MUVIDS: False MAVLink Injection Attack Detection in Communication for Unmanned Vehicles

Seonghoon Jeong, Eunji Park, Kang Uk Seo, Jeong Do Yoo, and Huy Kang Kim
School of Cybersecurity, Korea University
{seonghoon, epark911, tjrkddnr, opteryx25104, cenda}@korea.ac.kr

***Abstract*—MAVLink protocol is a *de facto* standard protocol used to communicate between unmanned vehicle and ground control system (GCS). Given the nature of the system, unmanned vehicles use MAVLink to communicate with a GCS to be monitored and controlled. Such communication continues to grow on the Internet due to its rapidly grown nature. In the past few years, the unmanned vehicle security has been one of the key research topics in the security field. However, existing research has mainly focused on the sensor- and GPS-based attack detection methods. To this end, we propose MUVIDS, a network-level intrusion detection system to protect MAVLink-enabled unmanned vehicles managed by GCS over the Internet. MUVIDS includes two Long short-term memory models that leverage a sequential MAVLink stream from a victim vehicle. The two models are designed to solve a binary classification problem (in case of labels are available) and a next MAVLink message prediction problem (in case of no label is available), respectively. The experiment was performed on a software-in-the-loop unmanned aerial vehicle (UAV) simulator and a hardware-in-the-loop UAV simulator. The experiment result confirms that MUVIDS detects false MAVLink injection attacks effectively.**

I. Introduction

Unmanned vehicles, specifically known as unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), are getting special attention because of shifting paradigms in commercial delivery, military operations, social protection such as rescue, disaster monitoring [4], and even keeping our health in the middle of COVID-19 pandemic [3]. Among unmanned vehicles, commercial drones are the most promising systems so far in which the drones are easy to be deployed due to their low dependence on ground infrastructure during flight. Given the nature of the system, UAVs communicate with a ground control system (GCS) using satellite links, radio frequency, Wi-Fi, LTE, 5G, and so on, to be monitored and controlled. Today, UAV-GCS communication continues to grow on the Internet due to its rapidly grown nature. Thus, the importance of ensuring secure UAV-GCS communications is highlighted.

While unmanned vehicles provide many positive effects on our lives, attacks against the vehicles lead to malfunction and fatal consequences for operators (lost vehicles) and the physical world (casualties). Some studies already showed the possibility of attacks targeting unmanned vehicles [8], [12]. Therefore, it is important to research methods for detecting abnormalities or intrusions on unmanned vehicles. Each unmanned vehicle has various external I/O interfaces, which could open up attack surfaces, notably inertial measurement units, global positioning system, and network interfaces (e.g., Wi-Fi card). However, existing research has mainly focused on the sensor- and GPS-based attack detection methods. Such research detects abnormal signs and intrusions by predicting the next sensor value and location information based on a given mission and the current driving context [11]. A drawback of proposed techniques is that sensor and GPS data highly rely on the operation environment and some parameters need to be tuned dependent on a specific vehicle. On the other hand, in the context of network security studies, the majority is in vehicular ad-hoc networks (VANETs). The main goal of such a study is to keep availability and reliability of wireless communication, as well as providing secure "wireless" communication channels.

Despite considerable efforts to detect intrusions for UAVs, however, no study has been conducted on intrusion detection methods in the field of UAV-GCS communication. Although sensor- or GPS-based anomaly detection methods could identify network-level intrusions because the UAV will not operate as expected, such previous methods will alert only after being compromised. While a communication protocol for unmanned systems and ground stations is getting used to major autopilot systems [7], some vulnerabilities were discussed by previous work. Kwon *et al.* [8] discover a vulnerability of MAVLink used for UAV-GCS communication. By exploiting the vulnerability (injecting a false command packet), they succeeded in hovering the drone against the pre-uploaded mission. In addition, through the fuzz test of the protocol, several vulnerabilities were discovered that could cause the autopilot system to crash. [5]. In summary, the studies imply that the target UAV can be exploited by the attacker if an attacker can inject false packets.

In this paper, we propose MUVIDS, a network-level intrusion detection system (IDS) to protect MAVLink-enabled unmanned vehicles managed by GCS over the Internet. We follow the previous literature [5], [8] that assumes an attacker injects false MAVLink messages to a target UAV without recognition of the connected GCS. Our goal is to let the GCS identify whether a connected unmanned vehicle is under attack or not based on a pattern of network traffic. For the experiment, we set up a network testbed. A GCS is connected to a software-in-the-loop (SITL) UAV simulator and a hardware-in-the-loop (HITL) UAV simulator. For the connection, MAVLink[1] is used, which is the *de facto* standard protocol designed to exchange

[1]https://github.com/mavlink/mavlink, https://mavlink.io/en/

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2021
25 February 2021, Virtual
ISBN 1-891562-68-1
https://dx.doi.org/10.14722/autosec.2021.23036
www.ndss-symposium.org

STX (0xFE)	LEN	INC FLAGS	CMP FLAGS	SEQ	SYS ID	COMP ID	MSG ID 3 bytes	PAYLOAD 0–255 bytes	CHECKSUM 2 bytes	SIGNATURE 13 bytes, *optional*

Fig. 1. MAVLink 2.0 frame header. Each field contains a single byte unless noted otherwise.

messages between vehicle and GCS over various network media. At the GCS, MUVIDS captures inbound MAVLink packets from the vehicle and detects injection attacks using LSTM-based recurrent neural networks. Though we have labels for training our models, we also provide another use case of MUVIDS in a case of lack of the label information.

Our contributions can be summarized as follows:

- We propose MUVIDS to identify attacks on unmanned vehicles using MAVLink communication between unmanned vehicle and GCS.
- We extend false MAVLink injection attacks from previous works [8], [12] to evaluate MUVIDS.
- By using deep learning-based detection models, MUVIDS successfully detects three types of false MAVLink injection attacks in both a SITL and a HITL environment regardless of availability of label. That is a huge benefit for unmanned vehicle operators because they do not have to throw vehicles in the wild to collect train set.
- MUVIDS works for monitoring various MAVLink-enabled vehicles easily since it does not require any parameters regarding vehicles.

II. Background

A. MAVLink

MAVLink stands for Micro Air Vehicle Link. The goal of MAVLink is to provide efficient and reliable communication and covers various common functionality used at most GCSs, autopilot systems of unmanned vehicles. Unlike its name, MAVLink supports not only *air* vehicles but also *ground/underwater* vehicles. Thanks to the open-source policy and hard-working team Dronecode underneath the Linux Foundation to support many programming languages, MAVLink is now one of the most commonly adopted protocols in personal and commercial vehicles (especially drones). Currently, MAVLink 2.0 is the up-to-date protocol.

Fig. 1 shows the MAVLink 2.0 protocol header. We here introduce packet fields briefly. Since MAVLink can be used on various channels, the *STX* field implies that the next buffer is a MAVLink 2.0 message. The *LEN* field means the length of the *PAYLOAD* field. The *SEQ* field contains a number whose value increases at each transmission. The *SYS/COMP ID* points to a target vehicle and an in-vehicle component, respectively. The *MSG ID*, which we use in this paper, indicates how the *PAYLOAD* field should be parsed and consumed. Two flag fields and the *SIGNATURE* field are designed for message signing. However, message signing is not always available in all implementations, unlike essential communication features. Note that there is no security mechanisms like encryption and authentication. To this end, some security solutions are discussed in the literature. For the technical details and following state-of-the-art security solutions about the MAVLink protocol, the reader is kindly referred to [7].

B. Communication between UAV-GCS

The MAVLink connection is established through the following brief explanations of the steps: A vehicle starts to transmit a HEARTBEAT message in the broadcast domain or to a designated address of GCS. At the same time, the GCS listens to the HEARTBEAT message on the ground. When the GCS discovers the vehicle, then both systems continuously send a HEARTBEAT message to each other every second to keep it alive. Then, the connection ends with some silence of HEARTBEAT messages.

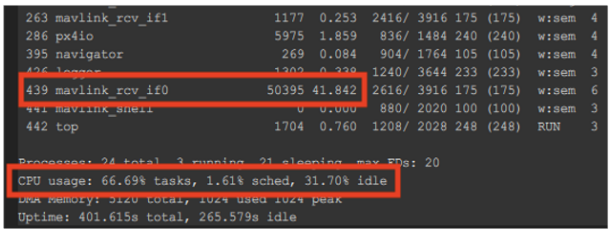
After a connection is established, the vehicle continually reports the current status toward the connection so that the GCS can follow the vehicle's status. The frequency of reports may vary depending on vehicle types, autopilot systems, link bandwidth, etc. On the other hand, the GCS sends MAVLink messages to the connected vehicle when triggered by humans (e.g., uploading mission, sending a new command), instead of a periodic transmission. The MAVLink message parser does not refer to any value in other protocol headers, such as source IP address. Thus, MAVLink communication is vulnerable to a false message injection if the message has a valid payload.

C. False MAVLink injection attack

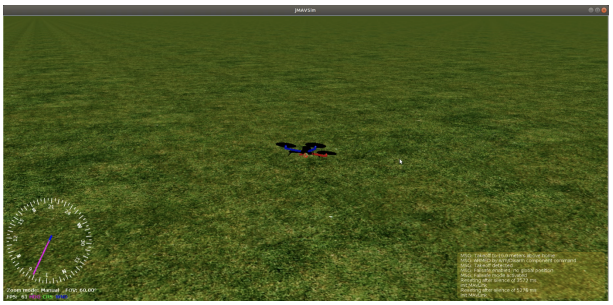
Here, we introduce false MAVLink injection attacks, which we want to detect using MUVIDS. The attacks are more critical than sensor- or GPS-level attacks in terms that the attacks can be performed by an far remote attacker even she/he does not have any specially designed hardware. We assume the ability of the attacker, the GCS, the UAV as follows: (1) The attacker can access the target UAV through the compromised network. (2) The GCS does not become aware of the attacker's existence or false packets generated by the attacker. (3) The UAV and the GCS use plain text (no encryption) by following a standard implementation. (4) The attacker interferes with an established session between UAV-GCS because the target UAV only listens to an established session.

The attacker's goal is to disguise an accident as occurring on UAVs. Here, we consider flooding attacks to occupy the drone's resources and reduce its capacity, such as limited action radius by battery drain, insufficient object perception. Based on the assumption, we present three types of attacks as follows:

- **Heartbeat flooding.** The message is consumed by the MAVLink receiver process and does not pass across the inside of the autopilot system.
- **Ping flooding.** The ping message is designed for networks that do not support ICMP. The receiver needs to send a ping response to the opposite system.
- **Request flooding.** The request message is used to query the value of a specific parameter. The receiver sends the opposite system the requested parameter. It consumes much more computing power because of additional kernel tasks.



(a) Process list in NuttX OS. The MAVLink message receiver over-consumes the CPU in our Pixhawk 4 autopilot system.



(b) Simulated UAV crash due to the considered flooding attack

Fig. 2. False MAVLink injection attack realized in our HITL simulator

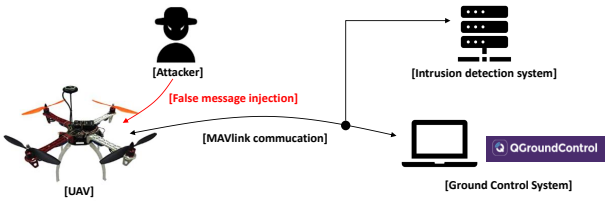


Fig. 3. Architectural overview of MUVIDS. The attacker inject false messages to the UAV whereas the IDS receives MAVLink communication from the UAV.

Fig. 2 shows the impact of the attack on our HITL simulator. The flooding attacks make the Arm processor of our Pixhawk 4 busy. The more messages an attacker sends per time, the busier the CPU becomes. As a result, the victim will consume much more battery power. Due to the bandwidth limitation in our Pixhawk 4 device, it seems the attacks does not overwhelm the system. However, the flooding may lead to blocking messages from GCS. Furthermore, when the target drone starts autonomous driving, the CPU utilization rises nearly 100%, and the drone falls in the simulation environment. In the meantime, the GCS receives usual status reporting messages and a large number of awkward ping/parameter responses.

III. MUVIDS: IDS FOR MAVLINK-ENABLED UNMANNED VEHICLES

We focus on the communication characteristics of UAVs that periodically report current driving situations. We assume that the periodic reports of a UAV will be ruined when the UAV is struggling with flooding attacks. In other words, we suppose the inherent sequence pattern of MAVLink messages will change. Consequently, we try to detect intrusion using *MSG ID* sequences from the MAVLink stream.

Our method is intended to recognize the network-level attacks toward UAVs effectively. The use case of the proposed

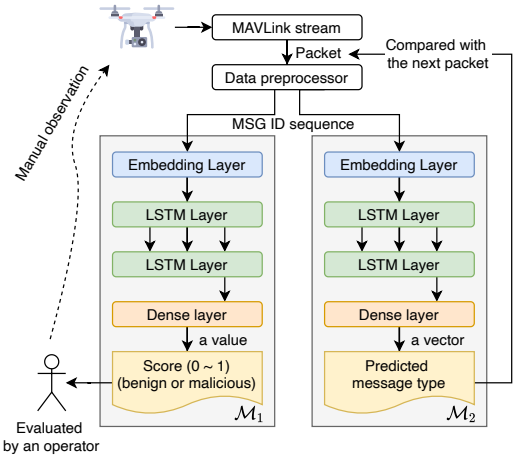


Fig. 4. MUVIDS scheme representing two models $\mathcal{M}_1$ and $\mathcal{M}_2$, and the model evaluation approaches

methodology against the attacker is shown in Fig. 3. The IDS is installed along with the GCS. The IDS takes every packet that comes from the UAV. Then, the IDS parses the value of the *MSG ID* field in any MAVLink messages and enumerates the sequence of observed values in the order of arrival. The sequence is then sliced into fixed-size subsequences by the sliding window method. Finally, the subsequences are fed to an intrusion detection model. To build the intrusion detection model, we implement recurrent neural networks using long short-term memory (LSTM) cells. The LSTM cell is commonly adopted in intrusion detection model exploiting sequential data where the output correlates a sequence of inputs.

To respond to various situations, we design two types of detection models. The first model can be chosen when the IDS can use both an attack-free dataset and an intrusion dataset in a training phase. For training the model, we design the first model to solve the binary classification problem that judges whether the remote UAV is under attack at a specific time or not. Broadly speaking, the more intrusive cases the train set contains, the more accurate estimates the deep learning model can infer. On the contrary, if the IDS has not been faced an attack yet or is not aware of attack models, the first model should work for nothing. Such a case is a common problem in the area of intrusion detection studies. To this end, we design a second model that can be trained with only an attack-free dataset. Fig. 4 shows the proposed intrusion detection scheme, which is composed of the data preprocessing step and the classification/prediction step as follows:

Data preprocessing. The data preprocessor receives inbound MAVLink messages stream at the GCS and outputs a $w \times 4$ size of an array into the deep learning model, where w is the window size, and the value 4 means the size of the integer. At the beginning of the procedure, w-sized queue is initialized. On the arrival of every MAVLink message, the preprocessor cast the three bytes of value in the *MSG ID* field to the four bytes Int32 type, then puts the value in the queue. If the queue is not full after queueing, then wait for the next message. Otherwise, if the queue is full, the preprocessor yields the queue's entire values, then dequeues the oldest value. The

procedure continues until the input stream is closed.

Intrusion detection model $\mathcal{M}_1$. The $\mathcal{M}_1$ is designed to solve a binary classification problem of whether a UAV generates the given subsequence under attack-free or under attack. To this end, a supervised intrusion detection model is built using a deep neural network. Specifically, the model is composed of an embedding layer, followed by two LSTM layers and a single dense layer. The embedding layer is used to convert an integer array (i.e., a subsequence of MAVLink *MSG ID*s) into a fixed-size vector for the LSTM layers. Two LSTM layers then extract features from given sequential data. As depicted in Fig. 4, the first LSTM layer returns a full sequence, whereas the second LSTM layer returns only the last output. The dense layer returns the detection result for each subsequence in a single floating point value of between 0 to 1 activated by a sigmoid function. To train the model, the operator should prepare benign and intrusive MAVLink stream and labels describing the situation (i.e., benign or under attack) of the given stream. The stochastic gradient descent algorithm is used to optimize the weights and biases of the model.

Message prediction model $\mathcal{M}_2$. In case of lack of label information, we further implement a predictive model that calculates the most promising value of the next *MSG ID*, instead of binary status, with a given subsequence of previous *MSG ID*s. Basically, the model shares the architecture of $\mathcal{M}_1$ that we discussed above, except the dense layer. The dense layer in $\mathcal{M}_2$ returns a vector representing possibilities of appearing MAVLink *MSG ID*. The Adam optimizer algorithm is used for the optimization of the model. In the training phase, only a "benign" MAVLink stream is required to allow the model to learn the next *MSG ID*. We expect that the trained $\mathcal{M}_2$ shows the poor prediction result when the input comes from an UAV under attack.

While incoming MAVLink messages contain operation information such as attitude and location of a vehicle, we design MUVIDS not to use them because using such data consumes much effort from a UAV operator for feature engineering, parameter tuning, by a specific vehicle model.

IV. Experiment Result

This section provides our packet generation environment, hyper-parameters for the proposed model, and the performance evaluation result. Our proposed method is tested by three common evaluation metrics: Precision, Recall, and F1-score.

Experiment environment. We deploy a network topology created with three virtual machine instances (i.e., GCS, SITL/HITL UAV simulator, an attacker) in a VMware ESXi hypervisor. Both SITL and HITL simulators are specially designed to replicates a real-world drone, including MAVLink 2.0 communication system. The only difference is that an autopilot system hardware (Pixhawk 4 in our experiment) operates actuators and executes MAVLink commands, whereas only a software-based autopilot system operates the SITL simulator. We use QGroundControl as the GCS to communicate with the UAV. Also, we implement three attack models (described in §II-C) using a Python library, pymavlink, that floods MAVLink messages to the UAV simulators. We implement $\mathcal{M}_1$ and $\mathcal{M}_2$ using Keras on a GeForce RTX 2080 installed PC. We set $w = 128$, the input/output dimension of embedding layer to 512/32, the units of each LSTM layer to 128, and the dropout ratio to 0.3. We set the learning rate to 10^{-3} and 10^{-4} for $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively.

TABLE I. Performance evaluation using intrusion dataset

Simulator	Attack type	Epochs	Precision	Recall	F1-score
SITL	Heartbeat flooding	93	0.98	0.97	0.98
	Ping flooding	1	1.00	1.00	1.00
	Request flooding	1	1.00	1.00	1.00
HITL	Heartbeat flooding	2	1.00	1.00	1.00
	Ping flooding	1	1.00	1.00	1.00
	Request flooding	1	1.00	1.00	1.00

Dataset. At the GCS, we collect four separated packet captures in the following situations: (1) attack-free, (2) heartbeat flooding, (3) ping flooding, and (4) request flooding. Consequently, we have eight datasets in total because we have two UAV simulators. The GCS and the UAV use UDP for MAVLink communication. We then generate a set of *MSG ID* subsequences using our data preprocessor. Note that each packet capture contains packets generated by an UAV and does not contain any packets generated by the attacker.

$\mathcal{M}_1$ evaluation. To train $\mathcal{M}_1$, we label the attack-free set to 0 and another set of each attack to 1, concatenate them, shuffle them, and split them into 80% of the train set and 20% of the validation set. The result of a performance evaluation using the validation set is given in Table I. The epochs column specifies when the model shows its best intrusion detection performance for the validation set. The experiment result shows that $\mathcal{M}_1$ provides satisfactory intrusion detection result. Interestingly, ping flooding and request flooding are completely identified by the proposed method. The subsequence is mostly dominated by response messages triggered as many as attacker requests, whereas such responses rarely appear in attack-free states. The heartbeat flooding is hard to be identified because it does not trigger any out-of-context messages from the target UAV to the GCS. Still, we confirm that $\mathcal{M}_1$ can accurately detect the heartbeat flooding after being trained with one more epoch. Note that the SITL is more robust to flooding with respect to computational power; a server CPU operates the autopilot firmware instead of a low-power embedded device. We find that some cases of heartbeat flooding are ambiguous to be identified in the SITL simulator, even after over 90 epochs of training. We achieve the best F1-score of 0.98 at 93 epochs.

$\mathcal{M}_2$ evaluation. We use the attack-free set to train $\mathcal{M}_2$. We randomly choose 20% of the attack-free dataset as the validation set to measure the prediction ability of trained $\mathcal{M}_2$. Table II provides the prediction evaluation of $\mathcal{M}_2$. For the reader's understanding, we use a descriptive name of the type instead of the *MSG ID* value. The empty count means no such message type found in the corresponding simulation environment. In summary, we achieve a F1-score of 0.96 in both simulators, which means that the $\mathcal{M}_2$ almost completely predicts which message will arrive at the GCS next time based on their given context MAVLink messages. However, it turns out that the $\mathcal{M}_2$ does not predict all message types well due to the nature of the packet generation strategy running on a UAV autopilot system. Specifically, we find that sporadic messages and event-driven messages were not correctly predicted by $\mathcal{M}_2$, while $\mathcal{M}_2$ predicts high-frequent low-jitter *MSG ID*s, e.g., attitude, location. Indeed this can be treated as a drawback of the proposed method since our model is not designed to

TABLE II. F1-SCORE FOR PREDICTION EVALUATION

MAVLink message type (MSG ID)	SITL		HITL	
	F1-score	Count	F1-score	Count
HEARTBEAT	0.96	126	0.76	111
SYS_STATUS	0.95	126	1.00	111
SYSTEM_TIME	–	–	0.96	110
PING	0.00	12	0.70	111
GPS_RAW_INT	0.97	126	0.90	2130
SCALED_IMU	–	–	0.97	2769
ATTITUDE	0.97	6302	1.00	5538
ATTITUDE_QUATERNION	0.99	6302	0.98	5538
LOCAL_POSITION_NED	0.98	6302	0.92	3323
GLOBAL_POSITION_INT	0.03	727	0.99	1108
SERVO_OUTPUT_RAW	0.99	6302	1.00	2215
MISSION_CURRENT	–	–	0.78	1047
VFR_HUD	0.93	504	0.94	2215
ATTITUDE_TARGET	0.98	6302	–	–
POSITION_TARGET_LOCAL_NED	0.99	6302	–	–
POSITION_TARGET_GLOBAL_INT	–	–	1.00	1108
HIGHRES_IMU	–	–	0.97	5538
TIMESYNC	–	–	1.00	1108
ACTUATOR_CONTROL_TARGET	–	–	0.99	3323
ALTITUDE	1.00	126	1.00	1107
BATTERY_STATUS	0.36	63	0.00	55
ESTIMATOR_STATUS	1.00	63	1.00	554
VIBRATION	1.00	12	1.00	554
HOME_POSITION	0.00	63	0.00	55
EXTENDED_SYS_STATE	0.75	126	0.78	221
UTM_GLOBAL_POSITION	0.06	63	–	–
F1-score, total sample count	0.96	39949	0.96	39949

understand a complete operational context of UAV.

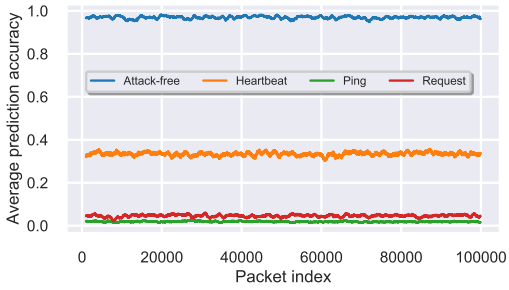
Next, we demonstrate detecting false MAVLink injection attacks using pre-trained $\mathcal{M}_2$ in SITL/HITL simulators. For the experiment, datasets are fed to the $\mathcal{M}_2$ in order of time. As discussed above, the prediction accuracy of the attack-free dataset is expected to be very high in an obvious manner. Meanwhile, the accuracy should be low when a victim UAV is under attack because the attack causes a concept drift of message sequences. We can identify the intrusion by applying a threshold to the average prediction accuracy.

In Fig. 5, we plot the 1000-messages moving average of accuracy over a series of MAVLink messages. We can see the significant differences between the attack-free dataset and the others. It proves the usefulness of $\mathcal{M}_2$ trained with only attack-free dataset. The gap between the heartbeat dataset and the attack-free dataset is interesting since the two datasets consist of a set of the same *MSG ID*s. The same applies to the heartbeat dataset of the HITL simulator; however, it fluctuates from time to time. It implies that our Pixhawk 4 hardware and the NuttX OS therein are lagging due to the attack. Notably, the fluctuation is not extreme. Thus, we can identify the three attacks by setting a detection threshold of 0.9 in a heuristic manner.

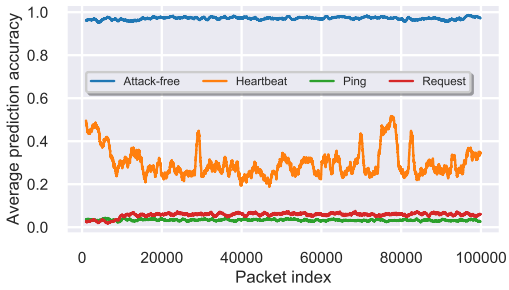
V. RELATED WORK

In the past few years, UAV security has been one of the key research topics in the security field. The directions of research are heading two purposes: vulnerability analysis and intrusion detection.

Vulnerability analysis. The vulnerability analysis research analyzes the UAV security characteristics and designs potential possible attacks. Kwon *et al.* [8] investigated network vulnerabilities using the MAVLink protocol. As the MAVLink communication protocol uses no message encryption, the researchers insisted ICMP flooding attacks are possible. Besides,



(a) Result from the SITL UAV simulator



(b) Result from the HITL UAV simulator

Fig. 5. Average accuracy of MAVLink *MSG ID* prediction over a continuous series of MAVLink traffic

packet injection attacks are possible, considering the protocol as a Waypoint protocol. A study conducted by Rani *et al.* [12] examined the Wi-Fi protocol's vulnerabilities for analyzing UAV's security. Using one of the known WEP vulnerabilities, they demonstrated an attack that forces UAVs to be unauthenticated. Rodday *et al.* [13] offered a packet broadcasting attack adding to the research of Rani and others. The attack collected initial vector values by sending random packets to targets and exploited a WEP vulnerability using the vectors.

Rule-based approach. Another field of study focused on attack detection beyond vulnerability analysis. There are two approaches to detect attacks. First, we introduce researches that detect attacks by setting rules or in a heuristic manner. Sedjelmaci *et al.* [15] designed a rule-based IDS to detect network attacks, such as GPS spoofing, jamming, and false information injection. The researchers proposed algorithms to track network traffic and detect attacks by abnormal traffic changes. The research by Birbaum *et al.* [2] estimated parameters to detect attacks on UAV. Their proposed method predicted the parameters for UAV's operational conditions, such as system parameters and control parameters, and it identified attacks through differences from confidence intervals implying no attacks. They conducted the experiments in a simulated environment using the MAVLink protocol and extracted sensors and actuators data. Muniraj *et al.* [10] investigated which sensors were susceptible to damage by which attacks and defined the information as an attack signature. With safe sensors getting no damage from any attack, the detection system disclosed attacks based on other sensors' status.

Machine learning approach. Some other works proposed machine learning- or deep learning-based approaches to detect intrusion. Sedjelmaci [14] proposed an IDS containing

a security game framework using the Bayesian game model. The framework monitored the network to catch abnormalities and ejected attackers in the case of attacks. Lukas *et al.* [9] researched on attack detection model using deep learning. Their research focused on network attacks such as a denial of service, command injection, and malicious code injection, and the proposed model combined multilayer perceptron (MLP) and LSTM. Arthur [1] designed a lightweight IDS that could be applied for UAVs using sensor data. With 27 kinds of sensor data collected in the simulated environment, the IDS was composed of self-taught learning and a multi-class support vector machine (SVM). Additional research that used the SVM model was a state estimation analysis done by Panice *et al.* [11]. The research was directed to estimate the UAVs' state, and it focused on GPS spoofing with other various attacks. Shoufan [16] studied to establish the operators' signature. The research used UAV's sensor data to build operators' behavior patterns and classify pre-defined operators based on their signatures. The classification experiment utilized a random forest classifier, and it demonstrated the significant performance of the identification. Kim *et al.* [6] used the MLP model to detect attacks and correlate them with a generative adversarial network to examine its effects. They composed a SITL simulator using MAVLink protocol and extracted sensor data for the experiment.

In other domains, several researchers decided to use sensor and actuator data from UAV for security purposes. The data gives us more intuitive views of the current state. Despite the advantage, sensors are too sensitive to the drone's operation environment, and actuators can be too many to consider in the lightweight system. For these reasons, we used the data extracted from the MAVLink protocol. It covers both sensor/actuator data and network communication. The protocol is specialized to the UAV and other vehicles to represent the UAV operation system.

VI. Conclusion

This paper proposes a novel IDS, MUVIDS, to detect false MAVLink injection attacks toward unmanned vehicles. To the best of our knowledge, this study is the first intrusion detection method analyzing MAVLink communication. Based on the experiment result, MUVIDS gives the following benefits to unmanned vehicle operators. First, the experiment result shows that MUVIDS can detect the three types of false MAVLink injection attacks effectively, even without label information. Second, MUVIDS works for various MAVLink-enabled vehicles. We confirm that by using both SITL and HITL UAV simulators. Third, MUVIDS does not require any vehicle- and environment-specific parameters. Fourth, the operator can implement MUVIDS at the GCS. Consequently, the operator does not have to risk failure while adding components to the vehicle directly.

The limitation of our work is that only a few flooding attacks are considered to validate MUVIDS. It is due to the lack of prior systematic study of attacks on MAVLink-enabled vehicles. At this moment, only three types of false MAVLink injection attacks in which the actual impact (e.g., resource consumption and vehicle crash) occurs in our simulators were used in the study. In future work, we will extend this work with various attacks. Also, we will use MAVLink payloads that contain sensor/actuator/GPS information to enhance intrusion detection performance on real driving/flying vehicles.

Acknowledgment

This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00374, Development of Security Primitives for Unmanned Vehicles).

References

[1] M. P. Arthur, "Detecting signal spoofing and jamming attacks in UAV networks using a lightweight IDS," in *Proceedings of Computer, Information and Telecommunication Systems (CITS)*, 2019.

[2] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, D. Muller, and C. Stracquodaine, "Unmanned aerial vehicle security using recursive parameter estimation," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 107–120, 2016.

[3] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90 225–90 265, 2020.

[4] G. Choudhary, V. Sharma, I. You, K. Yim, I.-R. Chen, and J.-H. Cho, "Intrusion detection systems for networked unmanned aerial vehicles: A survey," in *Proceedings of International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 560–565.

[5] K. Domin, "Security analysis of the drone communication protocol: Fuzzing the MAVLink protocol," Master's thesis, KU Leuven, 2016.

[6] K. H. Kim, S. Nalluri, A. Kashinath, Y. Wang, S. Mohan, M. Pajic, and B. Li, "Security analysis against spoofing attacks for distributed UAVs," in *Proceedings of Decentralized IoT Systems and Security (DISS)*, 2020.

[7] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, 2019.

[8] Y.-M. Kwon, J. Yu, B.-M. Cho, Y. Eun, and K.-J. Park, "Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 43 203–43 212, 2018.

[9] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2017.

[10] D. Muniraj and M. Farhood, "A framework for detection of sensor attacks on small unmanned aircraft systems," in *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

[11] G. Panice, S. Luongo, G. Gigante, D. Pascarella, C. Di Benedetto, A. Vozella, and A. Pescapè, "A svm-based detection approach for GPS spoofing attacks to UAV," in *Proceedings of International Conference on Automation and Computing (ICAC)*, 2017.

[12] C. Rani, H. Modares, R. Sriram, D. Mikulski, and F. L. Lewis, "Security of unmanned aerial vehicle systems against cyber-physical attacks," *The Journal of Defense Modeling and Simulation*, vol. 13, no. 3, pp. 331–342, 2016.

[13] N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *Proceedings of Network Operations and Management Symposium*, 2016, pp. 993–994.

[14] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Intrusion detection and ejection framework against lethal attacks in UAV-aided networks: A bayesian game-theoretic methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1143–1153, 2016.

[15] ——, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1594–1606, 2017.

[16] A. Shoufan, "Continuous authentication of UAV flight command data using behaviometrics," in *Proceedings of Very Large Scale Integration (VLSI-SoC)*, 2017, pp. 1–6.