Trust the Crowd: Wireless Witnessing to Detect Attacks on ADS-B-Based Air-Traffic Surveillance

Kai Jansen
Ruhr University Bochum, Germany
kai.jansen-u16@rub.de

Liang Niu
New York University, USA
liang.niu@nyu.edu

Nian Xue
New York University, USA
nian.xue@nyu.edu

Ivan Martinovic
University of Oxford, UK
ivan.martinovic@cs.ox.ac.uk

Christina Pöpper
New York University Abu Dhabi, UAE
christina.poepper@nyu.edu

***Abstract*—Automatic Dependent Surveillance-Broadcast (ADS-B) has been widely adopted as the de facto standard for air-traffic surveillance. Aviation regulations require all aircraft to actively broadcast status reports containing identity, position, and movement information. However, the lack of security measures exposes ADS-B to cyberattacks by technically capable adversaries with the purpose of interfering with air safety. In this paper, we develop a non-invasive trust evaluation system to detect attacks on ADS-B-based air-traffic surveillance using real-world flight data as collected by an infrastructure of ground-based sensors. Taking advantage of the redundancy of geographically distributed sensors in a crowdsourcing manner, we implement verification tests to pursue security by wireless witnessing. At the core of our proposal is the combination of verification checks and Machine Learning (ML)-aided classification of reception patterns—such that user-collected data cross-validates the data provided by other users. Our system is non-invasive in the sense that it neither requires modifications on the deployed hardware nor the software protocols and only utilizes already available data. We demonstrate that our system can successfully detect GPS spoofing, ADS-B spoofing, and even Sybil attacks for airspaces observed by at least three benign sensors. We are further able to distinguish the type of attack, identify affected sensors, and tune our system to dynamically adapt to changing air-traffic conditions.**

I. Introduction

The monitoring of air traffic has evolved from an analog Radio Detection and Ranging (RADAR)-based system to a digitally-aided surveillance infrastructure. Effective from January 1, 2020, all aircraft are required to be equipped with an Automatic Dependent Surveillance-Broadcast (ADS-B) system to access most of the world's airspace [54], which hence constitutes the *de facto* standard for air-traffic monitoring. ADS-B-capable transmitters periodically broadcast status reports that inform others about their identification, position, movement, and additional status codes.

While the aviation industry is characterized by very long development cycles—up to several decades—, applications that mandate high safety guarantees are usually lagging behind advancements on the security side. As such, ADS-B reports are neither encrypted nor authenticated. At the same time, the open specification of ADS-B promotes the collection and free usage of aircraft reports. Simple sensors can decode aircraft broadcast reports and gain a real-time view of their surrounding airspace. A network that combines more than 1000 user-operated ground-based sensors in a crowdsourcing manner is the OpenSky Network [39]–[42], [47]. This network collects and stores air-traffic data from around the world and makes them available for research.

Since ADS-B lacks fundamental security practices, the exposure to cyberattacks targeting air traffic has long been discussed [5], [19], [24], [35], [36], [43], [44], [48]. These works demonstrate how attackers can interfere with aircraft sensors and how fake aircraft messages can be injected into air-traffic monitoring systems [5]. For instance, adversaries with commercial off-the-shelf hardware and moderate knowledge can generate arbitrary messages mimicking valid ADS-B reports [44], [48]. The consequences of such attacks range from distraction on the flight deck or in the control room up to violations of mandatory safety separations, and eventually increasing the possibility of aircraft collisions. Since the implementation of these attacks is far from being only of academic nature, security solutions are urgently needed to protect the integrity of air-traffic surveillance [4]. In fact, data trust establishment is an open and central problem in the aviation industry and emerging concerns have already reached the public [4], [11], [14], [15], [63].

To answer the demands for more security in the safety-driven aviation industry, we propose a data-centric [32] *trust evaluation system* with the goal of assessing the trustworthiness of ADS-B reports using data that is already collected at wide scale. We refer to trust in the sense that messages are trustworthy when they originate from functional, non-malicious sources. In contrast, error-prone or attacker-controlled messages trying to harm the system should be detected. Furthermore, we explore the identification of the type of attack and the traceability of malicious sensors.

The development of such a system faces several challenges imposed by the highly regulated aviation industry. Viable solutions need to be non-invasive in the sense that they do not

Network and Distributed Systems Security (NDSS) Symposium 2021
21-24 February 2021
ISBN 1-891562-66-5
https://dx.doi.org/10.14722/ndss.2021.24552
www.ndss-symposium.org

require any modifications on the deployed hard- and software. In particular, security systems should not interfere with other systems already in place to avoid lengthy (re)certification processes [4]. Preferably, solutions are augmentation systems that operate autonomously with sensor input already available. We develop our system to fulfill all these challenges.

At the core of our system, we make use of the crowdsourcing nature of a sensor network in which user-collected data cross-validates data provided by other users. Forming a network of trusted sensors based on mutual auditing, we pursue *wireless witnessing*. Wireless witnessing is the collaborative process of observing the status of a distributed wireless system. We apply it in the security context to assess and validate the trustworthiness of ADS-B reports. In particular, we implement a Machine Learning (ML)-based verification test that is trained on typical message reception patterns[1]. The collaboration of sensors characterizes expected reception patterns of aircraft reports transmitted from certain airspace segments while automatically factoring in natural message loss.

Our system can reliably differentiate between normal air-traffic broadcasts and suspicious reports diverging from expected patterns if at least three sensors observe the same airspace. This assumption is already fulfilled by the majority of the considered airspace. Furthermore, our system can recognize the type of attack, e. g., GPS spoofing or ADS-B spoofing to trace affected sensors and identify the sensor redundancy as an important factor. While minimizing false alarm events, we achieve detection rates beyond $95\,\%$ for moderate GPS spoofing deviations and any form of ADS-B spoofing. To further harden the network against attacks, new sensors can be integrated by providing consistent snapshots of their airspaces. Since our system is solely based on an already existing infrastructure and does not require any modifications on aviation systems, it is non-invasive and could be implemented today easing very long certification processes. In contrast to existing solutions for air-traffic verification [10], [21], [22], [26], [37], [38], [52], [60], we do not require the measurement of time, frequency shifts, or any PHY layer features, but only use discrete sensor events.

In summary, the contributions of this paper are:

- We propose the first comprehensive approach to evaluate the trustworthiness of ADS-B aircraft reports based on an existing infrastructure of crowdsourcing sensors.
- We demonstrate the applicability of our approach by incorporating real-world flight data collected by geographically distributed sensors at a large scale.
- We simulate prominent attacks on GPS and ADS-B, detect their presence via validation in our trust system, and draw conclusions about their type and origin.
- We elaborate on network expansion and optimized sensor deployment to further harden the network against attacks in the future.

II. System and Attacker Models

We first describe today's air-traffic monitoring techniques with a focus on ADS-B. We then introduce our trust definition and present the consolidated system model. Finally, we define the considered attacker model.

[1] https://github.com/kai-jansen/ADSB-Trust-Evaluation

A. Air-Traffic Monitoring

In recent years, traditional analog RADAR-based systems for air-traffic monitoring have been augmented with digital means for active wireless communication. For the communication with ground stations and other aerial vehicles, aircraft are mandated to be equipped with ADS-B transponders that periodically broadcast status reports [54]. These reports contain aircraft identification, information on speed, track, and acceleration along with further observation data. The positioning information is mainly derived via GPS, which is the preferred method for self-localization.

Since the ADS-B protocol is openly specified, the modulation and data frame patterns are known. ADS-B operates at a frequency of $1{,}090\,\mathrm{MHz}$ and the typical reception range can reach up to $700\,\mathrm{km}$. The signals can thus be received by simple consumer-grade hardware such as Universal Software Radio Peripherals (USRPs) [9] or even cheaper Software Defined Radios (SDRs) like RTL-SDR dongles [33], which are available for as low as $\$20$. The availability of SDRs not only allows passive eavesdropping but also led to software tools for active ADS-B transmission [6] or the generation of fake GPS signals [28]. Surprisingly, the ADS-B protocol lacks fundamental security measures, and neither applies encryption nor authentication.

B. Trust Definition

We define *trust* in our system as the certainty of an ADS-B report to be the result of normal behavior and not disrupted by malfunctioning or active manipulation. To this end, a trusted report represents valid data transmitted by genuine sources. On the other hand, an untrustworthy report is either erroneous or contains fake data that should be discarded from further processing. While the traditional notion of trust had been *entity-centric* and rigid, today's fast-changing ad hoc networks necessitate the adjustment of trust models.

Hence, we seek to establish a *data-centric* trust model in consideration of short-lived associations in volatile environments as mentioned by Raya et al. [32]. In particular, we design a trust system that is driven by data collected by geographically distributed sensors that share their observations within a network. The combination of redundant views enables the system to cross-validate data and eventually establish a form of wireless witnessing.

C. Consolidated System Model

We consider the following system model. Aircraft that are equipped with an ADS-B transmitter periodically broadcast status reports which among other information include GPS-derived positions. A set of geographically distributed sensors receive these reports and their observations are shared with others in a crowdsourcing manner. A central server collects and processes the forwarded observations. Overall, we are faced with the high mobility of aircraft, while the receiving sensors are stationary and are less likely to move significantly. Figure 1 depicts an overview of our system model that we consider to assess the trustworthiness of ADS-B reports.

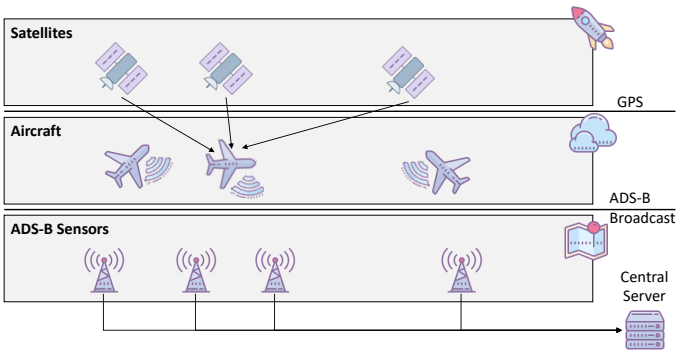

Fig. 1. Our considered system model of aircraft using GPS satellite signals for self-localization and ADS-B sensors forwarding aircraft reports to the processing central server.

D. Considered Adversary

Our adversary model comprises several prominent attack vectors, which we categorize according to their intended target and their scope. Table I shows an overview. We evaluate our proposed system against these attacks. Moreover, we will argue in Section VI-C that even attackers with complete knowledge about our verification scheme cannot bypass our implementation of wireless witnessing and can still be detected.

GPS Spoofing. The airborne (self)-positioning sensors process received GPS signals from multiple satellites to embed the results in the broadcasted ADS-B reports. One attack scenario considers the spoofing of GPS signals where an attacker sends out specially crafted signals at a considerable signal strength [16], [53]. As a result, an attacker can inject false positioning or timing information into the aircraft systems inducing the processing of fake attacker-controlled data [19].

ADS-B Spoofing (Single). An attacker capable of generating fake ADS-B messages can transmit arbitrary reports with full control over their contents [5], [24], [36]. These bogus reports may represent, e. g., any aircraft identifier, positioning solution, or movement information. Receivers of such messages will decode the message contents and forward the sensed information to the central server. We differentiate this attack according to the number of affected sensors. An attacker that is limited in its effective range is likely to only affect single sensors due to their broad spatial distribution.

ADS-B Spoofing (Multiple). A large-scale attacker may also be capable of targeting multiple geographically distributed sensors at the same time. This attacker, however, requires multiple antennas or a high elevated high power antenna. The attack is conducted in a broadcast fashion and is expected to affect all sensors within its targeted area. As a result, more than one sensor would receive the same fake report and forward it to the central server.

Sensor Control. Due to the open nature of the surveillance network, attackers may operate their own sensors and become part of the crowdsourcing infrastructure. Having full control over a sensor, an attacker is able to inject arbitrary data encapsulated in genuine ADS-B reports [36]. This attack can be performed without broadcasting any signals and can be directly conducted on the network level.

TABLE I. ATTACK VECTORS

Target	Attack	Scope	Effort
Aircraft	GPS Spoofing	-	Moderate
ADS-B Sensor(s)	ADS-B Spoofing	Single Multiple	Moderate High
Central Server	Sensor Control Sybil Attack	Single Multiple	Low High

Sybil Attack. A large-scale attacker operating a significant number of sensors can perform a Sybil attack [7] with the purpose of overruling the network's protection systems. The sensors may be deployed at different locations to influence several redundant views at the same time. This constitutes one of the most powerful attack against sensor networks.

III. Design of an ADS-B Trust System

We propose a system to establish a dynamic verification of ADS-B messages for air-traffic surveillance. We first describe the specifics of the analyzed data and state general network statistics. We then define (i) three verification tests checking the *contents* of a message and (ii) one ML-based classification of the report *metadata*, i. e., the reception pattern.

A. Data Source Specifics

As the source of our considered data, we utilize real-world air-traffic data from the OpenSky Network [39]–[42], [47]. The sensors are installed and operated by volunteers, who can either remain anonymous or opt to register by providing personal information. Over 1000 sensors promote the coverage of the network that exhibits a particular high sensor density in Europe and on the American continent. The network relies on user-provided data, processes it on centralized servers, and offers access to the collected data of around 20 billion messages per day. It is noteworthy that nodes in the network are not equipped with any cryptographic means or certificates, which would hinder the growth of the sensor network and contradict the easy access to the crowdsourcing platform. While other air-traffic sensor networks exist, we make use of the research-friendly data sharing of this network.

For the sake of simplicity, we initially restrict the considered ADS-B reports to the European airspace where the OpenSky Network sensor density is the highest. To further reduce complexity, we divide this space into non-overlapping square-shaped clusters C with edge lengths of approx. $10\,\mathrm{km}$. In total, the considered environment becomes the union of 232,139 different clusters $C_j \in C$.

In order to get a better understanding of the data provided by the OpenSky Network, we visualize the sensor coverages and the number of processed ADS-B messages with respect to their spatial distribution. These evaluations are based on data collected from an entire day (February 15, 2020) resulting in a total of 132,883,464 messages broadcasted by real aircraft. Figure 2 depicts a heat map of the spatial distribution of all recorded ADS-B reports. As one can see, most reports originated from a few cluster areas close to central European airports. Notably, the database only contains messages that reached at least one contributing sensor.

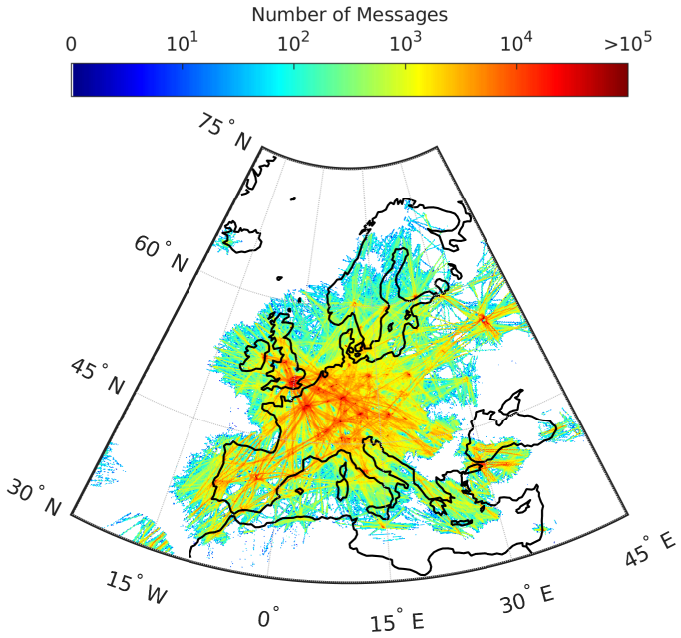

Fig. 2. Spatial distribution of captured ADS-B reports from the OpenSky Network in Europe as of February 15, 2020.

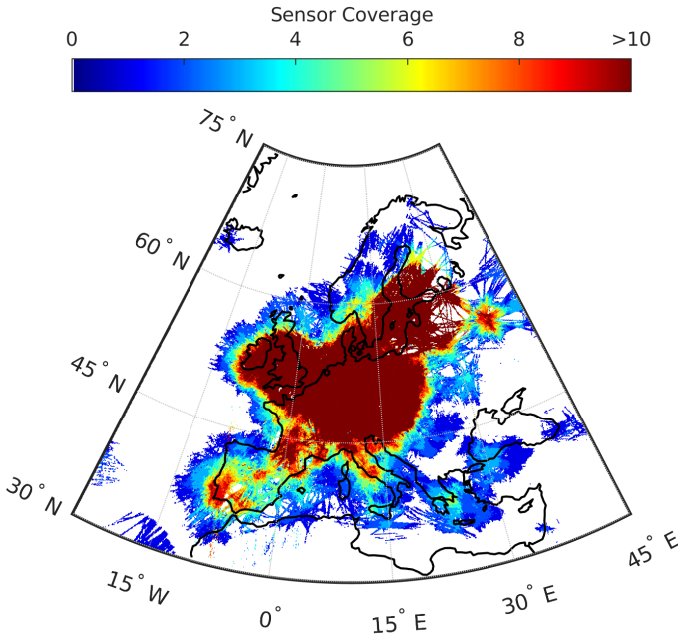

Fig. 3. The aggregated sensor coverage of the OpenSky Network with a strong dominance in Central Europe as of February 15, 2020.

The overall coverage of the network is the combination of all participating sensors. Since sensor coverages can significantly overlap with each other, the redundancy is higher in areas with more sensors as compared to rural areas. Figure 3 shows the aggregated sensor coverage of the OpenSky Network as of February 15, 2020. The heatmap depicts the number of sensors that simultaneously cover an indicated area. A total of 729 different sensors reported data for the considered airspace. We notice a strong dominance in Central Europe, where the most participating sensors are operated. Nevertheless, the coverage of the sensor network also limits the applicability of our system. Airspaces covered by no sensors are not protected.

B. Notations

For the remainder of this paper, we use the following notations. The network is formed by a set of ground-based sensors S, where each sensor is referred to as $S_i \in S$. Each ADS-B message m can be received by an arbitrary number ≥ 1 of sensors S_i, hence the link (m, S_i) exists. Due to noise effects and message collisions, message loss can naturally occur and we denote the probability that sensor S_i receives a message transmitted from cluster C_j as $P_{\text{rec}}(S_i, C_j)$. Moreover, the messages are timestamped by the receiving sensors, where t is the issued timestamp. When a message is not picked up by any sensor, it is consequently not in the considered database. Table II summarizes the used notations.

TABLE II. PARAMETER NOTATIONS

Parameter	Notation
Cluster	C
ADS-B Sensor	S
ADS-B Message	m
Time	t
Probability of Reception	$P_{rec}(S, C)$

C. ADS-B Message Trust

In order to assess the trustworthiness of ADS-B messages, we design an evaluation process consisting of four verification tests, namely (i) sanity, (ii) differential, (iii) dependency, and (iv) cross check. While the former three tests are stated for the sake of completion, we focus on the cross check that is tailored towards the existing sensor infrastructure to implement wireless witnessing. The system overview is depicted in Figure 4 and is developed in the following.

1) Sanity Check: The sanity check represents a message content verification with respect to defined value ranges. Where data values are not restricted by definition, we apply physical possibility bounds. Sanity checks are specific to the message content, i. e., the reported aircraft status. Table III provides an overview of the implemented sanity check.

Position. The reported position contains information about the latitude, longitude, and altitude. The latitude is only defined in the range of $-90°$ to $90°$, whereas the longitude is defined over $-180°$ to $180°$. The altitude is not bounded by its definition but by physical restrictions ranging from approx. $-3\,\text{m}$, which is the altitude of the lowest European

TABLE III. SANITY CHECK

Category	Parameter	Range
Position	Latitude	$-90°$ to $90°$
	Longitude	$-180°$ to $180°$
	Altitude	$-3\,\text{m}$ to $20{,}000\,\text{m}$
Movement	Velocity	$0\,\text{km/h}$ to $1{,}200\,\text{km/h}$
	True Track	$0°$ to $360°$
	Vertical Rate	$-50\,\text{m/s}$ to $50\,\text{m/s}$
Identification	ICAO Identifier	Registered Aircraft
	Call Sign	Assigned Call Signs

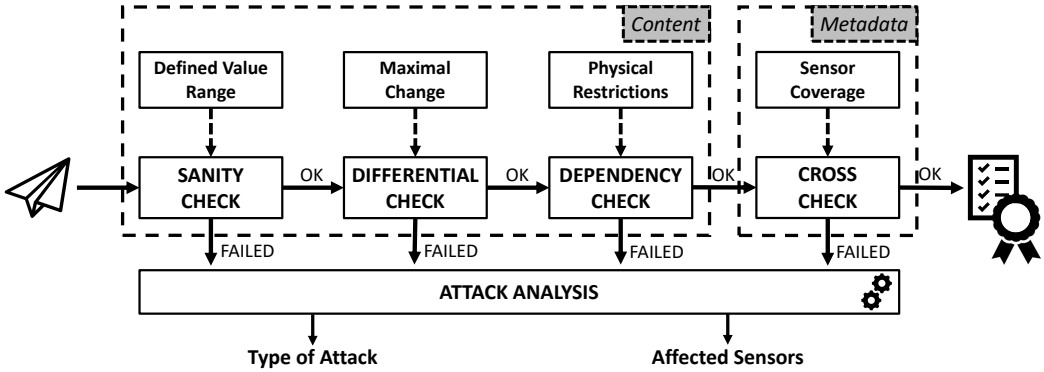

Fig. 4. The process of ADS-B trust evaluation including all four verification tests, their utilized data, and conditional branching to subsequent attack analysis, where the type of attack and the affected sensors are identified.

TABLE IV. DIFFERENTIAL CHECK

Parameter	Maximal Change per Second
Horizontal Position	$500\,\mathrm{m}$
Altitude	$100\,\mathrm{m}$
True Track	$10°$
Velocity	$25\,\mathrm{km/h}$
Vertical Rate	$10\,\mathrm{m/s}$

TABLE V. DEPENDENCY CHECK

Relationship	Tolerance
Horizontal Position ↔ Velocity + True Track	$100\,\mathrm{m}$
Altitude ↔ Vertical Rate	$100\,\mathrm{m}$
Altitude ↔ Aircraft on Ground	$1{,}707\,\mathrm{m}$

airport, Amsterdam Airport Schiphol. For the maximal altitude, we use a bound of $20{,}000\,\mathrm{m}$, which is hardly reachable for casual air traffic.

Movement. While airborne, the velocity is expected to be positive and bounded by the maximal speed of the specific aircraft type, usually less than approx. $1{,}200\,\mathrm{km/h}$. The direction of movement, referred to as the true track, is defined by the angle aligned with the True North in the range of $0°$ to $360°$. Moreover, the vertical rate is also aircraft-dependent and is expected to not exceed $\pm 50\,\mathrm{m/s}$.

Identification. Each aircraft is assigned a unique identification, the ICAO 24-bit registration identity. This identifier can be checked against databases that contain currently assigned ICAO registrations. In addition, each aircraft is assigned a volatile call sign, which can also be verified.

2) Differential Check: The differential check considers changes between succeeding ADS-B messages from the same aircraft. These checks, therefore, require the assignment of messages to tracks based on the included identifier. In consideration of the message update rate and broadcast frequency, we identify reasonable maximal changes per second that conform to the inertia and aircraft capabilities as well as covered by observations of real flight data. Table IV contains the implemented tolerable parameter changes. In cases where we receive updated ADS-B reports after a prolonged loss of communication, e. g., due to missing sensor coverage, we incorporate the lack of data by scaling the tolerable maximal change with the missed time period.

3) Dependency Check: The dependency check verifies the relationship between physically-dependent parameters of subsequent reports from the same aircraft. We validate reported horizontal and vertical changes based on predictions of the next position and allow for a tolerance up to $100\,\mathrm{m}$, which we have empirically derived from the available dataset. A further dependency exists between the reported altitude and the aircraft indicating to be on ground. We coarsely perform this check against the elevation of the highest European airport ($1{,}707\,\mathrm{m}$), Samedan Airport of Switzerland. Notably, more fine-grained information about the geographical topology would greatly benefit the validity. Table V shows the implemented dependency checks.

4) Cross Check: The cross check utilizes the spatial redundancy of the surveillance network in a collaborating manner. Participating sensors are widely distributed and their coverages overlap significantly, as shown in Figure 3. Even though the sensor locations are unknown, we can determine which sensors observe which airspace via inspecting the reported positions embedded in their received ADS-B reports. Hence, in our grid-based approach, each cluster C_j is dedicated to covering sensors S_i such that the following equation holds:

$$P_{\mathrm{rec}}(S_i, C_j) > 0. \tag{1}$$

If multiple sensors S_i cover the same cluster C_j such that $P_{\mathrm{rec}}(S_i, C_j) > 0$, we can countercheck received message by consulting all designated sensors. For each sensor that covers a reported aircraft position, we distinguish two discrete events—the sensor has *received* the message or the sensor has *not received* the message:

$$X_{m,S_i} = \begin{cases} 0 & \nexists(m, S_i) \\ 1 & \exists(m, S_i) \end{cases}. \tag{2}$$

Due to noise effects and signal collisions, sensors naturally experience a message loss in the range of $10\,\%$ to $75\,\%$ depending on the distance to the origin, obstacles in view, and the airspace density [39]. Hence, the case of missing a report does not causally imply unusual behavior or the existence of attacks and needs to be factored in accordingly. We refer to the combination of events $X_{m,S_i}, S_i \in S$ as the observed message reception pattern for a report broadcasted from the claimed

position. Each sensed message is therefore mapped to a vector representing the reception events for every sensor:

$$\vec{X}_m = \left[X_{m,S_1}, X_{m,S_2}, \cdots, X_{m,S_{n-1}}, X_{m,S_n}\right], \qquad (3)$$

where n is the total number of sensors in the network. For our considered scenario, we obtain a vector with 729 entries, which represents the message reception pattern. These patterns exhibit a certain variance and cannot be translated into fixed rules due to non-deterministic sensor reception. Hence, we choose a Machine Learning (ML) approach to handle the huge amount of available data and simultaneously consider unknown external effects.

In particular, for each of the 132,883,464 recorded ADS-B reports, we determine which of the 729 sensors reported that specific message. In combination with the embedded positioning information, we learn typical reception patterns for the entire day and label the data to be the result of normal operating air traffic and sensors. After processing all reports, each cluster C_j is assigned with actually observed message reception patterns and we assume these patterns to represent normal behavior. We discuss this assumption in Section VI-A and reason about its validity.

Algorithm Choice. Since our feature space is defined by the number of sensors and each feature is limited to either be 0 (not received) or 1 (received), we choose to use Decision Trees (DTs). This choice is in accordance with similar work classifying distributed sensor events [23], [59]. For more information on machine learning algorithms, we refer to an article by Leo Breimann [2].

D. Attack Analysis

In the case where at least one of our verification tests indicates unusual behavior, an attack analysis is triggered that tries to further reason about (i) the type of attack and (ii) the affected sensors. Depending on which test triggered the attack analysis, different conclusions can be drawn on the cause of an alarm.

1) Type of Attack: We notice that our three attack classes, i. e., GPS spoofing, ADS-B spoofing, and sensor control/Sybil attack, can be characterized by the type of manipulation they cause on the message, respectively on the network. This can either be on the content of the ADS-B messages directly, or more subtle on the message reception characteristic. While the sanity, differential, and dependency checks can verify the message payload, the cross check evaluates the reception pattern. For each attack vector, we identify which verification test is indicative and provide an overview in Table VI.

Sanity Check. The sanity check detects defined value range violations. These can occur when a report is either specifically crafted during an ADS-B spoofing attack or if a sensor is entirely under the control of an attacker.

Differential Check. The differential check is indicative to unusual jumps in the data. A GPS spoofing attack may hence be detectable if the position exhibits a sudden jump. All other attacks may also trigger an alarm depending on the variance in the generated fake data.

TABLE VI. SENSITIVITY TO ATTACKS

Attack Vector	Sanity	Differential	Dependency	Cross
GPS Spoofing	○	◐	●	●
ADS-B Spoofing (Single)	◐	◐	◐	●
ADS-B Spoofing (Multiple)	◐	◐	◐	◑
Sensor Control	◐	◐	◐	●
Sybil Attack	◐	◐	◐	◑

○ not indicative, ◐ potentially indicative
● always indicative, ◑ network dependent

Dependency Check. The dependency check detects inconsistencies between dependable data from independent sensors within the aircraft. Since a successful GPS spoofing attack only affects GPS-related sensors, other information on the movement or on the heading will likely result in a violation. Again, other attacks may also fail this test if the fake reports do not satisfy parameter dependencies.

Cross Check. The cross check tries to decide if a message reception pattern is the result of normal behavior or not. An aircraft report affected by a GPS spoofing attack indicates a wrong position and the reception pattern will likely differ from the actual reception pattern of the real location. For the other attacks, the validity of the cross check depends upon the number of benign sensors that observe the claimed aircraft position. The more sensors simultaneously cover an area, the less likely it will be that only a specific number of sensors, e. g., affected by an ADS-B spoofing attack, receive the specific message. Similar considerations apply for attackers adding sensors to the network. Unaffected sensors will not report injected messages which is eventually reflected in an unusual reception pattern. For both attack classes, reception patterns are easier to decide the more sensors are participating.

2) Affected Sensors: If we successfully detect unusual behavior and identify the type of attack, we try to also reason about the affected ADS-B sensors. We generally distinguish between passively and actively participating sensors during an attack. While we can tag all sensors that reported an untrustworthy message as potentially malicious, we are interested which sensors are indeed under the attacker's control. These compromised sensors are actively trying to disrupt the network. We, therefore, identify all sensors that report messages clearly assigned to a sensor control/Sybil attack as malicious. Their identification allows the disconnection from the network and to restore the network's integrity.

On the other hand, sensors that fell victim to an attack themselves may only be temporarily disconnected from the network. Sensors that are recognized in such a way can later be reactivated once the attack is over. The tracing of affected sensors also allows for a coarse localization of an attack. Even though sensor locations are unknown, coverages of the sensors can be determined and consequently a rough attacker position could be narrowed down.

IV. SIMULATION

While the characteristics of normally operating air traffic can be learned from the actually received ADS-B reports,

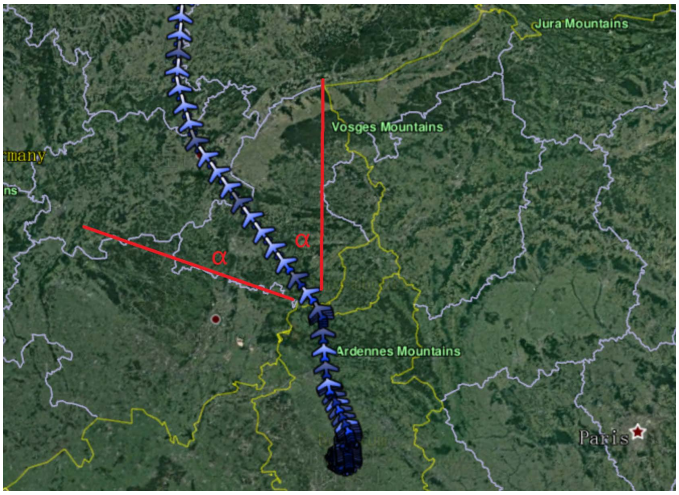

Fig. 5. Visualization of GPS spoofing: Starting at t_{attack}, we apply different deviations α in clockwise and counterclockwise direction. The generated ADS-B reports contain the spoofed positions along the red lines.

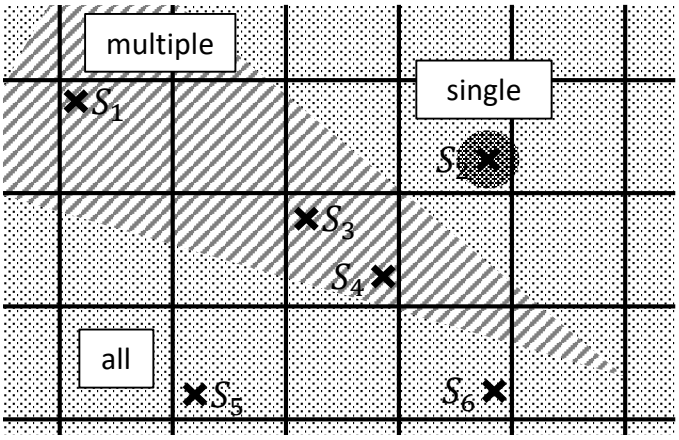

Fig. 6. Visualization of ADS-B spoofing: An attacker may follow three different strategies to inject fake reports. The attacker either affects (i) a single sensor (dark dotted area), (ii) multiple sensors (striped area), or (iii) all sensors (entire dotted area).

attack scenarios are required to be emulated based on realistic assumptions and experience. Assuming that no attacks were launched on the selected day (February 15, 2020), we use all reports to map typical reception patterns. In the following, we describe how we simulated the three considered attack classes, i. e., GPS spoofing, ADS-B spoofing, and sensor control/Sybil attack. For each attack, we generate at least the number of reports as received normally, i. e., more than 132 million different fake reports representing each respective attack. Note that this does not reflect the actual distribution between normal and attack reports, but is chosen to establish a reasonable database of fake reports. This allocation is used for the training process only.

A. GPS Spoofing

To emulate a successful GPS spoofing attack, we manipulate the reported GPS-derived positioning information embedded in ADS-B reports. More precisely, we randomly sample one ADS-B report from the entire dataset. We then gather all reports from the corresponding aircraft for the preceding $15\,\text{min}$ and the next $60\,\text{min}$ representing a $75\,\text{min}$ aircraft track. This track is then subject to selected deviations α of $1°$, $2°$, $5°$, $10°$, $20°$, or $45°$ to simulate an attack incrementally leading aircraft off their track starting at $t_{\text{attack}} = 15\,\text{min}$. Figure 5 depicts this procedure. For each deviation, we replace the GPS position in the reports while all other data fields and the sensors that received the message remain the same. We label the messages as resulting from a GPS spoofing attack after t_{attack} and also keep track of the applied deviation, the distance to the original track, and the elapsed time after the attack has been launched. We repeat this process of randomly sampling reports from the dataset and manipulating the GPS position until the desired number of reports is reached.

B. ADS-B Spoofing

When simulating an ADS-B spoofing attack, we are faced with the problem of unknown sensor locations. Even the tracing of observed clusters does not reveal a sensor position since the reception range can highly vary and may be distinct in different directions. It is noteworthy that an attacker would face the same problem and cannot pinpoint sensors but would need to blindly affect larger regions when targeting multiple sensors. We differentiate the attack according to how many sensors fall victim to the attack, i. e., a single sensor, multiple sensors, or all sensors within a selected region. Figure 6 illustrates these attacks. To simulate an attacker targeting multiple sensors, we randomly pick sensors up to the average number of observing sensors of the respective cluster.

We again generate fake messages for each scenario by randomized sampling from real-world aircraft reports. We extract the corresponding $75\,\text{min}$ long track and adjust the receiving sensors depending on the coverage of the considered cluster and how many sensors are affected by the attack. All other data fields remain the same. We use real aircraft reports to represent an attacker trying to inject authentic ghost aircraft into the network by sending those messages to the scenario-dependent number of sensors.

C. Sensor Control/Sybil Attack

In a sensor control/Sybil attack, an attacker adds sensors to the network that are under the attacker's synchronized control. We assume that the attacker's sensors initially behave normally to remain unnoticed prior to any fake message injection. When an attack is launched, all controlled sensors mutually try to report the same fake message. We again differentiate between the number of controlled sensors with regard to the number of benign sensors, i. e., a single sensor or equality between the attacker's sensors and benign sensors.

The process of sampling and selecting tracks is the same as for ADS-B spoofing. We assume that the attacker utilizes all controlled sensors to inject the same message. Notably, the benign sensors that cover the same area are not affected by a Sybil attack and will consequently not report the injection of such messages.

V. Evaluation

We split the evaluation of the developed ADS-B trust system into (i) performance of detecting each considered

attack, (ii) distinguishing between attack vectors, (iii) identifying affected sensors, (iv) analyzing the impact of different grid resolutions, (v) investigating the time dependency and (vi) estimating the computational performance.

A. Attack Detection Performance

We approach the attack detection performance in two different ways. First, we consider the classification results of single ADS-B reports without linking consecutive reports, and second, we make decisions on combined aircraft tracks. The training process uses all reports of the selected day as well as the simulated attack vectors based on randomly sampled $75\,\mathrm{min}$ long aircraft tracks from the OpenSky Network database according to Section IV. Our attack detection evaluation prototype uses clusters C_j with edge lengths of $10\,\mathrm{km}$. We assign each report to its originating cluster indicated by the embedded position splitting up all messages over the observed area. We then perform training with our selected DT classifier by iterating through all clusters.

For testing, we again query the database for 1000 untrained and randomly selected aircraft tracks. We do not make any restrictions on the selection process except that we require that at least $50\,\%$ of the broadcasted reports are actually recorded by the network. This filters tracks that would quickly leave the covered area, i. e., the scope of the network, and hence cannot be classified due to missing reports. We apply the different attack vectors, label each track accordingly, and then classify the resulting reports with the classifier for the designated cluster. For our three attack classes, i. e., GPS spoofing, ADS-B spoofing, and sensor control/Sybil attack, we shortly describe which test triggers an alarm and then focus on the ML supported cross check providing True Positive Rates (TPRs) and False Positive Rates (FPRs).

1) GPS Spoofing: While an incremental position deviation passes the differential check, our dependency check consistently indicates mismatches between predicted positions and the reported GPS position. Even though we account for a specific uncertainty threshold, at one point in time, the attack exceeds this threshold. In consideration of the cross check, the intuition is that the further away an aircraft claims to be from its real position, the more different the reception pattern will be. Notably, the selected cluster for the cross check is determined by the reported/claimed position. If the real position and the spoofed position are still within the same cluster, the reception patterns are the same and a decision towards the presence of a GPS spoofing attack is not possible.

To assess our detection performance of GPS spoofing attacks, we consider a classifier that has been trained with samples from normal operation and the simulated GPS spoofing reports. We further calculate a score based on the classifier outcome and the total number of reports. Following this metric, a score of 1 means that every report is labeled authentic while a score of 0 means that every report was labeled malicious. We evaluate (i) the average score over all 1000 runs of the classifier with respect to different deviations α from the original track and the elapsed time in Figure 7 and (ii) the average score with respect to the distance to the original track in Figure 8. The distance to the original track is a combination of the applied deviation and the time that has elapsed after the launch of the attack.

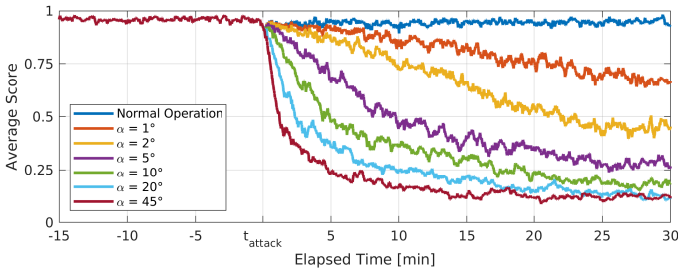

Fig. 7. When a GPS spoofing attack is launched, the classification score diverges from the normal operation score and continues to decrease over time. The rate is based on the applied deviation α and considers the average over all 1000 simulation runs.

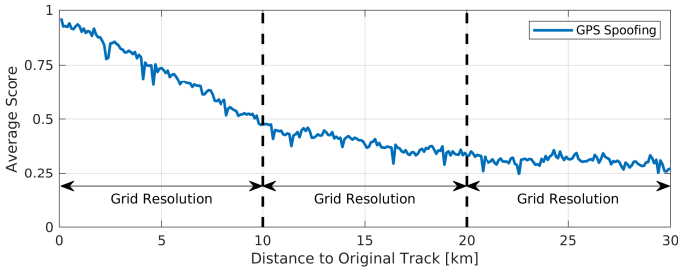

Fig. 8. The classification score under GPS spoofing decreases significantly with increasing distance to the original track. The vertical lines indicate distances in multiples of the grid resolution of $10\,\mathrm{km}$.

Results. While the dependency check is effective in detecting GPS spoofing attacks, in cases where additional information might be missing, the cross check is sufficient to detect such attacks with a high probability after a certain amount of time has passed, see Figure 7. For instance, considering $\alpha = 2°$, $\alpha = 10°$, and $\alpha = 45°$ the score falls below 0.5 after approx. $20\,\mathrm{min}$, $5\,\mathrm{min}$, and $1\,\mathrm{min}$, respectively. The rate at which the average score decreases is dominated by the applied deviation α. The higher the deviation, the faster the fake positions approach other clusters, leading to mismatches in the reception patterns. Notably, the average score, even under normal operation, never reaches 1 due to a portion of reports being wrongly classified. We will handle this problem by linking successive reports when deciding aircraft tracks.

Figure 8 condenses the deviation and the elapsed time into the distance to the original track. The average score quickly approaches 0.5 for distances up to one grid resolution, i. e., $10\,\mathrm{km}$ in our evaluation prototype. After this point has been reached, the decline slows down and reaches approx. 0.35 for a distance of two grid resolutions. Further distances only moderately decrease the average score and it nearly stabilizes at this point. We observe that the classifier can differentiate the reception patterns and perform increasingly better, the further away the spoofed track deviates from the real aircraft track. Note that in the worst case, a distance of approx. $\sqrt{2}$-times the grid resolution can still point to the same cluster. However, increasing the distance further guarantees different clusters.

We now approach the question of how we decide aircraft tracks, in contrast to the aforementioned evaluations where we showed average scores over all test runs for individual reports. Figures 7 and 8 show that the score fluctuates and that authentic reports are sometimes labeled as malicious. Even

TABLE VII. GPS SPOOFING DETECTION PERFORMANCE - FEBRUARY 15, 2020

Deviation α [°]	**Attack Detection** [%]			**Detection Delay (Median** $\pm$ **SD)** [min]			**FPR** [%]
	w = 5	w = 10	w = 15	w = 5	w = 10	w = 15	
1	64.98	75.64	**76.65**	41.63 ± 10.32	37.68 ± 10.88	$\mathbf{37.26 \pm 10.87}$	**0**
2	85.03	**90.61**	90.36	26.73 ± 10.88	$\mathbf{25.05 \pm 10.78}$	25.31 ± 10.23	**0**
5	96.19	96.45	**96.70**	16.50 ± 10.59	$\mathbf{15.14 \pm 9.57}$	16.70 ± 9.34	**0**
10	**98.73**	98.22	98.48	$\mathbf{10.97 \pm 10.11}$	11.07 ± 8.99	12.56 ± 8.52	**0**
20	**98.99**	**98.99**	98.48	$\mathbf{8.08 \pm 8.86}$	$8.81.23 \pm 8.07$	10.27 ± 7.56	**0**
45	**99.49**	**99.49**	**99.49**	$\mathbf{5.83 \pm 7.88}$	7.22 ± 7.47	8.52 ± 7.26	**0**

when no attacks are applied, we never reach a perfect score of 1. Hence, the detection of attacks cannot be based on single messages alone without triggering a high number of false alarms. Considering that we designed our system as an augmentation system for attack detection, false alarm events are disruptive and a high number is unacceptable.

To compensate for single false positives, i. e., malicious patterns detected when no attack is applied, we implement time windowing. In particular, we tested three different time windows w, i. e., 5 min, 10 min, and 15 min. The time windowing is only applied backwards such that the score at time t becomes the average score of all received reports within the last w minutes. The final decision is then based on score thresholds. With the target of minimizing false alarms, we set the threshold at the lowest score that we observed across all randomly selected 1000 aircraft tracks at any given time after t_{attack}. As a result, we achieve a false positive rate of 0 % by design with respect to the considered tracks. The selected threshold depends on the length of the time window, where shorter time windows lead to higher thresholds and larger time windows allow tighter thresholds.

In Table VII, we list the GPS spoofing detection performance considering different deviations and time windows. We analyzed the attack detection rate, i. e., the number of detected attacks compared to all tested runs and the detection delay, i. e., the time at which we observed the threshold violation and raised an alarm. We additionally state the median and the standard deviation. Bold entries mark the best results in each row. We want to highlight that for every configuration the FPR is 0 % due to how the threshold is chosen.

With increasing deviation α, the attack detection reaches up to approx. 99.5 %. An attack counts as detected when the threshold is undercut within the first hour after the launch of the attack. The missing 0.5 % that were not detected are due to very slow or even parking aircraft. The impact of GPS spoofing becomes negligible in such scenarios considering how we simulated it. The rest of the deviated aircraft tracks are detected with a very high probability. The detection delay strongly depends on the applied deviation α. For higher values, the average detection delay can go as low as approx. 6 min and standard deviations around 8 min. The time window w also impacts the performance. The implementation of different time windows is beneficial since the best attack detection rate and the detection delay is dependent on the applied deviation α.

2) ADS-B Spoofing: For the evaluation of the ADS-B spoofing detection performance, we specifically focus on the outcome of the cross check. Since an attacker is able to generate arbitrary reports, we assume that an attacker can successfully remain undetected by the sanity, differential, and

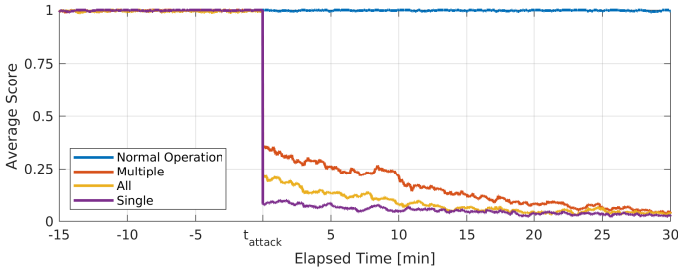

Fig. 9. As soon as the attacker starts to inject fake reports, the average score drops immediately. Affecting multiple sensors but not all is the most susceptible to misclassifications.

dependency check. Considering the testing set for the cross check, we take the same sampled aircraft tracks from the GPS spoofing evaluation but apply ADS-B spoofing according to Section IV. At time t_{attack}, the attacker launches the spoofing attack representing a scenario where an aircraft track would normally end, but is continued by fake injections into the system. We distinguish between three scenarios depending on the targeted number of sensors (see Figure 6). Notably, we use a classifier that is trained with samples from normal operation and simulated samples from ADS-B spoofing.

Results. The resulting average scores of all three scenarios are depicted in Figure 9. One can see that the score for normal operation is very close to 1, while any form of ADS-B spoofing drastically reduces the average score across all 1000 runs. This change is almost immediately after the attack has been launched and continues to decrease afterwards. Furthermore, the scenarios impact the scores differently. From an attacker's perspective, injecting reports from multiple but not from all sensors is superior to all other strategies.

We argue that even an optimized attacker strategy cannot emulate typical reception patterns by only affecting specific sensors. Since sensors are geographically distributed at unknown positions, an attacker cannot systematically control which and how many sensors receive the fake reports. Eventually, an attacker needs to broadcast from a location close to the claimed position to emulate realistic message reception patterns, virtually becoming a legitimate broadcast from the advertised position.

Even when targeting multiple sensors, constantly missing reports from sensors within the reception range is a strong indication for some kind of injection. Naturally, the number of sensors observing the cluster where the injection takes place impacts the significance. The patterns have less variations when fewer sensors are operated and the differences to malicious patterns will be less obvious. Figure 10 shows

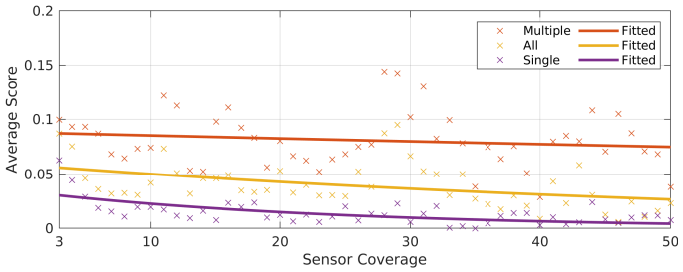

Fig. 10. The number of sensors observing the cluster of the reported position has an impact on the classification performance. We can detect a tendency towards lower scores when the sensor coverage increases.

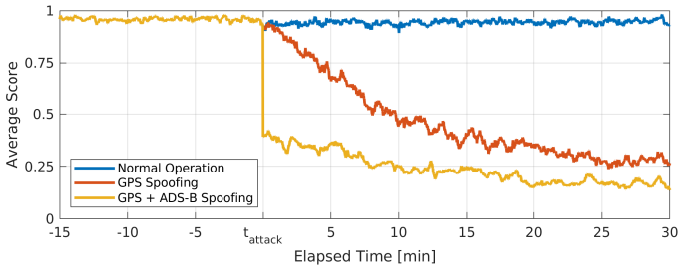

Fig. 11. The GPS spoofing classifier yields lower average scores for the combination of GPS spoofing and ADS-B spoofing. The attack parameters are set to $\alpha = 5°$ and multiple affected sensors.

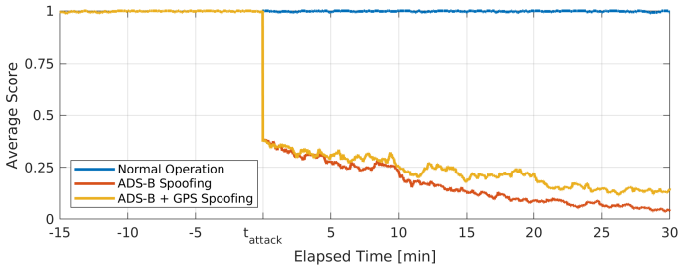

Fig. 12. The ADS-B spoofing classifier yields slightly better average scores in comparison to the combination of ADS-B spoofing and GPS spoofing. The attack parameters are set to $\alpha = 5°$ and multiple affected sensors.

the average score in relation to the number of observing sensors. Having only three sensors, the attacker can remain undetected in more cases than in clusters with a sensor coverage of 10, 30, or 50.

3) Sensor Control/Sybil Attack: To evaluate our detection performance of sensor control/Sybil attacks, we again focus on the outcome of the cross check. We consider two scenarios with different numbers of compromised sensors, i. e., a single sensor or equality between the attacker's sensors and the number of sensors already observing that specific airspace. Notably, the attackers' sensors initially participate normally and are already considered when message reception patterns are trained. After t_{attack}, the attacker starts to use the controlled sensors to inject an aircraft track. Compared to our assumptions for ADS-B spoofing, the attacker is now capable of emulating arbitrary reception patterns using all the controlled sensors while benign sensors within the same cluster remain unaffected.

Results. The results are very similar to the ADS-B spoofing results. The impact on the score is immediate and can be clearly distinguished from normal behavior. The reasoning behind the similar results are based on the benign sensors that are unaffected by the attacker. A message injection from the controlled sensors represents the very unlikely case of a high number of benign sensors missing on the same message. The detection of Sybil attacks is hence based on missing reports rather than all sensors agreeing on the same message. Figure 10 can be converted to this scenario when considering the sensor coverage of only the uncompromised sensors.

Nevertheless, some limitations need to be highlighted. If the attacker controls every sensor for one cluster, arbitrary patterns can be emulated and we have no chance of detecting the attack. However, as soon as the attacker tries to inject reports for clusters that are already observed by sensors, the attack can be detected. The vast majority of airspace is already observed by at least one sensor (see Table IX). We argue that as long as the majority of benign sensors operate normally, the attack can still be detected.

4) Combined Attacks: Thus far, we have evaluated the detection performance of individual attacks, i. e., GPS spoofing, ADS-B spoofing, and sensor control/Sybil attacks. We now analyze if any attack combination can increase the attacker's chance of remaining undetected. Notably, sensor control is superior to ADS-B spoofing since a fully compromised sensor cannot only inject any form of false ADS-B reports (as it is the case for ADS-B spoofing) but also drop any other messages the sensor may receive. Hence, ADS-B spoofing can be considered a subset of the sensor control/Sybil attack class. The success of their combination can be upper bounded by the success an attacker would have who instead also controls the sensors affected by ADS-B spoofing. While an attacker controlling a subset of sensors may still decide to additionally spoof other sensors, the detection performance is closely tied to the number of benign sensors.

We focus on reports affected by GPS spoofing and ADS-B spoofing at the same time, i. e., a fake GPS track that is injected via ADS-B spoofing. We set the deviation α to $5°$ and assume an attacker to inject the track via spoofing multiple sensors. We consider the impact on the detection performance from two different directions. Figure 11 shows the change based on a classifier that is indicative for GPS spoofing. Figure 12 depicts the other perspective, where the ADS-B spoofing classifier evaluates the attack combination.

Results. Comparing the detection performance of fake GPS spoofing reports to additional ADS-B spoofing, one can clearly notice the sudden drop in score due to the ADS-B spoofing in the combination. Over the cause of 30 min, the average score is constantly lower rendering the combination unfavorable for the attacker. Surprisingly, from the perspective of ADS-B spoofing, we can notice that the attack combination actually results in slightly higher scores and that the effect increases over time. It seems that a combination favors the attacker, however, the score differences are due to a change that is not reflected in the figure: By additionally manipulating the GPS positions, the fake track faster approaches edge areas that are observed by less sensors and hence the classification looses significance (compare Figure 10). As long as enough benign sensors are unaffected, any attack combination does not favor the attacker.

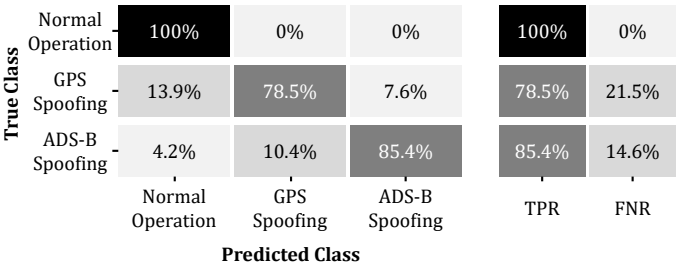

Fig. 13. The confusion matrix of our classifier deciding the type of attack when confronted with tracks representing: Normal Operation, GPS Spoofing, or ADS-B Spoofing. We set $\alpha = 20°$ in the GPS spoofing case and ADS-B spoofing affects multiple sensors.

5) From Single Reports to Moving Tracks: In our evaluation, we linked the classification results of individual reports to make a decision for an entire aircraft track. While single reports may be falsely classified as malicious, time windowing mitigates this effect. The trained models for different clusters are separated and some may be more concise than others. A fact that facilitates our detection scheme is the intrinsic movement of aircraft such that a track traverses many different clusters over its course. As a result, the combined decisions of multiple clusters benefits from clusters with higher sensor coverage, eventually yielding a very high classification performance even when clusters are involved that are hard to decide.

B. Attack Analysis: Type of Attack

So far, we have used a different classifier for each considered attack vector. The type of attack can be trivially determined by the classifier that indicated the attack. We neglected the possibility that classifiers, e. g., tailored towards GPS spoofing detection, may also raise an alarm when faced with ADS-B spoofing, and vice versa. Note that, when no attack is applied no classifier will yield any false alarm due to the way we set our thresholds. We now analyze whether we can tell attack patterns apart. In order to evaluate the ability to differentiate between our simulated attacks, we transform the binary classification into a multiclass classification that decides the type of attack. We trained a DT classifier with reports from GPS spoofing and ADS-B spoofing. Since both attacks have multiple configurations, we chose a deviation of $20°$ for GPS spoofing and multiple sensors affected for ADS-B spoofing. We apply a time windowing of $w = 15\,\mathrm{min}$ and evaluate the result at $t_{\mathrm{attack}} + 30\,\mathrm{min}$. Figure 13 depicts the confusion matrix of the classification results.

Results. Considering aircraft tracks without any attack modification applied, the combined classifier yields no false classifications. For GPS spoofing with $\alpha = 20°$, $78.5\,\%$ of the randomized runs are detected and correctly identified, while $13.9\,\%$ are still considered normal. Approx. $7.6\,\%$of the cases are assigned as ADS-B spoofing. In comparison, $85.4\,\%$ of ADS-B spoofing tracks are classified correctly, $4.2\,\%$ are decided to be normal, and $10.4\,\%$ are mixed with GPS spoofing. Our classifier struggles with this separation due to the similar impact on reception patterns in the early phases of GPS spoofing. All in all, the majority of attacks were correctly assigned and separated.

C. Attack Analysis: Affected Sensors

We generally differentiate between sensors that fell victim to an attack themselves and sensors that are actively collaborating. For instance, in a GPS or ADS-B spoofing attack, sensors may be faced with bogus input data, however, they are still functioning correctly and are otherwise conform with their intended behavior. While for GPS spoofing attacks the reception patterns reflect normal behavior—but for a different message origin as claimed, the reception patterns for ADS-B spoofing attacks are altered. When our attack analysis reveals the type of attack being of the latter case, the reporting sensors may be disconnected from the network and excluded from the cross checking procedure of other reports. These sensors are directly affected by the attack and their recordings cannot be trusted. However, once the attack is concluded, the identified sensors may be reactivated to again contribute to the network.

On the other hand, if the attack analysis reveals a sensor control/Sybil attack, we are faced with compromised sensors actively launching attacks on the network. All sensors that reported the reception of identified fake reports need to be considered as part of an attacker-controlled sensor union. Any shared reports from such sensors cannot be considered trustworthy. Their participation in the crowdsourcing network must be shut down and their forwarded reports filtered out accordingly to recover the integrity of the network.

D. Impact of Grid Resolution

The resolution of our considered underlying grid determines the process of assigning reports and sensors to cluster C_j. The higher the grid resolution, the finer is the differentiation between regions and eventually their reception patterns. However, increasing the grid resolution not only increases the computational load but can also lead to overfitting areas to the monitoring sensors. For instance, since we do not know the exact locations of sensors, we need to learn the observed area from reported ADS-B messages. The chances that a sensor did not report any message from a specific area increase with smaller sizes even though the sensor actually observes that airspace. While we chose a grid size with edge lengths of $10\,\mathrm{km}$ to compare the attack detection performance, we also evaluated the impact of different grid resolutions and gained the following insights.

Results. The greater the proliferation of a cluster is, the more sensors are potentially observing at least parts of the area. As a consequence, the reception patterns feature more active sensors and have a higher variance within the same cluster. However, this also makes it harder to have a clear distinction between normal operation and malicious patterns. On the other hand, clusters with very tight areas actually prevent the estimation of meaningful reception patterns and thus also decrease the validity. Since the attack detection performance is related to the differences in the reception patterns, we determined a reasonable trade-off between sensitivity and generalization, which resulted in the grid resolution of $10\,\mathrm{km}$.

E. Time Dependency

To evaluate the time dependency of our detection scheme, we additionally assess its performance on a dataset gathered for February 17, 2020. This dataset represents a normal weekday,

TABLE VIII. GPS SPOOFING DETECTION PERFORMANCE - FEBRUARY 17, 2020

Deviation α [°]	**Attack Detection** [%]			**Detection Delay (Median $\pm$ SD)** [min]			**FPR** [%]
	w = 5	w = 10	w = 15	w = 5	w = 10	w = 15	
1	79, 51	86.83	**91.71**	39.33 ± 10.08	34.27 ± 10.08	$\mathbf{27.93 \pm 10.30}$	**0**
2	90.73	93.66	**94.63**	21.55 ± 9.47	20.45 ± 9.91	$\mathbf{18.65 \pm 8.80}$	**0**
5	97.07	97.07	**98.04**	12.63 ± 9.46	$\mathbf{11.92 \pm 8.80}$	12.00 ± 8.30	**0**
10	**99.02**	98.54	**99.02**	$\mathbf{8.17 \pm 9.23}$	9.33 ± 7.83	9.68 ± 7.72	**0**
20	**99.51**	99.02	99.02	$\mathbf{6.28 \pm 9.04}$	7.50 ± 7.10	7.68 ± 6.86	**0**
45	**100**	**100**	**100**	$\mathbf{5.15 \pm 7.15}$	6.53 ± 7.04	6.88 ± 6.91	**0**

two days after the previously analyzed day. This day was chosen due to a temperature drop and rainy weather and thus represents unfavorable conditions. The number and paths of flights on this new day is similar (but not identical) to the previously selected dataset. During this day, the OpenSky Network recorded over 135 million ADS-B reports and 728 active sensors. The structure of the sensor network on both days is strongly overlapping showing very minor fluctuations. The evaluations steps are kept the same to our previous analysis, revealing the following results.

Results. Overall, the results show very little deviations from the previous results and the extent of variation is comparable to the homogeneity of the sensor network. In particular, we present results showing the detection performance considering GPS spoofing attacks in Table VIII. The results for both ADS-B spoofing and sensor control/Sybil attacks are overlapping with the prior results such that differences cannot be captured visually, hence we abstain from presenting identical figures. All in all, this provides evidence which suggests that (i) different flight paths, (ii) varying airspace density, and (iii) changing weather conditions only slightly influence the detection performance of our scheme, indicating its robustness against these parameters.

F. Computational Performance

The implementation of the ML-based cross check imposed the challenge of handling more than 132 million reports from more than 700 sensors, just for a single day and only in Europe. With this massive amount of data, training on the entire dataset became infeasible on off-the-shelf equipment. To bring down the required time for training and classification, we decided to split the data into grids, where the data in each grid can be processed separately. Moreover, the training duration is a one-time cost and was well doable on standard hardware. If implemented on a designated server, the required time is expected to be lowered by magnitudes. As a result, even retraining on a regular basis becomes possible. The recurring costs of classifications, on the other hand, are only a minor fraction of the training duration such that all classifications for an entire day only took a few minutes and can thus be performed efficiently in real-time.

VI. DISCUSSION

We discuss important properties of our developed system: (i) implicit trust in the data source, (ii) limitations, (iii) attacker's knowledge, (iv) false alarm events, (v) the current attack resilience, (vi) optimized sensor deployment, and (vii) further extensions.

A. Implicit Data Source Trust

We base the evaluation of our trust system on data provided by the OpenSky Network, which records real-world air-traffic reports. However, we take the data "as is" and consider it to represent normal behavior. We cannot exclude the existence of erroneous data or even reports that resulted from some kind of attack. Nevertheless, we thoroughly analyzed the reports of our selected day (February 15, 2020) without any findings. While our system is designed to analyze live data, it can also be used to find unusual events and potential attacks in the recorded air-traffic reports in a retrospective view.

B. Limitations

While we state that our system can detect all considered attacks (i. e., GPS spoofing, ADS-B spoofing, and sensor control/Sybil attack), our system is subject to limitations. Independent of the attack, any verification can only be applied in covered airspaces (see Figure 3) which excludes, e. g., the open sea. For the cross check, we further require at least three sensors to yield meaningful results. Given these requirements, we achieved detection delays on the order of minutes, which is a limiting factor in situations where fast reactions are required. We tuned our system towards minimal false alarm events requiring us to delay decision. Allowing the existence of false alarms can significantly lower this delay.

Some limitations are specific to the types of attacks as we explain as follows:

1) GPS Spoofing: The limitations of GPS spoofing detection are based on the extent of applied deviation and the grid resolution. With finer grid resolution, the more subtle deviations can be detected. However, the resolution can only be increased to a certain degree. Based on our simulations, a resolution of 10 km was identified as a good choice. Fixing the grid resolution to 10 km, we consider our system to reliably detect more than 96 % of GPS spoofing attacks with a deviation of at least 5°. Less deviation can only be detected with lower probability or after significantly more time.

2) ADS-B Spoofing: When facing an ADS-B spoofing attack, the detection capability of our system requires the positions of sensors to remain concealed such that an attacker cannot selectively target individual sensors with, e. g., multiple antennas. If an attacker can pinpoint sensors to emulate realistic reception patterns, our system would not be able to detect malicious injections.

3) Sensor Control/Sybil Attack: Naturally, an attacker controlling every sensor could overcome any verification scheme due to full control over reported data. Our detection system relies on the existence of benign sensors. In an area with active

malicious sensors, we require at least three benign sensors to be able to detect the attack. Notably, we do not consider any form of identity spoofing, in which reports are injected with sensor identities without any control over the indicated sensors. This must be prevented on other layers.

In circumstances that stay within these limitations, our detection scheme achieves the stated performance figures. Outside the limitations, the performance may be heavily degraded. Fortunately, areas where the number of sensors is a limitation are constantly shrinking due to increasing sensor coverage (see Section VI-E).

C. Attacker's Knowledge

In our performance analysis of detecting ADS-B spoofing and Sybil attacks, we considered attackers controlling a certain number of sensors. An attacker with full awareness of our system might try to optimize the pursued attack strategy and imitate authentic reception patterns. For both ADS-B spoofing and Sybil attacks, this can only be achieved to a certain degree and we argue that an attacker cannot overcome the detection scheme in regions with enough sensor redundancy. Even a fully aware attacker does not know the exact locations of other sensors, and hence it is not possible to manipulate them in a targeted manner (e. g., through ADS-B spoofing). Moreover, an attacker cannot access the unprocessed readings of other sensors in an effort to localize them. In the case of ADS-B spoofing, where an attacker affects multiple sensors, the actual victims cannot be targeted separately. In the case of a Sybil attack, the attacker could try to emulate realistic reception patterns using the controlled sensors, but cannot do so with the sound user-operated sensors. The better a cluster is covered by benign sensors, the more conspicuous an attack will be. We, therefore, argue that even an attacker, fully aware of our system, cannot overcome the detection scheme due to the concealed locations of other sensors.

D. False Alarm Events

We acknowledge that any false alarm event, i. e., a falsely detected attack, greatly hinders the acceptance of our developed system. Especially when considering safety-related air-traffic surveillance, false alarm events would distract air-traffic controllers leading to the opposite of what we wanted to achieve. With our choice of setting thresholds, we obtained $0\,\%$ false positives over a dataset of 1000 randomly sampled tracks. Admittedly, this does not guarantee the absent of false alarms. However, our system can be tuned with updated thresholds and time windows if false alarms arises. Even for broader thresholds, we expect meaningful attack detection rates within reasonable delays.

E. Current Attack Resilience

The crowdsourcing sensors are at the core of our trust system and their distribution and density are of utter importance for the detection of attacks. The validity of the cross check, i. e., wireless witnessing, increases with the number of sensors covering the same air segments. Thus, the more redundancy, the more variations exist in the reception patterns and the better malicious attacks and sensors can be detected. We analyzed the current resilience of the OpenSky Network by considering regions related to different coverages. Table IX states the breakdown of the total covered area and relates it to the total surface of the European continent.

TABLE IX. COVERAGE REGIONS - FEBRUARY 15, 2020

Coverage	≥ 3	≥ 5	≥ 10	≥ 20	≥ 50
Area [km^2]	6,449,000	4,842,500	3,115,400	1,970,700	659,200
Total [%]	63.35	47.59	30.60	19.36	6.48

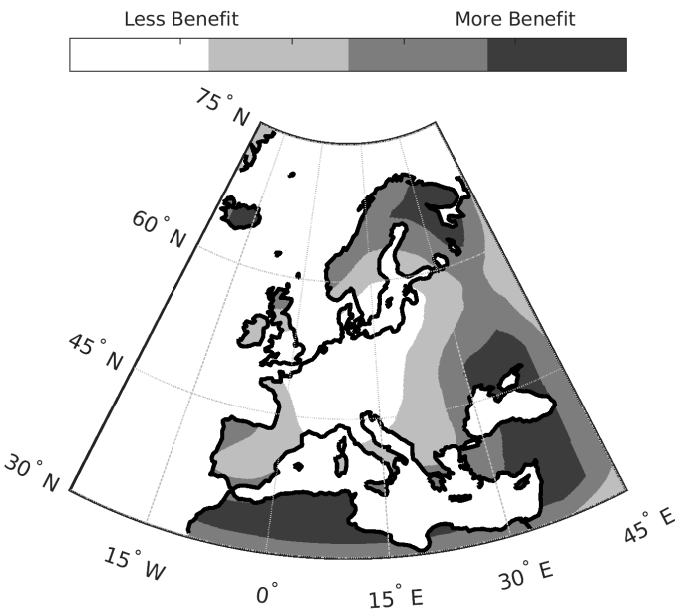

Fig. 14. The optimized deployment of new sensors identifies regions that benefit the most from better coverage. We consider the resilience increase with respect to the entire network, where darker colors indicate higher benefits.

F. Optimizing Sensor Deployment

To further develop the security of the network, we encourage the deployment of new sensors in less covered areas to optimize the current geographical distribution by optimized network expansion. Based on the coverage information of the existing sensors in the network (see Figure 3), we optimize the placement of new sensors with the goal of filling blind spots. Our optimization target is an overall coverage increase and therefore a hardening against attacks.

To provide an overview of areas that would benefit the most from the deployment of new sensors, we weight the need for better coverage according to the current sensor redundancy of the network. The lower the coverage, the higher is the demand for new sensors. We restrict possible locations to be on land. We further assume an average reception range of $400\,\mathrm{km}$ and simplify the observable airspace to be a circle around the sensor. Figure 14 depicts areas according to their coverage increase for the entire network. While in Central Europe the deployment of new sensors does not significantly impact the overall resilience against attacks, new sensor setups close to the coastlines can greatly increase the attack resilience.

G. Extensions

We discuss three extensions of our trust system with the goal of better reflecting real-world characteristics as well as

introducing sensor reputation to weight their impact on the trust assessment process. Further, dynamic learning strategies can keep attack detection strategies updated.

Time Dependence. Since ADS-B broadcasts use the wireless medium, message collisions can occur when the frequency band is saturated. The resulting rate of message loss is dependent on the airspace density which in turn changes over time based on the operating hours of airports. The more aircraft share the same medium, the higher the chances are of messages being lost. While our current system estimates reception probabilities based on averaged one-day observations, a future extension of our trust system may account for time-dependent message loss.

Sensor Reputation. In the currently deployed crowdsourcing network, we consider each sensor as equivalent to any other sensor. To refine this assumption, sensors may be assigned a reputation rating. A portion of the sensors are operated by personal contacts or registered users. Those sensors are expected to be less likely to participate in active attacks and we could link the reputation of the operator to possessed sensors. Furthermore, the hardware implementation could also be taken into account, where some implementations are more robust to defects than others. By incorporating sensor reputation, the validity of telling normal behavior and attack scenarios apart could be further improved.

Dynamic Learning. Finally, we envision the implementation of dynamic learning techniques. A dynamic learning approach could constantly update the trained message reception patterns. This allows to incorporate shifts which can occur when, e. g., sensors are joining or leaving the network, the reception range of sensors changes, or transmission ranges are altered. Moreover, new attack vectors may arise in the future. A (re-)training of our classifiers with updated attack vector definitions ensures that the trust evaluation process keeps its validity when facing currently unknown attacks.

VII. Related Work

This paper is partly based on the work by Raya et al. [32] who were the first to propose a framework for *data-centric* trust establishment with a focus on short-lived associations in volatile environments and on resulting work approaching distributed sensor events [23], [59]. While our proposal for trust establishment specifically targets ADS-B based air-traffic surveillance, similar trust requirements exist in Vehicular Ad Hoc Networks (VANETs) or industrial wireless sensor networks. While Petit et al. [29] discuss detection systems for VANETs based on dynamic thresholds, Ruj et al. [34] focus on validating message consistency to identify misbehavior. Whereas Sun et al. [51] present a trust framework to detect faulty data in VANETs, Hundman et al. [17] apply similar data verification schemes for spacecraft. Dästner et al. [8] classify military aircraft based on their ADS-B report trace. Wang et al. [55] analyzes the feasibility of false data filtering in general sensor networks and Henningsen et al. [13] especially focus on industrial networks. In comparison, our system is tailored towards a network of geographically distributed sensors.

While in practice still vulnerable, the insecurity of ADS-B has long been highlighted from an academic perspective. Purton et al. [31] analyzed critical information flows and focused primarily on technical solutions. They applied a qualitative assessment method [56] that identified potential shortcomings. In contrast, McCallie et al. [24] applied a risk analysis to assess the impact of different attack vectors and recommended solutions to be incorporated into the ADS-B implementation plan. Moreover, Strohmeier et al. [44], [48] provide an overview of system-inherent problems and illustrate the security challenges of ADS-B in future air-traffic monitoring. Smith et al. [43] empirically analyze pilots' reactions to wireless attacks on avionic systems and show that undetected attacks can lead to dangerous distractions. There are several open attack vectors that, from a scientific perspective, would allow attacking ADS-B on different levels. Chevrot et al. [3] present a framework for arbitrary false data injection and outline detection strategies. Nevertheless, we must always consider the necessary effort for an attack and its feasibility in a real-world scenario.

Moser et al. [25] take a perspective on the feasibility of attacking ADS-B communication and consider an attacker using a multi-device setup. Recent work showed that such strong adversaries become increasingly realistic [18]. Furthermore, Costin and Francillon [5] demonstrated that the step from a scientific attack concept to a real attack is not necessarily too wide and managed to inject fake aircraft messages into live surveillance monitors. Later, Schäfer et al. [36] experimentally analyzed the practicability of known threats revealing startling results. In particular, aircraft instrument landing systems are prone to wireless attacks [35]. Besides these works, which all focus on aviation applications, Balduzzi et al. [1] proved that also maritime traffic via Automatic Identification System (AIS) broadcast messages can be the target of successful attacks. While the physical constraints of vehicles differ a lot, the similarity of communication channels helps to map well-known attacks to this new context.

Besides the large body of offensive work, defensive proposals exist in recent research. Strohmeier et al. [46], [49] survey the existing research on countermeasures. More specifically, Ghose and Lazos [10] as well as Schäfer et al. [37], [38] and Liu et al. [22] propose the usage of timing or Doppler-shift characteristics to detect attacks on ADS-B. While this cannot protect from attacks, it still helps to identify malicious or inaccurate messages. Other location verification schemes and anomaly detection methods are based on RADAR observations [30], statistical tests [45], or PHY layer information [60]. Habler and Shabtai [12] use flight route modelling and anomaly detection to identify malicious ADS-B messages, achieving a false alarm rate of $4.5\,\%$. Similar false alarm rates are achieved by Naganawa et al. [26] based on Angle of Arrival (AoA) measurements. Sun et al. [52] also use AoA verification but with a single receiver.

First results based on cross-referencing within a distributed sensor network are illustrated by Strohmeier et al. [50]. Oligeri et al. [27] use IRIDIUM signals to validate GNSS position solutions. While Wesson et al. [57] discuss solutions based on cryptography, Kim et al. [21] evaluate a solution based on protocol extension with timestamps. Our system, on the other hand, requires no additional measurement information different from already collected data and can thus be implemented without any modifications.

Aside from ADS-B and AIS, the insecurity of GPS has been repeatedly demonstrated, while Humphreys et al. [16] were the first to publish an attack on GPS, where they managed to spoof GPS signals. Tippenhauer et al. [53] later analyzed the requirements of successful GPS spoofing attacks and reasoned about possible attacker positions when facing a specific sensor deployment. Zeng et al. [62] demonstrate the insecurity of road navigation systems via a stealthy manipulation based on GPS spoofing. Considering multiple sensors, countermeasures exist for the detection of GPS spoofing attacks [20], [58], [61] and also for spoofer localization [19], [61]. However, these countermeasures depend on ground-based sensors and do not exploit the network volatility. This limits the impact and consequences to a fraction of real-world use cases.

Overall, we experience a gap between scientifically proposed defenses and deployed countermeasures. As a consequence, protecting ADS-B is an open challenge that demands scientific advances to consider the requirements and limitations of the real world.

VIII. Conclusion

This work approached a trust evaluation system for ADS-B based air-traffic surveillance using an already existing infrastructure of crowdsourcing sensors. We demonstrated how our solution leverages sensor redundancy to establish wireless witnessing to protect an otherwise unsecured open system. To this end, we tested our system against prominent attack vectors showing that we cannot only detect them but also draw conclusions about their type and the participating sensors. The validity of our trust evaluation depends on the redundancy of sensors observing same airspace segments. Moreover, we outlined considerations for future sensor deployment hardening the network's security by optimized expansions.

Acknowledgment

This work was supported by the Center for Cyber Security at New York University Abu Dhabi (NYUAD).

References

[1] M. Balduzzi, A. Pasta, and K. Wilhoit, "A Security Evaluation of AIS Automated Identification System," in *Annual Computer Security Applications Conference*, ser. ACSAC '14. New Orleans, LA, USA: ACM, Dec. 2014, pp. 436–445.

[2] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[3] A. Chevrot, A. Vernotte, A. Cretin, F. Peureux, and B. Legeard, "Improved Testing of AI-based Anomaly Detection Systems using Altered Surveillance Data," in *OpenSky Symposium*, ser. OpenSky '20. Brussels, Belgium: MDPI, Nov. 2020.

[4] P. Cooper, "Aviation Cybersecurity–Finding Lift, Minimizing Drag," Atlantic Council, Tech. Rep., Nov. 2017, underwritten by Thales.

[5] A. Costin and A. Francillon, "Ghost in the Air(Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," Black Hat USA, Tech. Rep., Jul. 2012.

[6] crescentvenus, "WALB (Wireless Attack Launch Box)," 2017. [Online]. Available: https://github.com/crescentvenus/WALB

[7] J. R. Douceur, "The Sybil Attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. Cambridge, MA, USA: Springer, Jan. 2002, pp. 251–260.

[8] K. Dästner, S. Brunessaux, E. Schmid, B. von Haßler zu Roseneckh-Köhler, and O. Felix, "Classification of Military Aircraft in Real-time Radar Systems based on Supervised Machine Learning with Labelled ADS-B Data," in *Sensor Data Fusion: Trends, Solutions, Applications*, ser. SDF '18. Bonn, Germany: IEEE, Oct. 2018.

[9] Ettus Research, "Universal Software Radio Peripheral (USRP)," 2017. [Online]. Available: https://www.ettus.com

[10] N. Ghose and L. Lazos, "Verifying ADS-B Navigation Information Through Doppler Shift Measurements," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '15. Prague, Czech Republic: IEEE, Sep. 2015.

[11] A. Greenberg, "Next-Gen Air Traffic Control Vulnerable To Hackers Spoofing Planes Out Of Thin Air," Jul. 2012. [Online]. Available: https://www.forbes.com/sites/andygreenberg/2012/07/25/next-gen-air-traffic-control-vulnerable-..to-hackers-spoofing-planes-out-of-thin-air

[12] E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Computers & Security*, vol. 78, pp. 155–173, Sep. 2018.

[13] S. Henningsen, S. Dietzel, and B. Scheuermann, "Misbehavior Detection in Industrial Wireless Networks: Challenges and Directions," *Mobile Networks and Applications*, Apr. 2018.

[14] T. E. Humphreys, "Statement on the Vulnerability of Civil Unmanned Aerial Vehicles and Other Systems to Civil GPS Spoofing," The University of Texas at Austin, Tech. Rep., Jul. 2012, submitted to the Subcommittee on Oversight, Investigations, and Management of the House Committee on Homeland Security.

[15] ——, "Statement on the Security Threat Posed by Unmanned Aerial Systems and Posssible Countermeasures," The University of Texas at Austin, Tech. Rep., Mar. 2015, submitted to the Subcommittee on Oversight and Management Efficiency of the House Committee on Homeland Security.

[16] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, Jr., "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," in *International Technical Meeting of The Satellite Division of the Institute of Navigation*, ser. ION GNSS '08, Savannah, GA, USA, Sep. 2008, pp. 2314–2325.

[17] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18. London, United Kingdom: ACM, Aug. 2018, pp. 387–395.

[18] K. Jansen and C. Pöpper, "Opinion: Advancing Attacker Models of Satellite-based Localization Systems—The Case of Multi-device Attackers," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '17. Boston, MA, USA: ACM, Jul. 2017, pp. 156–159.

[19] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks," in *IEEE Symposium on Security and Privacy*, ser. SP '18. San Francisco, CA, USA: IEEE, May 2018, pp. 1018–1031.

[20] K. Jansen, N. O. Tippenhauer, and C. Pöpper, "Multi-Receiver GPS Spoofing Detection: Error Models and Realization," in *Annual Computer Security Applications Conference*, ser. ACSAC '16. Los Angeles, CA, USA: ACM, Dec. 2016, pp. 237–250.

[21] Y. Kim, J.-Y. Jo, and S. Lee, "A Secure Location Verification Method for ADS-B," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '16. Sacramento, CA, USA: IEEE, Sep. 2016.

[22] Y. Liu, J. Wang, S. Niu, and H. Song, "Deep Learning Enabled Reliable Identity Verification and Spoofing Detection," in *International Conference on Wireless Algorithms, Systems, and Applications*, ser. WASA '20. Qingdao, China: Springer, Sep. 2020, pp. 333–345.

[23] M. R. Manesh, M. S. Velashani, E. Ghribi, and N. Kaabouch, "Performance Comparison of Machine Learning Algorithms in Detecting Jamming Attacks on ADS-B Devices," in *IEEE International Conference on Electro Information Technology*, ser. EIR '19. Brookings, SD, USA: IEEE, May 2019, pp. 200–206.

[24] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *Inter-*

national Journal of Critical Infrastructure Protection, vol. 4, no. 2, pp. 78–87, Aug. 2011.

[25] D. Moser, P. Leu, L. Vincent, A. Ranganathan, F. Ricciato, and S. Čapkun, "Investigation of Multi-device Location Spoofing Attacks on Air Traffic Control and Possible Countermeasures," in *ACM Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, USA: ACM, Oct. 2016.

[26] J. Naganawa, H. Tajima, H. Miyazaki, T. Koga, and C. Chomel, "ADS-B Anti-Spoofing Performance of Monopulse Technique with Sector Antennas," in *IEEE Conference on Antenna Measurements & Applications*, ser. CAMA '17. Tsukuba, Japan: IEEE, Dec. 2017, pp. 87–90.

[27] G. Oligeri, S. Sciancalepore, and R. Di Pietro, "GNSS Spoofing Detection via Opportunistic IRIDIUM Signals," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '20. Linz, Austria: ACM, Jul. 2020, pp. 42–52.

[28] osqzss, "Software-Defined GPS Signal Simulator," 2017. [Online]. Available: https://github.com/osqzss/gps-sdr-sim

[29] J. Petit, M. Feiri, and F. Kargl, "Spoofed Data Detection in VANETs using Dynamic Thresholds," in *IEEE Vehicular Networking Conference*, ser. VNC '11. Amsterdam, Netherlands: IEEE, Nov. 2011, pp. 25–32.

[30] K. Pourvoyeur and R. Heidger, "Secure ADS-B Usage in ATC Tracking," in *Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, ser. TIWDC/ESAV '14. Rome, Italy: IEEE, Sep. 2014, pp. 35–40.

[31] L. Purton, H. Abbass, and S. Alam, "Identification of ADS-B System Vulnerabilities and Threats," in *Australasian Transport Research Forum*, ser. ATRF '10, Canberra, Australia, Sep. 2010.

[32] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks," in *IEEE Conference on Computer Communications*, ser. INFOCOM '08. Phoenix, AZ, USA: IEEE, Apr. 2008, pp. 1912–1920.

[33] RTL-SDR, "RTL-SDR (RTL2832U) and software defined radio news and projects. Also featuring Airspy, HackRF, FCD, SDRplay and more." 2017. [Online]. Available: https://www.rtl-sdr.com/

[34] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On Data-Centric Misbehavior Detection in VANETs," in *IEEE Vehicular Technology Conference*, ser. VNC Fall '11. San Francisco, CA, USA: IEEE, Sep. 2011.

[35] H. Sathaye, D. Schepers, A. Ranganathan, and G. Noubir, "Wireless Attacks on Aircraft Instrument Landing Systems," in *USENIX Security Symposium*, ser. USENIX '19. Santa Clara, CA, USA: USENIX, Aug. 2019, pp. 357–372.

[36] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next Generation Air Traffic Communication," in *International Conference on Applied Cryptography and Network Security*, ser. ACNS '13. Banff, Alberta, Canada: Springer, Jun. 2013, pp. 253–271.

[37] M. Schäfer, V. Lenders, and J. Schmitt, "Secure Track Verification," in *IEEE Symposium on Security and Privacy*, ser. SP '15. San Jose, CA, USA: IEEE, May 2015, pp. 199–213.

[38] M. Schäfer, P. Leu, V. Lenders, and J. Schmitt, "Secure Motion Verification using the Doppler Effect," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '16. Darmstadt, Germany: ACM, Jul. 2016, pp. 135–145.

[39] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A Large-scale ADS-B Sensor Network for Research," in *ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '14. Berlin, Germany: IEEE, Apr. 2014, pp. 83–94.

[40] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, V. Lenders, M. Liechti, and I. Martinovic, "OpenSky Report 2017: Mode S and ADS-B Usage of Military and other State Aircraft," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '17. St. Petersburg, FL, USA: IEEE, Sep. 2017.

[41] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, V. Lenders, and I. Martinovic, "OpenSky Report 2018: Assessing the Integrity of Crowdsourced Mode S and ADS-B Data," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '18. London, United Kingdom: IEEE, Sep. 2018.

[42] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, R. Pinheiro, V. Lenders, and I. Martinovic, "OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '16. Sacramento, CA, USA: IEEE, Sep. 2016.

[43] M. Smith, M. Strohmeier, J. Harman, V. Lenders, and I. Martinovic, "A View from the Cockpit: Exploring Pilot Reactions to Attacks on Avionic Systems," in *Network and Distributed System Security Symposium*, ser. NDSS '20. San Diego, CA, USA: Internet Society, Feb. 2020.

[44] M. Strohmeier, V. Lenders, and I. Martinovic, "On the Security of the Automatic Dependent Surveillance-Broadcast Protocol," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1066–1087, Oct. 2014.

[45] ——, "Lightweight Location Verification in Air Traffic Surveillance Networks," in *ACM Cyber-Physical System Security Workshop*, ser. CPSS '15. Singapore, Republic of Singapore: ACM, Apr. 2015, pp. 49–60.

[46] M. Strohmeier, I. Martinovic, and V. Lenders, "Securing the Air–Ground Link in Aviation," in *The Security of Critical Infrastructures*. Springer, May 2020, pp. 131–154.

[47] M. Strohmeier, M. Schäfer, M. Fuchs, V. Lenders, and I. Martinovic, "OpenSky: A Swiss Army Knife for Air Traffic Security Research," in *IEEE/AIAA Digital Avionics Systems Conference*, ser. DASC '15. Prague, Czech Republic: IEEE, Sep. 2015.

[48] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 111–118, May 2014.

[49] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic, "On Perception and Reality in Wireless Air Traffic Communications Security," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1338–1357, Jun. 2017.

[50] M. Strohmeier, M. Smith, M. Schäfer, V. Lenders, and I. Martinovic, "Crowdsourcing Security for Wireless Air Traffic Communications," in *International Conference on Cyber Conflict*, ser. CyCon '17. Tallinn, Estonia: IEEE, May 2017.

[51] M. Sun, M. Li, and R. Gerdes, "A Data Trust Framework for VANETs Enabling False Data Detection and Secure Vehicle Tracking," in *IEEE Conference on Communications and Network Security*, ser. CNS '17. Las Vegas, NV, USA: IEEE, Oct. 2017.

[52] M. Sun, Y. Man, M. Li, and R. Gerdes, "SVM: Secure Vehicle Motion Verification with a Single Wireless Receiver," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '20. Linz, Austria: ACM, Jul. 2020, pp. 65–76.

[53] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Čapkun, "On the Requirements for Successful GPS Spoofing Attacks," in *ACM Conference on Computer and Communications Security*, ser. CCS '11. Chicago, IL, USA: ACM, Oct. 2011, pp. 75–86.

[54] *Automatic Dependent Surveillance-Broadcast (ADS-B) Out Performance Requirements To Support Air Traffic Control (ATC) Service; Final Rule*, United States Department of Transportation - Federal Aviation Administration, Feb. 2010.

[55] J. Wang, Z. Liu, S. Zhang, and X. Zhang, "Defending Collaborative False Data Injection Attacks in Wireless Sensor Networks," *Information Sciences*, vol. 254, pp. 39–53, Jan. 2014.

[56] H. Weihrich, "The TOWS Matrix - A Tool for Situational Analysis," *Long Range Planning*, vol. 15, no. 2, pp. 54–66, 1982.

[57] K. D. Wesson, T. E. Humphreys, and B. L. Evan, "Can Cryptography Secure Next Generation Air Traffic Surveillance?" The University of Texas at Austin, Tech. Rep., Mar. 2014.

[58] N. Xue, L. Niu, X. Hong, Z. Li, L. Hoffaeller, and C. Pöpper, "DeepSIM: GPS Spoofing Detection on UAVs using Satellite Imagery Matching," in *Annual Computer Security Applications Conference*, ser. ACSAC '20. ACM, Dec. 2020, pp. 304–319. [Online]. Available: https://doi.org/10.1145/3427228.3427254

[59] H. Yang, S. Fong, G. Sun, and R. Wong, "A Very Fast Decision Tree Algorithm for Real-Time Data Mining of Imperfect Data Streams in a Distributed Wireless Sensor Network," *International Journal of Distributed Sensor Networks*, vol. 8, no. 12, Dec. 2012.

[60] X. Ying, J. Mazer, G. Bernieri, M. Conti, L. Bushnell, and R. Poovendran, "Detecting ADS-B Spoofing Attacks using Deep Neural Net-

works," in *IEEE Conference on Communications and Network Security*, ser. CNS '19. Washington, D.C., USA: IEEE, Jun. 2019, pp. 187–195.

[61] D.-Y. Yu, A. Ranganathan, T. Locher, S. Čapkun, and D. Basin, "Short Paper: Detection of GPS Spoofing Attacks in Power Grids," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '14. Oxford, United Kingdom: ACM, Jul. 2014, pp. 99–104.

[62] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems," in *USENIX Security Symposium*, ser. USENIX '18. Baltimore, MD, USA: USENIX, Aug. 2018, pp. 1527–1544.

[63] K. Zetter, "Air Traffic Controllers Pick the Wrong Week to Quit Using Radar," Jul. 2012. [Online]. Available: https://www.wired.com/2012/07/adsb-spoofing/