# OpenScope-sec: An ADS-B Simulator to Support the Security Research

Riccardo Cestaro
riccardo.cestaro.1@studenti.unipd.it
Department of Mathematics
University of Padua
Padua, Italy

Mauro Conti
conti@math.unipd.it
Department of Mathematics
University of Padua
Padua, Italy

Eleonora Mancini
eleonora.mancini@studenti.unipd.it
Department of Mathematics
University of Padua
Padua, Italy

Federico Turrin
turrin@math.unipd.it
Department of Mathematics
University of Padua
Padua, Italy

## ABSTRACT

Automatic Dependent Surveillance–Broadcast (ADS-B) protocol is employed in air-ground communication systems to replace legacy radar-based air traffic control systems. However, despite being a recent technology, ADS-B communication does not include security measures. This exposes the communication to potential threats, including message spoofing or fake aircraft generation. To cope with such a security lack, the security community is actively proposing innovative solutions to protect ADS-B communication. However, testing and evaluating security frameworks is complex due to the limited number of simulators and the impossibility of conducting real-world experiments.

In this paper, we present an *OpenScope-sec* an ADS-B simulator to support the security research and the implementation of novel anomaly detection systems. Our simulator extends the existing ADS-B simulator tools with the possibility of implementing a wider range of attacks. The list of attacks included is based on a preliminary analysis of the current literature, where we collected the most common attacks proposed on ADS-B communication and the existing simulators. Finally, for each attack implemented, we discuss possible anomaly detection approaches to detect the attacks and the consequent changes in legitimate parameters. order to detect possible attacks in real ADS-B messages.

## CCS CONCEPTS

• **Security and privacy** → **Security protocols**; Intrusion detection systems; **Distributed systems security**.

## KEYWORDS

ADS-B, ATC, Wireless Security, Simulator


*ARES 2023, August 29–September 01, 2023, Benevento, Italy*

ACM ISBN 979-8-4007-0772-8/23/08…$15.00
https://doi.org/10.1145/3600160.3605065

**ACM Reference Format:**
Riccardo Cestaro, Mauro Conti, Eleonora Mancini, and Federico Turrin. 2023. OpenScope-sec: An ADS-B Simulator to Support the Security Research. In *The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3600160.3605065

## 1 INTRODUCTION

The industrial revolution and the digitization of processes led to the introduction of new technologies and standards in the aviation system. In fact, the traditional radar-based aviation communication system has been replaced by the Automatic Dependent Surveillance–Broadcast (ADS-B) system, which has nowadays become the *de-facto* standard system to monitor aircraft traffic monitoring. According to the US Federal Aviation Administration, from 2020, all aircraft are required to support Automatic Dependent Surveillance ADS-B transmission system [19].

The ADS-B system brings many advantages for pilots and air traffic controllers. The pilot can receive traffic information from nearby aircraft equipped with ADS-B, including information about weather, terrain, and neighbor aircraft positions. Also, ADS-B allows aircraft to know their relative positions, simplifying air-traffic conflict detection and resolution. Moreover, ADS-B has better resolution than traditional radar systems, which helps optimize and compact air traffic. On the other hand, ADS-B ground stations are cheaper to install and operate and have a longer service life than radar systems [7, 22].

However, despite being a recent standard, ADS-B does not include security features like encryption, authentication, or integrity verification. This security shortfall makes ADS-B prone to cyber attacks like eavesdropping, message injection, message deletion, message modification, and jamming [20, 23], leading to dangerous threats to the users, the surrounding environment. Recent history also highlights cyber attacks attempt on aircraft systems by malicious actors. For instance, in 2015, a WestJet aircraft allegedly transmitted a modified code, imputed to a hijacking modification [16].

To cope with these vulnerabilities in recent years, numerous scientific works proposed security mechanisms to increase the security properties of the ADS-B communication [4, 8, 9, 11]. Unfortunately,

most studies rely on private datasets or data collected with expensive instrumentation. For this reason, it is still difficult to develop novel solutions in this field and cross-validate the results.

**Contribution.** In this work, we present *OpenScope-sec*, an open-source tool to simulate ADS-B attacks and to collect ADS-B messages originating from both regular and under attack aircraft. To deploy the simulator, we first collected all the attacks in the literature implemented in this kind of communication, and then we compared the existing simulation tools, highlighting their limitations. *OpenScope-sec* is built on top of the existing *OpenScope* tools and includes all the common attacks in the literature on ADS-B systems, resulting in the possibility of implementing nine different attacks. We release the source code of the implementation of GitHub*. Finally, to support the beginner in this field, we discuss possible anomaly detection techniques for each attack implemented to identify the malicious behavior of the aircraft under attack. We summarize the contribution of this paper as follows.

- We review the literature and identify the different existing attacks implemented on the ADS-B communication and the existing ADS-B simulation tool and their limitations.
- We extend the existing ADS-B simulator to support a wider range of attacks simulation to support the security research community in developing ADS-B security frameworks. The simulator is available in an open-source fashion.
- We discuss the possible anomaly detection techniques to identify the aircrafts under attack.

**Organization.** This paper is organized as follows. Section 2 briefly recalls the functioning of ADS-B system and its security issues. Section 3 describes the threat model considered in *OpenScope-sec*, the existing attacks on ADS-B, and compares the proposed simulator functionalities with the related work. Section 5 discusses the possible anomaly detection approaches to detect the attacks. Finally, section 6 concludes the paper.

## 2 ADS-B COMMUNICATION

ADS-B lies in the category of Air Traffic Management and Control (ATM/ATC) surveillance system and enables an aircraft to determine its position (e.g., via satellite navigation) and periodically broadcast it to be tracked by the ground station and the other aircraft.

ADS-B system comprises two entities, the broadcast transmitter, referred to as ADS-B OUT, and the broadcast receiver, referred to as ADS-B IN. The first one is used to provide air traffic controllers with real-time position information (that is usually more accurate than the one obtained with radar-based systems); the second one is used to receive Flight Information Services-Broadcast (FIS-B), Traffic Information Service-Broadcast (TIS-B) data and other ADS-B data such as direct communication from nearby aircraft [7].

Generally, ADS-B employs the Global Positioning System (GPS) to determine the aircraft's current position and speed. Then this information is broadcasted through a digital data link channel and other aircraft information. The standard data link operates at 1090MHz, and it is a modified Mode-S transponder with enabled extender squitter [9, 20, 22]. Each ADS-B message includes a preamble for synchronization and a data block that can be either short or extended. The data block consists of the downlink format (message type), a capability field, the *ICAO* 24-bit address, which is a unique identifier of the sender, various surveillance information, and a 24-bit parity check [23]. Figure 1 schematizes the structure of ADS-B messages. The surveillance information field of the messages contains aircraft attributes [9, 23]: timestamp, current coordinates (i.e., longitude and latitude), horizontal and vertical velocity, altitude above the sea, heading (i.e., the direction of progress), and emergency codes.

| Preamble (8 bits) | Downlink format (8 bits) | Capability (3 bits) | ICAO (24 bits) | Payload (56 bits) | Parity Check (24 bits) |
|---|---|---|---|---|---|

**Figure 1: ADS-B message fields.**

**ADS-B Security.** The ADS-B communication is insecure by design since they do not implement any security mechanism. This leads to five major vulnerabilities [7, 15]:

(1) Lack of *authentication*, which is needed to prevent an unauthorized entity from sending and receiving messages;
(2) Lack of *message signature*, to avoid aircraft impersonation and message modification;
(3) Lack of *message encryption*, which could protect sensitive data from eavesdropping;
(4) Lack of *challenge-response mechanisms*, to prevent replay attacks;
(5) Lack of *ephemeral identifiers*, to guarantee data privacy.

Therefore, due to the vulnerabilities of ADS-B system, an attacker who has access to the wireless channel can easily modify, inject or delete messages. Also, the lack of confidentiality enables eavesdropping, and the nature of the wireless channel, i.e. the fact that there is not physical barrier to access the data link, allows the attacker to perform a Denial of Service attack through jamming [15, 18, 20].

## 3 OPENSCOPE-SEC SIMULATOR

In this section, we introduce the simulator we conceived. In Section 3.1 we outline the adversary model considered, while in Section 3.2, we scout and collect the existing works in the literature of ADS-B communication. Then, in Section 3.3 we compare the existing simulators with our simulator regarding available attacks. In Section 4 we provide an overview of the different functions of our ADS-B simulator.

### 3.1 Adversary Model

The adversary model considered in *OpenScope-sec* is similar to the one proposed in the other related work [9, 23]. As explained in previous sections, ADS-B communication lacks security mechanisms, enabling any Man-in-the-middle related attack. Therefore, the attacker requires only a sufficiently strong antenna that transmits and receives in a targeted area. We consider an attacker able to intercept and modify the communication of a target aircraft but also forge new messages. This attacker can be ground-based (e.g., using a low-cost SDR-based spoofer) or air-based (e.g., a drone or a malicious aircraft). We schematize the threat model in Figure 2.

*OpenScope-sec repository: https://github.com/RiccardoCestaro/OpenScope-sec

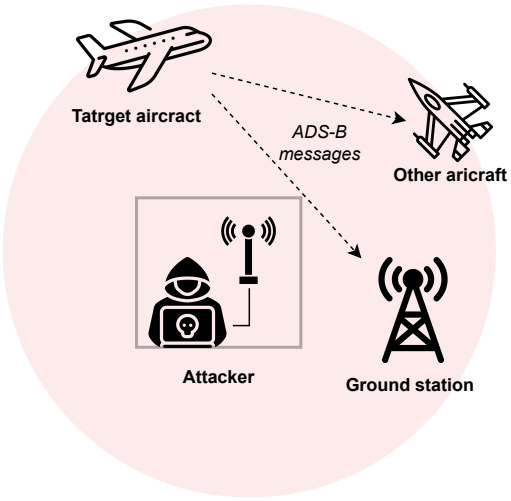


**Figure 2: The threat model considered.**

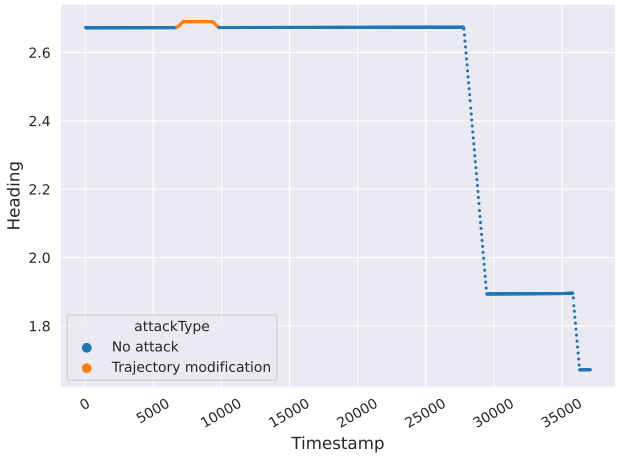


**Figure 3: Virtual trajectory modification attack before landing phase.**

## 3.2 Existing Attacks

We analyzed the existing literature to identify the most common attacks implemented in ADS-B communication. All the attacks proposed to exploit the insecurity by the design of the ADS-B protocol.

In recent years, numerous works surveyed the possible attacks that can afflict ADS-B communication [15, 18, 22]. Other works instead implemented such attacks and deployed anomaly detection systems to identify the malicious data [8, 9, 12, 23]. In particular, in these works, the authors synthetically generate new malicious messages and inject them into an existing dataset. Afterward, the researchers implement an anomaly detector on the resulting dataset. However, all these works do not provide any dataset to the community, creating a lack of reputability or the study or novel approaches. Moreover, it is difficult to test these attacks in real-world scenarios due to the impossibility of transmitting data at the ADS-B corresponding frequency with the risk of affecting real systems. For this reason, we decided to develop a simulator to reproduce several attacks on ADS-B communication, to support the security research. To this end, we implemented the attacks identified in the literature scouting simulator proposed in this paper. In particular, we focused on attacks at the application layer (i.e., the ADS-B dataframe) and not at the physical layer (i.e., raw signal level). In the following, we summarize the attacks identified in the literature and implemented in our proposed simulator. Note that some attacks may have a different name in different works, but the goal and methodology are the same.

**Non-responsive aircraft.** This attack affects the communication between aircraft and Air Traffic Controls (ATCs) by destroying all messages directed from ATCs and destined for the target aircraft (i.e., black hole attack). As a result, the aircraft will no longer respond to commands issued by the ATCs.

**Jumping aircraft.** This attack consists in a message modification attack in which the adversary spoofs the longitudinal and latitudinal coordinates field. In this way, the aircraft will virtually "jump" to another location. After the jump, the attacker will continue to send modified messages with a different position until the aircraft reaches its destination (which is still the original one).

**Aircraft displaying false information.** In this message modification attack, the adversary randomly spoofs velocity or altitude message fields. Unlike the previous attack, the fake messages are not continuous in this case. Indeed the attacker may randomly allow the aircraft to send authentic messages between the fake ones.

**Virtual trajectory modification.** In attack, we modify the aircraft's flight direction in the heading field. Instead of changing it randomly like in the previous attacks, we allow the user to choose the degree of change and the slope. At the end of the attack, the aircraft returns to the original with a movement similar to the previous change. We show an example of the effect of this attack on the heading field of the ADS-B message Figure 3.

**False Alarm.** A false attack alarm consists in spoofing the transponder code field of the ADS-B message of the aircraft to substitute it with a corresponding alarm field (e.g., emergency code 7700). In our implementation, when an aircraft is affected by this attack, the transponder code is replaced with an emergency code, chosen randomly. The original code is restored when the attack is over.

**Aircraft Spoofing.** Here the adversary modifies the aircraft's *ICAO* 24-bit identifier. Therefore, this attack aims to masquerade aircraft to the ground station and the other aircraft to hide or camouflage its presence. However, by design, the *OpenScope* simulator does not use the *ICAO* field in the messages. Instead, it uses an identifier field called *id* and assigns a unique address to each plane. Therefore, to simulate this attack, we modified the *id* field by assigning it the identifier of another existing aircraft.

**Ghost Injection.** The Ghost injection attack is performed by injecting fabricated ADS-B messages on the same frequency as the legitimate ones, making non-existing aircraft appear on the radar. To simulate this, we generated a set of ghost aircraft, i.e., fake planes created with random altitude, callsign, heading, speed, and transponder code, and added them to the simulator. The user can choose the number of random temporary aircraft.

**Message Delay.** The aircraft affected by this attack sends ADS-B messages with a lower frequency than normal. In a real-world scenario, this could be obtained by the adversary by deleting some of the ADS-B messages that the aircraft broadcasts, make possible to lose the precise real-time monitoring of the aircraft.

**Aircraft standing still.** After the launch of this attack, the longitudinal and latitudinal coordinates sent by the aircraft will no longer change, and the velocity will be set to 0, making the aircraft stand still.

### 3.3 Existing Simulators

Despite there being many works in literature related to the security of ADS-B communication, currently, the available open-source simulators provide limited functions. We provide a comparison between the existing simulators and our proposal in Table 1. The most famous ADS-B traffic simulator is probably *OpenScope* [2], developed with JavaScript. *OpenScope* is an open-source Air Traffic Control simulator allowing users to enter commands to issue instructions and communicate with aircraft. These commands include for example departure, arrival, routing, basic control instruction commands. The user enters the commands using a GUI available on the *OpenScope* website [3], which consists of an airport in which the user can monitor various simulated flights that are following a predetermined route. However, this original version of the simulator was not designed for the purpose of security studies and therefore do not include any attack integration to the communication. Blåberg et al. extended *OpenScope* [5] to allow the user to simulate attack on ADS-B communication. However, the attacks implemented are limited to three: *Non-responsive aircraft*, and *Jumping aircraft*, and *Aircraft displaying false information*.

In their thesis Thorn and Wahlgren [21] further extended the *OpenScope* version of Blåberg et al. [5] by including two additional attacks: *Trajectory modification* and *Transponder code alteration*. More recently, Boström and Börjesson in their thesis [6], extended the previous work [21] by implementing two additional attacks: *Impersonation attack* and *Sybil attack*. Unfortunately, at the writing time the source code is not anymore available. Therefore we re-implement those attacks and to include them in our proposed simulator. To comply with other work in literature [9, 15, 21, 23] we also re-named the *Impersonation attack* as *Aircraft spoofing* and *Sybil attack* as *Ghost injection*. Following the attack existing in the literature, differently from these simulators, we also include the *Message Delay*, and *Aircraft standing still* attacks. Another simulator available is proposed by Van Thuan et al. [10]. However, this simulator is conceived for educational purposes only; therefore, it does not include any possibility to implement attacks on communication. BlueSky[1] is an ADS-B simulator entirely developed in Python. Similarly to the original version of *OpenScope*, it does not include any attack implementation by design. However, despite offering similar functionalities of *OpenScope* it did not attract the attention of previous security researchers. For this reason, we decided to focus on *OpenScope*, in line with previous works.

## 4 FUNCTIONS OVERVIEW

Besides implementing attacks on aircraft communication, we kept all the original functions of *OpenScope*, such as managing flight routes, issuing communication between ATC aircraft, and measuring various flight parameters. In the following, we overview the new functionalities and utilies we implemented on *OpenScope-sec* compared with the existing simulators.

**Target aircraft selection.** In the extension of *OpenScope* implemented by Wahlgren and Thorn [21], the attacks are assigned randomly to existing aircraft according to parameters that can be set by the user through the GUI, like how many aircraft should be affected and how the attack types should be distributed between them. While we kept these functionalities, we added another option allowing the user to select a specific aircraft and assign a particular attack type. The aircraft can be chosen according to its identifier.

**Attack duration.** The duration of each attack in Wahlgren and Thorn [21] solution can be selected by the user. In addition to this mechanism, we added an option to start a timer: when it's over it will cease the attacks and reset all the settings to their default value. This can be useful when it is required to make the attack last for a precise amount of time.

**Labeled dataset generation.** To support the generation of the dataset and the subsequent data analysis, we added the possibility to save all ADS-B data in a *csv* file using the download function. This was already implemented in the version of Wahlgren and Thorn [21]. In addition to this, to facilitate the classification tasks we added the possibility to download only a specific flight and we implemented a field in the `.csv` with the label of the occurring attack.

## 5 ATTACK DETECTION

The goal of *OpenScope-sec* is to support the research and training of researchers on the ADS-B communication field from the cybersecurity perspective. Among the different functionalities of *OpenScope-sec*, there is the possibility to download a labeled dataset containing all the attacks performed on a session. For instance, this dataset can be used to develop novel attack detection techniques. In the following, we discuss the possible detection techniques that can be applied to the different attacks available on *OpenScope-sec*. The approach to detecting anomalies can vary based on the detection goal and the application. For this reason, every attack we implemented in the simulator may require a specific detection strategy (e.g., changing point detection, flight whitelisting). Furthermore, most of the attacks may be straightforward to detect with very simple approaches. Indeed, to detect the trajectory modification attack reported in Figure 3, a simple time series forecast trained on the normal route may be sufficient to identify the attack with high precision.

We want to emphasize that the goal of *OpenScope-sec* is not to craft complex attacks but instead to provide the community a platform to practice with ADS-B communication and security threats, generating datasets and testing efficient and innovative detection techniques. For this reason, in the following, we discuss possible existing methods to detect the attacks. The detection techniques for each attack are summarized in Table 2.

Most attacks implemented in *OpenScope-sec* aims to modify a flight parameter that the aircraft under attack communicates to the ATC ground station. These typologies of attacks include *Jumping aircraft*, *Aircraft displaying false information*, *Aircraft standing still*,

| Attack | Bluesky [1] | OpenScope [2] | ∗Blaberg et al. [5] | Van Thuan et al. [10] | OpenScope [21] | ∗OpenScope [6] | **OpenScope-sec** |
|---|---|---|---|---|---|---|---|
| Non-responsive aircraft | | | ✓ | | ✓ | ✓ | ✓ |
| Jumping aircraft | | | ✓ | | ✓ | ✓ | ✓ |
| Aircraft displaying false information | | | ✓ | | ✓ | ✓ | ✓ |
| Trajectory modification | | | | | ✓ | ✓ | ✓ |
| Transponder code alteration | | | | | ✓ | ✓ | ✓ |
| Aircraft spoofing | | | | | | ✓ | ✓ |
| Ghost injection | | | | | | ✓ | ✓ |
| Message delay | | | | | | | ✓ |
| Aircraft standing still | | | | | | | ✓ |

**Table 1: Attack implementation comparison among the different simulators. ∗ means that the source code is not shared.**

| **Attack name** | **Anomalies location** | **Possible detection approach(es)** |
|---|---|---|
| Non-responsive aircraft | ADS-B packet | Measuring aircraft response delay |
| Jumping aircraft | Latitude and longitude coordinates | Time-series forecasting |
| Aircraft displaying false information | Speed and Altitude | Time-series forecasting |
| Trajectory modification | Heading | Time-series forecasting |
| Transponder code alteration | Transponder code | Whitelisting |
| Aircraft spoofing | Aircraft ID | Time-series forecasting |
| Ghost injection | New aircraft pop up | Whitelisting, Time-series forecasting |
| Message delay | ADS-B packet | Time-based packet detection monitoring |
| Aircraft standing still | Speed | Time-series forecasting |

**Table 2: List of attack points and possible detection approaches.**

and *Trajectory modification* modify one or more fields of the ADS-B packet during the transmission. These attacks can be detected with traditional time-series forecasting methods able to detect data shift behaviors from the expected one, like auto-regressive model [14], changing point detection [17], or the more complicate LSTM [13]. Indeed, since aircraft with the same source and destination airports will always pursue the same route, it will be sufficient to train the algorithm on a legitimate route to detect potential anomalies. Similarly, the *Ghost Injection* can be detected with a forecasting and whitelisting technique. This attack, in fact, generates new aircraft spread around the map. If the fake aircraft has a new identifier, a comparison with the allowed flight will be sufficient. Otherwise, if the attacker is smart enough to use a legitimate identifier, verifying if the aircraft respects the expected route and parameters will be sufficient. The whitelisting approach can also be used to identify possible manumission of the transponder code in the *Transponder code alteration* attack. Indeed every aircraft is allowed to send a subset of transponder code. This property can be used to identify illegitimate transmissions.

*Non-responsive aircraft* and *Message delay* attacks instead target the communication timing, making the affected aircraft non-responsive or less responsive. Generally, ADS-B communication, similarly to other Cyber-Physical System (CPS) applications, are based on a well-defined message exchange protocol with deterministic polling time. Therefore, it is possible to identify potential anomalies in the communication flow by using a time-based detection, i.e., monitoring the frequency of the communication (i.e., the number of messages exchanged by time unit).

## 6 CONCLUSION

In this paper, we presented *OpenScope-sec* simulator to support the cybersecurity research on ADS-B communication. As discussed, ADS-B communication lacks every essential security property. For this reason, it is essential to study and develop new security measures to prevent possible communication attacks. However, physically implementing attacks on real-world communication is challenging and sometimes unfeasible, mainly to law constraints. To this end, a software simulator can overcome this limitation and support the community in generating reliable datasets.

In the first part of the work, we surveyed existing works on ADS-B security to identify the most common attacks implemented by researchers. Then we analyzed the existing simulators and their limitations to improve them. We note that the existing *OpenScope* simulators did not offer an exhaustive implementation of existing attacks in literature. Moreover, the code of some of them is not publicly available, not allowing other researchers to work on it. *OpenScope-sec* is open-source and includes a total of nine different attacks, two more than the most advanced existing simulator.

Finally, to support new researchers on the topic, we discussed possible anomaly detection approaches to implement on the dataset collected to identify the attacks. In conclusion, *OpenScope-sec* simulator can support the security research community in studying ADS-B security issues and developing protection frameworks. Furthermore, *OpenScope-sec* can also be used for educational purposes with novel security researchers approaching the topic.

# REFERENCES


[1] 2015. *Bluesky An Experiment Specification & Orchestration Engine.* https://github.com/bluesky/bluesky [Online; accessed 13-January-2022].

[2] 2021. *openScope Air Traffic Control Simulator.* https://github.com/openscope/openscope [Online; accessed 13-January-2022].

[3] 2021. *openScope Air Traffic Control Simulator.* https://www.openscope.io/ [Online; accessed 13-January-2022].

[4] Aniqua Baset, Christopher Becker, Kurt Derr, Shamik Sarkar, and Sneha Kumar Kasera. 2022. AviSense: A Real-time System for Detection, Classification, and Analysis of Aviation Signals. *ACM Transactions on Sensor Networks* 19, 1 (2022), 1–35.

[5] Anton Blåberg, Gustav Lindahl, Andrei Gurtov, and Billy Josefsson. 2020. Simulating ADS-B attacks in air traffic management. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE, 1–10.

[6] Axel Boström and Oliver Börjesson. 2022. Simulating ADS-B vulnerabilities by imitating aircrafts: Using an air traffic management simulator.

[7] Andrei Costin and Aurélien Francillon. 2012. Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. *black hat USA* 1 (2012), 1–12.

[8] Edan Habler and Asaf Shabtai. 2018. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages. *Computers & Security* 78 (2018), 155–173.

[9] Edan Habler and Asaf Shabtai. 2021. Analyzing Sequences of Airspace States to Detect Anomalous Traffic Conditions. *IEEE Trans. Aerospace Electron. Systems* (2021).

[10] Thi Vu Hien Ho, Minh Vuong Pham, et al. 2021. Development Of Air Traffic Control Simulator System Applied In Education And Training. *Transportation Research Procedia* 56 (2021), 47–54.

[11] Kai Jansen, Liang Niu, Nian Xue, Ivan Martinovic, and Christina Pöpper. 2021. Trust the Crowd: Wireless Witnessing to Detect Attacks on ADS-B-Based Air-Traffic Surveillance.. In *NDSS*.

[12] Tengyao Li. 2021. An Adaptive-Data-Driven Attack Detection Framework on ADS-B Data. (2021).

[13] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series.. In *ESANN*, Vol. 2015. 89.

[14] Igor Melnyk, Bryan Matthews, Hamed Valizadegan, Arindam Banerjee, and Nikunj Oza. 2016. Vector autoregressive model-based anomaly detection in aviation systems. *Journal of Aerospace Information Systems* 13, 4 (2016), 161–173.

[15] Kayvan Faghih Mirzaei, Bruno Pessanha De Carvalho, and Patrick Pschorn. 2019. *Security of ADS-B: Attack Scenarios.* Technical Report. Technical Report. EasyChair.

[16] Nancy Moran and Gerrit De Vynck. 2015. *WestJet Hijack Signal Called False Alarm.* Bloomberg. https://www.bloomberg.com/news/articles/2015-01-10/westjet-hijack-signal-called-false-alarm#xj4y7vzkg [Online; accessed 04/04/2023.

[17] VM Morgenstern, BR Upadhyaya, and M Benedetti. 1988. Signal anomaly detection using modified cusum method. In *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE, 2340–2341.

[18] Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. 2013. Experimental analysis of attacks on next generation air traffic communication. In *International Conference on Applied Cryptography and Network Security*. Springer, 253–271.

[19] TC Smith. 2010. Automatic dependent surveillance broadcast (ADS-B) out performance requirements to support air traffic control (ATC) service final rules. *FAA, Federal Aviation Admin., Washington, DC, USA, Tech. Rep., FAA, Rule* 14 (2010).

[20] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2014. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys & Tutorials* 17, 2 (2014), 1066–1087.

[21] Alex Wahlgren and Joakim Thorn. 2021. Detecting ADS-B spoofing attacks: using collected and simulated data.

[22] Zhijun Wu, Tong Shang, and Anxin Guo. 2020. Security issues in automatic dependent surveillance-broadcast (ads-B): a survey. *IEEE Access* 8 (2020), 122147–122167.

[23] Xuhang Ying, Joanna Mazer, Giuseppe Bernieri, Mauro Conti, Linda Bushnell, and Radha Poovendran. 2019. Detecting ADS-B spoofing attacks using deep neural networks. In *2019 IEEE conference on communications and network security (CNS)*. IEEE, 187–195.