Drone Pilot Identification by Classifying Radio-Control Signals

Abdulhadi Shoufan, Haitham M. Al-Angari, Muhammad Faraz Afzal Sheikh, and Ernesto Damiani

***Abstract*— Analysis of interactions with remotely controlled devices has been used to detect the onset of hijacking attacks, as well as for forensics analysis, e.g., to identify the human controller. Its effectiveness is known to depend on the remote device type as well as on the properties of the remote control signal. This paper shows that the radio control signal sent to an unmanned aerial vehicle (UAV) using a typical transmitter can be captured and analyzed to identify the controlling pilot using machine learning techniques. Twenty trained pilots have been asked to fly a high-end research drone through three different trajectories. Control data have been collected and used to train multiple classifiers. Best performance has been achieved by a random forest classifier that achieved accuracy around 90% using simple time-domain features. Extensive tests have shown that the classification accuracy depends on the flight trajectory and that the pitch, roll, yaw, and thrust control signals show different levels of significance for pilot identification. This result paves the way to a number of security and forensics applications, including continuous identification of UAV pilots to mitigate the risk of hijacking.**

***Index Terms*— Pilot identification, behavioral biometrics, unmanned aerial vehicles, random forest.**

I. Introduction

AS THE panoply of mobile devices has become more numerous and diverse, user-device communication and interaction channels have been increasingly targeted by masquerade attacks [1], where attackers trick a target device into believing they are someone they are not (usually, the legitimate owner). A particularly dangerous example of masquerade attack occurs when an attacker hijacks a remotely controlled device, as this may lead to the attacker forcing the device to self-destroy or harm others. In the simplest case, the onset of a masquerade attack occurs when the attacker starts sending commands to somebody else's device over an open channel, or after having stolen the legitimate user's credentials. In this case, this attack requires virtually no technical expertise to carry out.

Manuscript received July 29, 2017; revised December 2, 2017; accepted February 27, 2018. Date of publication March 23, 2018; date of current version May 9, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hafiz Malik. *(Corresponding author: Abdulhadi Shoufan.)*

A. Shoufan and E. Damiani are with the Information Security Center, Khalifa University, Abu Dhabi 127788, UAE (e-mail: abdulhadi.shoufan@kustar.ac.ae; ernesto.damiani@kustar.ac.ae).

H. M. Al-Angari is with the Biomedical Engineering Department, Khalifa University, Abu Dhabi 127788, UAE (e-mail: haitham.alangari@kustar.ac.ae).

M. F. A. Sheikh is with the Electrical and Computer Engineering Department, New York University, Abu Dhabi 129188, UAE (e-mail: faraz.afzal@nyu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2819126

Countermeasures to masquerade attacks have been studied since the Eighties. In his seminal paper Denning outlined a framework for a real-time system to detect masquerade attacks by comparing the commands issued by the user with previously stored profile of how that user has typically behaved in previous interactions [2]. Along the years, the idea of using command sequence analysis to identify unexpected behavior was put forward in further studies and systems, sometimes with remarkable success [3], [4]. However, these early techniques were designed for devices connected to fixed information and communication technology infrastructures and did not apply well to mobile cyber-physical environments.

Let us consider a thief who takes hold of another person's device. If the device includes a classic masquerade detection tool, it will periodically (e.g., at each device restart) record how the current user is manipulating it and compare the observed behavior to a previously generated profile of the legitimate owner, until a decision can be reached and an alert can be sent. This scenario relies on a threefold assumption: (i) there will be a single moment (the device theft) in which interaction with the thief will replace the interaction with the legitimate user (ii) the device thief will behave in a manner that is clearly inconsistent with the legitimate owner's previously observed behavior (iii) analysis can go on at leisure while the thief is using the device, without strict deadlines to produce a result.

These assumptions may look reasonable for hand-held devices,[1] but are debatable for remotely controlled devices. For this reson, later attempts to detecting masquerade attacks to mobile, remotely controlled devices have tried to dispense with behavioral analysis entirely, using independent verification of control signal features like power fading that may reveal the onset of a masquerade attack even when interaction style is consistent [5]. These approaches assume the power spectrum of the control signal coming from the attacker to be somehow distinguishable from the one of the legitimate user, e.g. due to different distances to the device under attack. Again, we argue that the underlying assumption may not apply depending on the communication channels used. In the case of Line-Of-Sight (LoS) control, for instance, the attacker trying to hijack the device can roughly at the same distance from it as a legitimate user.

This paper focuses on a scenario where civil Unmanned Aerial Vehicles (UAVs) are remotely controlled by differ-

[1] The latter may be optimistic, as the thief will almost certainly try to inactivate the detector.

ent pilots. We assume no (or weak) authentication on the ground-to-aircraft command channel, and no difference in the low-level timing or power of the control signals. We also assume that (i) pilots carry out identical maneuvers and (ii) trustworthy recordings of each pilot's behavior are available.

We show that pilots can be quickly identified based on *how* they control the UAV rather than on *what* they are doing with it, i.e. on the structure, cadence, and timing of users' command patterns. Also, we show that the classification algorithm is simple enough to be executed on board the UAV, and is therefore suitable for prevention of hijacking or masquerading attacks. Our results pave the way to online behavioral control systems capable of detecting hijackers even when they are performing the same operations as the real pilots, as well as to black box forensics analysis to detect hijackers takeover in post-mortem analysis of UAV crashes.

A. Background

Unmanned Aerial Vehicles (UAVs) (often called *drones*) can fly autonomously or in manual piloting mode. In both cases, a communication link with a ground control station (GCS) is required. In the autonomous flight mode, GPS data are usually used for navigation. In the manual flight mode the drone moves under the direct control of a pilot who either has the drone in Line-of-Sight (LoS) or controls it from a virtual cockpit.

In turn, the flight behavior of a pilot can be monitored by recording the sequence of flight commands sent to the drone, by recording the behavior of the drone measured using on-board sensors, or both. Selecting one of these approaches depends on various factors including the desired security and system performance levels. In principle, the first approach can prevent any malicious command from reaching the UAV on-board flight controller. It, however, can cause delay in command execution, deteriorating responsiveness to legitimate commands. In contrast, using the UAV on-board sensors to analyze the pilot behavior means that commands –both legitimate or malicious– have already reached the flight controller and taken effect.

In this paper we adopt the first approach and model pilot behavior as a sequence of flight commands. Each flight command is defined as a set of four values that represent the required level of control for the drone's pitch, roll, yaw, and thrust. It is important to remark that when a Radio Control (RC) transmitter is used for ground-to-drone communication, these values may have different physical representations. A typical RC transmitter used for remote control has two joysticks that can be moved by the pilot in two directions each to generate the desired four control signals. Possible representations of these signals are the voltage levels at the joystick potentiometers, the position or the code of the generated pulse in the transmitter (PPM vs. PCM), the digital representation of these signals at the input of the drone on-board flight controller, or the pulse wide modulation signal (PWM) at the output of the flight controller, which controls the power flow to the drone actuators.

B. Contributions

A first contribution of this paper is establishing a technology-independent, standard experimental setting for command-based UAV pilot identification techniques, whose final goal is classifying flight commands on-board. Thus, only representations readily available to the drone embedded system are interesting. Among representations available on board the UAV, the output of the flight controller toward actuators is inappropriate for two reasons. First, this signal does not represent the desired control level but the error between the desired control level and the current state of the drone. More importantly, the flight controller output heavily depends on the drone type. In multi-copter drones, for example, the flight controller translates the desired levels of pitch, roll, yaw, and thrust into motor speed signals. Consequently, we selected the digital representation of control signals at the input of flight controller as the most appropriate input for classification. The supplier of the drone used in our experimentation provides a communication protocol that allows bouncing radio-control data back to the ground at a frequency of 10 Hz. We developed a software application to capture and log these commands. This way, each line in the log file corresponds to a flight command comprising the values of pitch, roll, yaw, and thrust in addition to a time stamp, a standard format for benchmarking and comparing classifiers.

The second major contribution are the results of the experimentation itself. Twenty trained pilots repeatedly flew the drone through three different trajectories over multiple sessions. A total of 105,261 data points were collected, labeled, and pre-processed. Nine simple time-domain features were determined. Several classifiers were tested and several aspects were investigated. The results of our experimentation are summarized below:

1) Pilots can be identified with high accuracy using a simple random forest classifier with ten trees.
2) Identification accuracy - in general very good - differs from pilot to pilot and depends on the flight trajectory.
3) Raw RC signals are the only significant features for pilot identification (in particular, their time derivatives do not improve classification). Thrust is the most significant RC signal.
4) Pilots' behaviors differ more clearly during the take-off phase of a flight.

The remainder of the paper is structured as follows. Section II reviews the related work. Section III describes in detail our experimental setup and the data gathering. Section IV describes the data preprocessing and feature selection. Classification results are presented in Section V and discussed in Section VI. Section VII presents some implications and limitations of this research and Section VIII concludes the paper.

II. Related Work

To the best of our knowledge, drone pilot identification using behavior analysis has never been attempted so far, and the solution proposed in this paper is the first of its kind. Our solution, however, relates to drone security as an application domain and to behavioral biometrics as methodology. Relevant related work in both fields is reviewed in the next two subsections.

A. Related Work on Drone Security

Although highly critical, the security of UAVs has not yet attracted sufficient attention on the part of the research community. On the one hand, this can be due to the fact that research on military drones is regarded as classified and access to them is restricted. On the other hand, suppliers of civil and commercial drones are still busy with functional aspects of drones such as platform design and construction, dynamic modeling, flight control, as well as navigation and guidance [12]. A few papers on UAV security are available. For example, in [13] and [14] different approaches to UAV risk assessment were presented. Other papers focus on the vulnerabilities of specific UAV models. Due to its using open Wi-Fi, the Ar.Drone quadrotor was found to be vulnerable to different attacks including hijacking, data eavesdropping, and persons' tracking [14]. In [15] the issue of unencrypted Wi-Fi link was resolved by configuring the Ar.Drone as client and the control device as access point.

GPS spoofing is another threat to civil drones: UAVs can be easily hijacked [16], [17] by sending them fake GPS data, a technique already used by hackers for deceiving mobile phones' location system. GPS spoofing can be mitigated using a redundant system (e.g., a Doppler radar [18]) for estimating the GPS source distance. Military GPS signals are usually encrypted, which makes GPS spoofing less of an issue. In [19] a light-weight hardware cryptoengine was presented to secure the command and data communication between the ground control station and the UAV.

B. Related Work on Behaviometrics

As we outlined in Section I-A, behavioral biometrics (behaviometrics) studies the behavior of users interacting with devices, for the purpose of identification. In a comprehensive study, Yampolski and Govindaraju identified 28 behaviors used in behaviometrics research [20]. Keystroke dynamics [21], mouse dynamics [22], and touch dynamics [23] are examples for well-studied behaviors. Car driver behaviometrics is probably the closest area to the presented work. Driving signals have long been investigated to detect specific driver behaviors such as fatigue, drowsiness, and distraction [24]–[26]. In the last ten years some researchers applied behaviometrics for the purpose of driver identification using different control signals, such as the pressure applied on gas and brake pedals as well as the steering wheel angle [6]–[11]. In summary, researchers in this area found out that the identification accuracy of a driver depends on the driving manoeuvre or route and that the driving signals provide different classification performance, whereas braking, acceleration, and steering signals are the most significant signals.

III. Experimental Setup and Data Gathering

For our experimentation, the Hummingbird quadrotor UAV by Ascending Technologies was used. The vendor provides a low-level protocol that allows to acquire different data including radio control data at a frequency of 10Hz. A small application was developed to receive and log these data into a CSV file. Each line in the file represents what is referred to as a *flight command* in this paper. A flight command at any time point t is a 5-tuple of the form $C_t = (t, th(t), p(t), r(t), y(t))$. The elements of this tuple are explained in Table I. Table II explain the effect of the drone piloting signals.

TABLE I
SPECIFICATION OF A CONTROL COMMAND IN THE USED UAV

Command Element	Meaning	Range	Neutral Value
t	Time stamp of the command	$0-\infty$	NA
$th(t)$	Value of thrust signal at t	$0-4096$	NA
$p(t)$	Value of pitch signal at t	$0-4096$	2048
$r(t)$	Value of roll signal at t	$0-4096$	2048
$y(t)$	Value of yaw signal at t	$0-4096$	2048

TABLE II
UAV PILOTING SIGNALS

Rotary-wing UAV flying signals	Effect
Thrust	Increasing thrust: Vertical acceleration (flying upwards)
	Decreasing thrust: Vertical deceleration (flying downwards)
Pitch	Positive pitch: Longitudinal acceleration (flying forwards)
	Negative pitch: Longitudinal acceleration (flying backwards)
Roll	Positive roll:Lateral acceleration (flying right)
	Negative roll: Lateral acceleration (flying left)
Yaw	Positive yaw: Turning the UAV head to the left
	Negative yaw: Turning the UAV head to the right

TABLE III
FLIGHT TRAJECTORIES

Trajectory	Specification	Repetition	Total No. of data points
Vertical	The pilot flew the drone vertically up and down five times without stop. The maximum height was almost five meters.	Each pilot repeated this flight at least five times over multiple sessions.	19,901
Triangular or horizontal	The pilot flew the drone between three points at the same height. This was repeated five times without stop. The distance between every two points is almost four meters.	Each pilot repeated this flight at least three times over multiple sessions.	21,950
Random	The pilot flew the drone for five minutes in a random manner.	Some pilots repeated this flight a second time.	63,410

A neutral value refers to the value sent by the transmitter when the corresponding stick is released. Note that the pitch, roll, and yaw sticks return to the center when they are released. In contrast, the thrust has no neutral value because its stick keeps position when released.

Twenty trained pilots were involved in the data collection process over multiple sessions. The pilots are referred to as Pilot 1 to Pilot 20 in this paper. Each pilot flew three trajectories as specified in Table III. A total of 105,261 data points, i.e. commands, was collected. All flights took place in the indoor robotics lab dedicated to UAV flights at Khalifa University, in the UAE.

Fig. 1 shows plots of the RC signals generated by two pilots in the first 30 seconds of a vertical flight. It is important to remark that a vertical flight cannot be accomplished just by providing a thrust signal only as it might be expected. Rather, a UAV pilot repeatedly needs to apply pitch, roll, and yaw controls to counter drone drift. By observing these plots, it can be seen that Pilot 4 tended to control the drone more frequently and with more power than Pilot 5, in general. Also, while Pilot 5 tended to roll the drone in one direction

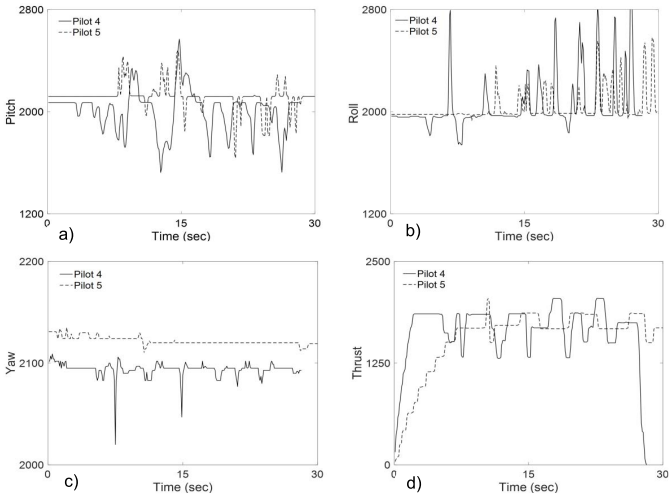

Fig. 1. Samples of the RC signals from two pilots during a vertical flight.

only, Pilot 4 applied the roll signal in two directions. With respect to the pitch control, however, Pilot 4 did not change the direction of the corresponding signal as much as Pilot 5 did. Such observations indicate individual flight behaviors that can be characteristic to pilots. Note that the pitch, roll, and yaw signals return to values, which are different from the neutral value 2048 defined in Table I. This indicates that the pilots have trimmed the flight controls differently which resulted in adding different DC offsets to the respective signals. A typical RC transmitter has four small trim levers that can be used by the pilot to compensate for any tendency of the specific UAV to drift in an unwanted direction. Trimming is performed by each pilot individually and we saw no obvious reason to remove the associated offset before applying machine learning. Nevertheless, the findings of this study would be less interesting, if the pilot identification performance would be determined by the trim offset entirely or significantly. Therefore, we decided to remove the DC offset from the pitch, roll, and yaw signals in the preprocessing phase and to perform most tests without this component. However, an additional test keeping the trim offset unchanged has not shown any significant variation in the classification performance (see Section V).

IV. Data Preprocessing and Feature Selection

Regardless of the classification algorithm, selecting relevant features is essential for classification accuracy. Initially, we selected nine simple features in the time domain including:

1) The pitch, roll, yaw, and thrust signals $p(t)$, $r(t)$, $y(t)$, and $th(t)$.
2) The time derivatives of these signals $p'(t)$, $r'(t)$, $y'(t)$, and $th'(t)$.
3) Control Simultaneity Variable $CSV(t)$. This variable describes which control signals are available simultaneously at any time point.

In order to clarify why we included the CSV feature, we remark that UAV pilots differ in the number of controls that they apply at each time point. Some pilots, for example, tend to provide multiple signals, e.g., pitch and roll at the same time. $CSV(t)$ takes a value between 0 and 15 which corresponds to the decimal value of a binary signal of the form $S_pS_rS_yS_{th}$. $S_p = 1$ when the pilot applies a pitch control, otherwise $S_p = 0$. Similar statements apply to S_r, S_y, and S_{th}. If $S_pS_rS_yS_{th} = 1101$, for example, then the pilot is applying pitch, roll, and thrust controls at the same time, so, $CSV(t) = 13$ in this case.

TABLE IV
Performance of Five Classifiers Using All Features and All Data Points (%)

Classifier	Vertical	Triangular	Random
RF	89.1	87.8	81.6
KNN	62	54.2	56.3
SVM	49.9	50.8	46.1
QD	22.6	22.7	23.1
LD	27.3	18.3	17.3

The tests described in Section V were performed with and without Principal Component Analysis (PCA) to verify whether the problem's feature space can be downsized by decorrelation. We found out that PCA leads to a deterioration of the classification accuracy by 10% on average. On the one hand, this can be explained by the fact that a few features (essentially the four raw signals) is significant for the classification as will be discussed later. So, the addressed classification problem has low dimensionality. On the other hand, the correlation between the pitch and roll signals as well as the correlation between the thrust and the yaw signals seem to have a significant role in the classification. Recall that the pitch and roll are controlled by one joystick. Pilots differ in how they provide these signals (e.g., simultaneously or successively). Also, it happens frequently that the joystick moves in an unintended direction depending on the pilot's experience. For example, when the pilot aims to provide pitch, she/he needs to push the stick in one direction. However, it is difficult to avoid fluctuation in the perpendicular direction especially in critical situations under high speeds. The same applies to the thrust and yaw signals. Such fluctuations seem to be important for pilot identification.

Alternative preprocessing techniques were tested, however, without significant impact on the classification performance. These include removing the DC offset caused by trimming as described in the previous section as well as filtering out the mode (representing the most frequent value) of the pitch, roll, yaw, and thrust signals. The results presented in the next section correspond to tests without PCA or mode rejection, but after removing the DC-offset.

V. Tests and Results

Five different classifiers were tested: linear discriminant (LD), quadratic discriminant (QD), support vector machine (SVM) with a Gaussian kernel ($C = 1$, $\gamma = 0.75$), weighted k-nearest neighbors (KNN), and random forest (RF) using 30 trees. A 5-fold cross validation was applied in all the tests.

The classification results using all the nine features are shown in Table IV. The random forest classifier clearly gave the best classification accuracy ranging between 81.6-89.1%

TABLE V

TRUE POSITIVE RATES FOR DIFFERENT PILOTS IN DIFFERENT FLIGHTS IN PERCENT (%)

Pilot	Vertical	Triangular	Random	Average
Pilot 1	92	81	88	87
Pilot 2	98	91	90	93
Pilot 3	98	96	96	97
Pilot 4	81	78	66	75
Pilot 5	70	95	65	77
Pilot 6	96	96	91	94
Pilot 7	98	99	96	98
Pilot 8	82	91	89	87
Pilot 9	70	89	89	83
Pilot 10	90	95	75	87
Pilot 11	90	97	72	86
Pilot 12	94	77	82	84
Pilot 13	93	76	76	82
Pilot 14	76	79	64	73
Pilot 15	79	96	73	83
Pilot 16	83	72	75	77
Pilot 17	77	71	66	71
Pilot 18	88	88	85	87
Pilot 19	85	98	92	92
Pilot 20	87	99	85	90

while the discriminant classifiers had the lowest performance. Therefore, most of the other tests described below are limited to random forests unless mentioned explicitly.

Table V summarizes the true positive rates extracted from three confusion matrices for the vertical, triangular, and random flight. Obviously, not all pilots can be identified with the same accuracy and the behavior of most pilots depend on the flight trajectory. Pilot 1, for example, can be best identified by his behavior during a vertical flight. In contrast, Pilot 20 shows a highly distinguishable behavior during the triangular flight. On the other hand, the behavior of Pilot 3 seems to be less affected by the type of flight trajectory.

We investigated the classification performance when the model is trained with data from some trajectory and tested with data from the other two trajectories. Expectedly, the classification behavior dropped significantly with this type of test. For example, when we trained with data from vertical flights and tested with data from horizontal (i.e. triangular) and random flights we obtained a classification accuracy of 26.8%. Similarly, training with data from horizontal or random flights and testing with the other data gave an accuracy of 20.7% or 31.6%, respectively. These results suggest that an on-line classifier should support different models that are activated according to current flight situation which can be estimated using data from the drone's inertial measurement unit (IMU).

Table VI shows the results of a test, which was performed to understand the impact of the number of trees on the RF classification performance. Accordingly, increasing the number of trees beyond ten does not add a significant improvement in classification accuracy.

A single-feature analysis was performed to identify the contributions of different features to the classification performance. Table VII shows the results of this test.

Interpretation of the analytics results is given below:

1) Any raw control signal provides better classification performance than any derivative in any flight trajectory.
2) On average, the thrust signal provides the highest accuracy followed by yaw, pitch, and roll signals in this order.

TABLE VI

CLASSIFICATION ACCURACY FOR DIFFERENT NUMBER OF TREES (%)

Number of Trees	Vertical	Triangular	Random
1 Tree	74.6	79.5	67.7
5 Trees	84	85.5	76.7
10 Trees	87.2	87	79.5
20 Trees	88.4	87.5	81.2
30 Trees	89.1	87.8	81.6

TABLE VII

SINGLE-FEATURE PERFORMANCE USING RANDOM FOREST AND ALL DATA POINTS (%)

Feature	Vertical	Triangular	Random
$p(t)$	25.2	27.6	43.3
$r(t)$	27.4	24.2	38.1
$y(t)$	36.5	41	44
$th(t)$	47.1	64.4	49.6
$p'(t)$	13.6	11.4	21.7
$r'(t)$	14	11.4	20.5
$y'(t)$	13.2	13.3	20.6
$th'(t)$	13.8	13.6	19.6
$CSV(t)$	16.1	14.5	22.8

TABLE VIII

GROUPED FEATURES PERFORMANCE USING RANDOM FOREST AND ALL DATA POINTS (%)

Feature Set	Vertical	Triangular	Random
All features	89.1	87.8	81.6
Four RC signals only	89.4	87.9	82.2
Four derivatives of RC signals only	25	20.9	19.6

3) The control simultaneity variable $CSV(t)$ provides a classification performance that is on average comparable with the performance of the derivative signals. In other words, applying multiple controls is a common, but not a strong distinctive feature of UAV pilot behavior to be taken alone.

An additional test showed that the pitch, roll, yaw, and thrust signals together provide classification results that are slightly better than the results obtained when all nine features are considered, as can be seen in Table VIII. In contrast, when only derivative signals are considered, the classification accuracy drops significantly.

To investigate the effect of limiting the classification to relevant data a new test was performed where only 10-point data segments, which include significant changes, were considered. A significant change was defined by 20 discrete levels in any signal value including the raw signals, their derivatives, or $CSV(t)$. This filtering reduced the amount of data points to 10.9%, 16.4%, or 15.1% of the original data for the vertical, triangular, and random flights, respectively. As can be seen from Table IX, however, this resulted in a considerable drop in the classification performance. Specifically, the classification accuracy dropped by 18.5%, 5.8%, or 11% for the vertical, triangular, and random flights, respectively.

Another test was performed to investigate the pilot behavior during take-off. In this test only the first 30 points (corresponding to the first three flight seconds) of the flight data were considered. This test showed some

TABLE IX

CLASSIFICATION RESULTS FOR SIGNIFICANT-CHANGE SEGMENTS COMPARED TO ALL-DATA CLASSIFICATION (%)

Feature Set	Vertical	Triangular	Random
All data	89.1	87.8	81.6
Data segments with significant changes	70.6	82	70.6

TABLE X

TRAINING WITH TAKE-OFF DATA ONLY IMPROVES THE CLASSIFICATION ACCURACY EVEN FOR THE SVM CLASSIFIER (%)

Classifier	Data	Vertical	Triangular	Random
RF	All flight data	89.1	87.8	81.6
	Take-off data (3 sec)	96.3	91	90.4
SVM	All flight data	49.9	50.8	46.1
	Take-off data (3 sec)	79.7	75.6	71.3

improvement in the identification accuracy using random forests. However, the SVM classifier provided significantly higher accuracy as can be seen in Table X. While performing identification at take-off time would decrease the generality of the approach, this result can be useful for some applications like feature-based authentication on shared devices. An additional application of only authenticating during take-off could be the check for a remote control steal, even if the communication between the RC and the UAV is secure.

A further test was performed to investigate the influence of the trim offset discussed in Section III. It was found out that this offset has only a slight positive impact on the classification accuracy. The average improvement for all classifiers and all trajectories was found to be 0.54%. For the random forest classifier, the improvement was 0.3% or 0.2% for the vertical and triangular data, respectively. Keeping the offset in the random flight data, in contrast, did not add any improvement. The highest increase in the classification accuracy was 4.4% using the linear discriminant classifier on the triangular flight's data.

Finally, to investigate the classification performance on data from subjects, who were not used to train the classifier (i.e. out-of-sample data), the RF classifier was trained with one authorized pilot and a variable number of unauthorized pilots (from 1 to 19). In each case, unseen data from all other pilots were used to test the classifier. All possible combinations of authorized and unauthorized pilots were tested and the mean value as well as the standard deviation of the classification accuracy were determined. The results are shown in Fig. 2. Accordingly, the mean value of the specificity (true negatives) consistently increases when more unauthorized pilots are used to train the classifier. It starts below 50% with a single unauthorized pilot and increases above 75% with three. The specificity keeps increasing till it reaches a plateau around 96-97% with higher number of unauthorized pilots in the trained model (17-18 pilots). The standard deviation decreases with the number of unauthorized pilots involved in the training.

VI. DISCUSSION

A major outcome of this experimentation is showing that the behavior of a pilot using a typical RC transmitter to control a

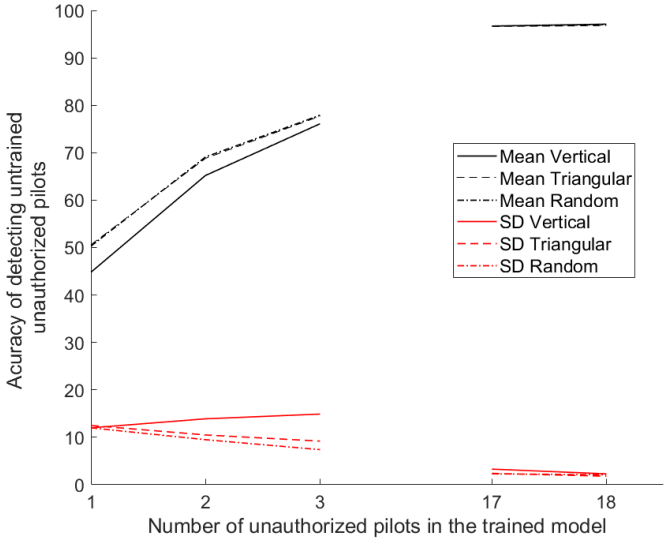

Fig. 2. Average classification performance (specificity) when the model is trained with data from one authorized pilot and different numbers of unauthorized pilots; and tested with all pilots' data.

UAV is rich enough to be used for identification purposes. In other words, RC signals can be used as a behavioral biometric feature for pilot identification. Among five tested classifiers, the random forest is the most promising one. Good performance of random forest in this type of problems is not per se a new result; in their survey work, Verkas *et al*. found out that there are more reports on cases where RF outperforms other techniques than the other way round [27]. Also, Fernandez-Delgado *et al*. tested 179 classifiers on 121 data sets and found out that random forest versions provide the best results [28].

Test results presented in the previous section show a better classification performance during a vertical flight. As mentioned previously, flying a drone vertically is a more complex task than it might be expected because a straight up-down flight cannot be achieved solely by changing the thrust signal. Rather, a pilot must apply pitch, roll, and/or yaw controls frequently to overcome drone drift. This can be understood better by investigating the contributions of the signals $p(t)$, $r(t)$, $y(t)$, and $th(t)$ to the classification performance given in Table VII. While $th(t)$ is the most significant signal in all the trajectories, its contribution in the vertical flight is less than its contribution in the other two flights.

The relationship between classification accuracy and the flight trajectory can have practical importance. For example, if an on-line classifier has identified a drone hijack during a horizontal flight, which allows a lower classification accuracy, a decision can be delayed until this result is confirmed in a vertical flight, which allows a higher accuracy. We remark that multi-copter UAVs usually take off and land vertically as a rule. So, a higher classification accuracy in these flight phases can be especially relevant.

In terms of the masquerading attack scenario discussed in Section I-A, classification performance figures given in Table IV suggest that the classifier has the potential to differentiate between an authorized and a non-authorized pilot.

The classifier's ability to identify a specific authorized pilot is given in Table V. In particular, this table shows that not all pilots can be identified with the same accuracy and that the identification accuracy depends on the flight, in general. For example, when Pilot 19 or Pilot 20 fly the drone horizontally, the classifier can identify them with almost perfect accuracy. In contrast, the accuracy of identifying Pilot 1 or Pilot 12 during a horizontal flight is only 81% or 77%, respectively. A safeguard against hijacking can make use of this aspect by linking the output of the classifier to the current flight path. For example, assuming that the classifier has identified Pilot 12 during a horizontal flight, the safeguard function will be aware that the accuracy of identifying Pilot 12 during a horizontal flight is not very high. So, the safeguard can wait until the pilot flies a vertical trajectory to confirm the first result.

Some pilots like Pilot 14 and Pilot 17 show relatively lower identification accuracy in all three trajectories. In a practical application, this may cause a problem if these subjects would be the actual authorized pilots. It would be interesting to understand the behavioral components that affect the identification accuracy of such pilots. This will be addressed in future work.

In practical applications, a specific drone may have a few number of pilots who are authorized to fly it. In a specific flight, the drone usually has even only one authorized pilot (civil drones). The results of the tests given in Fig. 2 show that the model should be trained with data from multiple unauthorized pilots to obtain a good classification performance in this case. The described tests can be used as a guideline for performing similar tests with two or more authorized pilots.

Table VII shows that the thrust signal provides the most significant contribution to the identification accuracy. This can be explained by the fact that a "non-zero" thrust signal must always be available for the drone to be in the air regardless of the flight trajectory or target. In contrast, pitch, roll, and yaw signals are only applied occasionally by the pilot to change the flight direction horizontally or to rotate it around its vertical axis. It is, however, interesting to observe that the thrust signal is more significant in the horizontal (triangular) flight than in the vertical flight although the thrust is responsible for the vertical movement. However, significance of the pitch, roll, and yaw signals does not differ between vertical and horizontal flights considerably.

The derivatives of the RC signals $p'(t)$, $r'(t)$, $y'(t)$, and $th'(t)$ represent how fast the pilot moves the joysticks to change the level of the respective signals. According to our experiments (see Table VII and Table VIII in the previous section), the derivatives of the RC signals are not significant for pilot identification. This goes somewhat against expectations, as derivatives of control signals should be related to pilot reaction time, which is well known to be linked to individual physiological traits [29].

To explain this apparently counter-intuitive result, we plotted derivative signals from all pilots and observed that these signals are indeed sparse, i.e. zero most of the time. Further statistical analysis showed, for example, that the signal $p'(t)$ is zero in 59%, 81%, and 80% of all data points related to the vertical, triangular, or random flight, respectively. This is an important remark, showing unequal sparsity among the different features in the original feature set. Sparsity reduction by interpolation could conceivably be used to translate our sparse derivative representation to a dense one, e.g. by using the last recorded derivative as an estimate while the current value is zero (i.e. the pilot is showing a steady hand). We leave this post-processing to a future paper.

Another important aspect is the higher accuracy of pilot identification at take-off time shown in Table X. It looks of special relevance for continuous monitoring applications, as it promises to detect drone hijack very early in a mission. However, the classification model we used in the take-off time is not the same as the one extracted from all flight data. Depending on resource availability and how critical pilot identification in the take-off phase is, two models may be implemented with a mechanism to switch the classifier after the take-off phase.

Finally, UAVs operate under strict power, time, and memory constraints. Limiting classification to data segments that show significant changes can reduce the classification overhead significantly. Selecting a random forest with less trees reduces the memory usage and the computation overhead on the drone's embedded system. However, such mechanisms always come at the cost of classification accuracy as was shown in Table VI and Table IX. The length of data segments that represent significant changes and the number of trees can be used as trade-off parameters while designing an on-line classifier under consideration of actual constraints.

VII. Research Implications and Limitations

In this section we first discuss how the presented method can be used in a real application. Then we outline some limitations of this research.

The final target is to implement an on-line classifier that can identify authorized pilots on-the-fly. An on-line classifier uses models generated by the off-line machine learning process as presented in this paper. After buying a new drone, therefore, the first step is to collect flight data from multiple pilots to train off-line classifiers. Depending on the drone type, this can demand a hardware/software setup. Pilots involved in the training are known to the drone operator and can all be assumed as authorized. In actual usage later, however, only one or some of these pilots will be authorized to fly the drone. Depending on the security requirements by the drone operator, it may be sufficient to label the data generated by these pilots as authorized and the data generated by the remaining pilots as unauthorized resulting in a binary classification model. If the drone operator demands an exact identification of the authorized pilot, then the data generated by each authorized pilot should be labeled separately, e.g., by the name of the respective pilot. The data generated by the remaining unauthorized pilots belong to one class, resulting in a multi-class model as presented in this paper. A database should be setup and maintained gradually to facilitate access to flight data when new drones or new pilots are added. Such a database can also be used to capture behavior changes by the same pilot, e.g., due to experience and skill improvement.

To deal with such aspects, models should be adjusted from time to time by re-executing the training process with new data.

Finally, the classification/identification model generated by the off-line training process can be implemented on the drone under consideration of real-time aspects. The on-line classifier should be embedded into a complete system that evaluates the classification results, generates decisions based on built-in rules, and triggers appropriate safeguard functions.

This research has some limitations. First, while the presented concepts and methods are of general value, the results are specific to the drone used in this research and to the selected flight conditions and trajectories. It should be expected that pilots' flying behavior varies with the drone in use. For example, due to their weight, larger drones such as military ones can be less sensitive to small changes in the control signals and pilots can feel more comfortable flying such drones. On the other hand, military drones are highly sophisticated and they travel under higher speeds. This can make flying them much more difficult especially when they fly critical missions. This limitation can be dealt with by training classifiers for each drone type under flight conditions expected in actual operation.

Another limitation of this work is that it studies pilots' behavior with respect to flying a drone only and does not consider other behaviors related to executing different functions on the drone. Such functions include sensing the environment using cameras and other sensors as well as performing actions based on the sensor data such as firing a missile. Also, the addressed flying behavior is restricted to manual flight using typical RC transmitters whereas the drone is in line of sight. Flight modes using satellite or other communication channels were not considered. Although the basic flight controls (thrust, pitch, roll, yaw) are in general independent of the way they are sent to the drone, it should be expected that pilot's flying behavior depends on the communication link. So, classifiers for pilot identification should be trained using data collected in a corresponding setting.

VIII. Conclusion

In this paper, we showed that UAV pilot behavior is a unique signature that can be effectively used in pilot identification. In the first step, we described a experimental setup and gathered data from multiple drone flights in a controlled environment. In the second step we focused on feature engineering, identifying a complete set of features. We tested PCA and other preprocessing techniques but found out that these techniques do not improve classification accuracy in this application, mainly due to low dimensionality of the present problem. Our feature selection was validated by comparing the feature set performance with its subsets, and a compact version without signal derivatives was found to be equivalent to the original one. In the third step we used our feature set to compute multiple classification algorithms, comparing the accuracy of the results for complete and reduced feature sets. Identification accuracy was in general very good, depending on the pilot and on the flight trajectory under consideration. Different control signals show different levels of significance for the classification accuracy. In future work we will implement and test an on-board classifier based on the models established in this work, providing continuous protection against hijacking. Our component will be able to act as evidence provider in the framework of access control systems based on monitoring [30].

References

[1] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statist. Sci.*, vol. 16, no. 1, pp. 58–74, 2001.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 2, pp. 222–232, Feb. 1987.

[3] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Proc. IEEE Int. Conf. Depend. Syst. Netw. (DSN)*, Jun. 2002, pp. 219–228.

[4] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proc. ACM Workshop Vis. Data Mining Comput. Secur.*, 2004, pp. 1–8.

[5] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. 5th ACM Workshop Wireless Security*, 2006, pp. 43–52.

[6] Y. Nishiwaki, K. Ozawa, T. Wakita, C. Miyajima, K. Itou, and K. Takeda, "Driver identification based on spectral analysis of driving behavioral signals," in *Advances for In-Vehicle and Mobile Systems*. Boston, MA, USA: Springer, 2007, pp. 25–34.

[7] H. Qian, Y. Ou, X. Wu, X. Meng, and Y. Xu, "Support vector machine for behavior-based driver identification system," *J. Robot.*, vol. 2010, Mar. 2010, Art. no. 397865.

[8] E. Öztürk and E. Erzin, "Driver status identification from driving behavior signals," in *Digital Signal Processing for In-Vehicle Systems and Safety*. New York, NY, USA: Springer, 2012, pp. 31–55.

[9] M. Van Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Proc. Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1040–1045.

[10] V. Martinez, I. Del Campo, J. Echanobe, and K. Basterretxea, "Driving behavior signals and machine learning: A personalized driver assistance system," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 2933–2940.

[11] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proc. Privacy Enhancing Technol.*, vol. 2016, no. 1, pp. 34–50, 2016.

[12] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Syst.*, vol. 2, no. 2, pp. 175–199, 2014.

[13] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks—An approach to the risk assessment," in *Proc. 5th Int. Conf. Cyber Conflict (CyCon)*, Jun. 2013, pp. 1–23.

[14] F. Samland, J. Fruth, M. Hildebrandt, T. Hoppe, and J. Dittmann, "AR. Drone: Security threat analysis and exemplary attack to track persons," *Proc. SPIE*, vol. 8301, p. 83010G, Jan. 2012.

[15] J.-S. Pleban, R. Band, and R. Creutzburg, "Hacking and securing the AR. Drone 2.0 quadcopter: Investigations for improving the security of a toy," *Proc. SPIE*, vol. 9030, p. 90300L, Feb. 2014.

[16] S. M. Giray, "Anatomy of unmanned aerial vehicle hijacking with signal spoofing," in *Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2013, pp. 795–800.

[17] K. Wesson and T. Humphreys, "Hacking drones," *Sci. Amer.*, vol. 309, no. 5, pp. 54–59, 2013.

[18] M. S. Faughnan *et al.*, "Risk analysis of unmanned aerial vehicle hijacking and methods of its detection," in *Proc. IEEE Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2013, pp. 145–150.

[19] A. Shoufan, H. AlNoon, and J. Baek, "Secure communication in civil drones," in *Information Systems Security and Privacy*. Cham, Switzerland: Springer, 2015, pp. 177–195.

[20] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: A survey and classification," *Int. J. Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[21] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *Sci. World J.*, vol. 2013, Aug. 2013, Art. no. 408280.

[22] K. Revett, H. Jahankhani, S. T. de Magalhães, and H. M. Santos, "A survey of user authentication based on mouse dynamics," in *Global E-Security*. Berlin, Germany: Springer, 2008, pp. 210–219.

[23] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Comput. Secur.*, vol. 59, pp. 210–235, Jun. 2016.
[24] M. V. Yeo, X. Li, K. Shen, and E. P. Wilder-Smith, "Can SVM be used for automatic EEG detection of drowsiness during car driving?" *Saf. Sci.*, vol. 47, no. 1, pp. 115–124, 2009.
[25] A. B. R. Gonzalez, M. R. Wilby, J. J. V. Diaz, and C. S. Ávila, "Modeling and detecting aggressiveness from driving signals," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1419–1428, Aug. 2014.
[26] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1892–1897.
[27] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, 2011.
[28] M. Fernáandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.
[29] T. P. Colgate, "Reaction and response times of individuals reacting to auditory, visual, and tactile stimuli," *Res. Quart. Amer. Assoc. Health, Phys. Edu. Recreation*, vol. 39, no. 3, pp. 783–784, 1968.
[30] E. Damiani, M. Anisetti, and V. Bellandi, "Toward exploiting location-based and video information in negotiated access control policies," in *Proc. Int. Conf. Inf. Syst. Secur.*, 2005, pp. 21–35.

Abdulhadi Shoufan received the Dr.-Ing. degree from Technische Universität Darmstadt, Germany, in 2007. He is currently an Assistant Professor of information security and electrical and computer engineering and a member of the Information Security Center, Khalifa University, Abu Dhabi. He is interested in drones' security and safe operation as well as in embedded security, learning analytics, and engineering education.

Haitham M. Al-Angari received the Ph.D. degree in electrical engineering from Northwestern University, Evanston, IL, USA, in 2005. He is currently a Researcher with the Biomedical Engineering Department, Khalifa University. His research interests include biomedical instrumentation and signal processing, machine learning, and finite-element modeling.

Muhammad Faraz Afzal Sheikh received the B.Sc. degree in mechatronics and control engineering from the University of Engineering and Technology, Lahore, in 2006, and the M.Sc. degree from the University of Genova, Italy, in 2007. He was a Research Scientist with the Department of Mechanics and Robotics, University of Duisburg Essen, Germany. He is currently an Associate Instructor with the Electronics and Computer Engineering Division, New York University, Abu Dhabi.

Ernesto Damiani is the leader of the Big Data Initiative at EBTIC/Khalifa University, Abu Dhabi, UAE, and a Full Professor with the Università degli Studi di Milano, where he leads the SESAR Research Lab. He is the Principal Investigator of the H2020 TOREADOR Project. He received the Chester-Sall Award from the IEEE IES Society in 2007. He was a recipient of the Stephen S. Yau Services Computing Award in 2016. He was named ACM Distinguished Scientist in 2008.