트워크를 가진 중대형 무인 비행체를 대상으로 무인 비행체가 상태 메시지를 생성하는 빠르기와 탐지 시스템의 컴퓨팅 능력 등을 고려한 최적의 window size를 결정하는 연구를 진행할 계획이다.

References

[1] Hayat, S., Yanmaz, E., Muzaffar, R., “Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint,” IEEE Communications Surveys Tutorials, 18 (4), pp. 2624 - 2661, 2016.

[2] Meier, L., “Dynamic Robot Architecture for Robust Realtime Computer Vision,” Ph.D. thesis ETH Zurich, 2017.

[3] Uncomplicated application-layer vehicular computing and Networking. Uncomplicated Application-layer Vehicular Computing And Networking, https://uavcan.org/, November 5. 2021.

[4] Wikimedia Foundation. “Iran - U.S. RQ-170 incident”. https://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident, October 19. 2021.

[5] Bozdal, M., Samie, M., & Jennions, I., “A survey on can bus protocol: Attacks, challenges, and potential solutions,” In 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), pp. 201-205, Aug. 2018.

[6] Olufowobi, H., Young, C., Zambreno, J., Bloom, G., “Saiducant: Specificationbased automotive intrusion detection using controller area network (can) timing,” IEEE Transactions on Vehicular Technology, 69(2), pp. 1484 - 1494, 2020.

[7] Sun, H., Lee, S.Y., Joo, K., Jin, H., Lee, D.H., “Catch id if you can: Dynamic id virtualization mechanism for the controller area network,” IEEE Access, 7, pp. 158237 - 158249, 2019.

[8] Tariq, S., Lee, S., Kim, H.K., Woo, S.S., “Can-adf: The controller area network attack detection framework,” Computers & Security, 94, pp. 101857, 2020.

[9] Wang, Q., Lu, Z., Qu, G., “An entropy analysis based intrusion detection system for controller area network in vehicles,” In 2018 31st IEEE International System-on-Chip Conference (SOCC), pp. 90 - 95, 2018.

[10] Birnbaum, Z., Dolgikh, A., Skormin, V., O’Brien, E., Muller, D., “Unmanned aerial vehicle security using recursive parameter estimation,” In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 692 - 702, 2014.

[11] Kim, K.H., Nalluri, S., Kashinath, A., Wang, Y., Mohan, S., Pajic, M., Li, B., “Security analysis against spoofing attacks for distributed uavs,” In: Decentralized IoT Systems and Security, 2020.

[12] Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., Gan, D., “Cloud-based cyber-physical intrusion detection for vehicles using deep learning,” IEEE Access, 6, pp. 3491 - 3508, 2018.

[13] Sedjelmaci, H., Senouci, S.M., Ansari, N., “A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(9), pp. 1594 - 1606, 2018.

[14] Meng, L., Ren, S., Tang, G., Yang, C., & Yang, W.,“ Uav sensor spoofing detection algorithm based on gps and optical flow fusion,” In Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, pp. 146-151, Jan. 2020.

[15] Kapoor, P., Vora, A., Kang, K.D., “Detecting and mitigating spoofing attack against an automotive radar,” In: 2018 IEEE 88th Vehicular

Technology Conference (VTCFall). pp. 1-6, 2018.
[16] Panice, G., Luongo, S., Gigante, G., Pascarella, D., Di Benedetto, C., Vozella, A., Pescap'e, A., "A svm-based detection approach for gps spoofing attacks to uav," In: 2017 23rd International Conference on Automation and Computing (ICAC), pp. 1-11, 2017.
[17] Ebeid, E., Skriver, M., & Jin, J., "A survey on open-source flight control platforms of unmanned aerial vehicle," In 2017 Euromicro Conference on Digital System Design (DSD), pp. 396-402, Aug. 2017.
[18] Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., & Khalgui, M., "Micro air vehicle link (mavlink) in a nutshell: A survey," IEEE Access, 7, pp. 87658-87680, 2019.
[19] Gomaa, W. H., & Fahmy, A. A., "A survey of text similarity approaches," international journal of Computer Applications, 68(13), pp. 13-18, 2013.
[20] Chandola, V., Banerjee, A., & Kumar, V., "Anomaly detection for discrete sequences: A survey," IEEE transactions on knowledge and data engineering, 24(5), pp. 823-839, 2010.
[21] Levenshtein, V. I., "Binary codes capable of correcting deletions, insertions, and reversals," In Soviet physics doklady, 10(8), pp. 707-710, Feb. 1966.
[22] Jaccard, P. "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," Bull Soc Vaudoise Sci Nat, 37, pp. 547-579, 1901.
[23] Jaro, M. A., "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida," Journal of the American Statistical Association, 84(406), pp. 414-420, 1989.
[24] Pant, S., & Lee, S., "Design and Implementation of a CAN Data Analysis Test Bench based on Raspberry Pi," Journal of Multimedia Information System, 6(4), pp. 239-244, 2019.

〈저자소개〉

서 강 욱 (Kang Uk Seo) 학생회원
2020년 2월: 고려대학교 컴퓨터학과 졸업
2020년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 데이터 중심 보안, 네트워크 보안, 이상 탐지 시스템

김 휘 강 (Huy Kang Kim) 종신회원
1998년 2월: KAIST 산업경영학과 학사
2000년 2월: KAIST 산업공학과 석사
2009년 2월: KAIST 산업 및 시스템공학과 박사
2004년 5월~2010년 2월: 엔씨소프트 정보보안실장. Technical Director
2010년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 온라인 게임 보안, 자동차 보안, 침입탐지시스템, 네트워크 보안