Jing *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:160
https://doi.org/10.1186/s13638-020-01756-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

ADS-B spoofing attack detection method based on LSTM

Jing Wang[1], Yunkai Zou[2*] and Jianli Ding[1]

Abstract

The open and shared nature of the Automatic Dependent Surveillance Broadcast (ADS-B) protocol makes its messages extremely vulnerable to various security threats, such as jamming, modification, and injection. This paper proposes a long short-term memory (LSTM)-based ADS-B spoofing attack detection method from the perspective of data. First, the message sequence is preprocessed in the form of a sliding window, and then, an LSTM network is used to perform prediction training on the windows. Finally, the residual set of predicted values and true values is calculated to set a threshold. As a result, we can detect a spoofing attack and further identify which feature was attacked. Experiments show that this method can effectively detect 10 different kinds of simulated manipulated ADS-B messages without further increasing the complexity of airborne applications. Therefore, the method can respond well to the security threats suffered by the ADS-B system.

Keywords: ADS-B, Attack detection, LSTM, Sliding window, Security threat

1 Introduction

With the significant increase in airspace density, traditional surveillance technologies such as primary surveillance radar (PSR), secondary surveillance radar (SSR), and multilateration (MLAT) technology will have increasing difficulty meeting the future need for the development of air traffic management (ATM) systems. Because ADS-B technology has the advantages of high accuracy, large coverage, support for data sharing, and air surveillance, it has become an important part of the next-generation (NextGen) air transport system. However, the extensive application of data-driven related technologies in the area of cloud computing [1, 2], Internet of Things [3, 4], service recommendation [5, 6], blockchain [7], etc. provides attackers with powerful hardware and software support and richer attack methods, which makes aviation community lost the considerable technical advantage that protected its communication. Since the protocol of ADS-B has the characteristics of open sharing, its security faces great challenges. Specifically, the protocol does not provide any relevant data encryption and authentication, and its messages are broadcast in a simple and open format, which is very vulnerable to eavesdropping, jamming, modification, and injection. In addition, authorized aircraft and air traffic controller (ATC) stations do not perform identity authentication before sending ADS-B messages, and the protocol cannot distinguish authorized entities from unauthorized ones. All these factors make the ADS-B system extremely vulnerable to various spoofing attacks. At present, many studies have successfully verified the possibility of attacking the ADS-B system [8, 9]. Therefore, concerns about its safety will continue to increase with the development of air traffic and the further popularization and application of ADS-B.

This paper proposes an ADS-B spoofing attack detection method based on an LSTM network [10]. We have noticed that the idea of prediction is widely used in various fields, such as web service quality prediction, link prediction in recommender system, and web traffic anomaly detection [11–13], which leads to the core idea of the method used in this paper, namely prediction. Specifically, the ADS-B message sequence data are first preprocessed in the form of a sliding window, and then a neural network

*Correspondence: zouyunkaicauc@qq.com
[2]Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Jinbei Road, Tianjin, 300300, China
Full list of author information is available at the end of the article

composed of LSTM units is used for predictive training. Finally, a threshold is set by calculating the predicted data residual set to determine whether there is an anomaly in the ADS-B data. By setting corresponding thresholds for different features, we can further identify the specific features under attack. In this paper, anomaly data (anomaly) refer to data that have been manipulated and need to be detected. The main contributions of this paper are the following:

1 By analyzing the ADS-B attacks, we construct a neural network made up of LSTM units to detect different types of anomalous data we simulate. Compared with the existing machine learning methods [14, 15], our method does not require complicated feature engineering.
2 We set different thresholds for different features, so that we can determine the specific features containing anomalies. In addition, the experiments show that when a single feature is attacked, it can trigger the overall anomaly threshold and does not affect the abnormal scores of other features. In actual applications, the overall threshold can be used to determine whether an anomaly occurs first, and then use the thresholds of different features to determine the specific features that contain anomalies.

The rest of the paper is organized as follows. In Section 2, we introduce the related work. Then, in Section 3, we describe the process and detailed steps of the anomaly detection method. Using this method, we perform detection experiments on different simulated anomalous data, and analyze and discuss the results in Section 4. Finally, we conclude in Section 5.

2 Related work

2.1 Research status

In recent years, researchers have carried out related research on the security issues of the ADS-B system and have given suggested security measures and solutions, mainly including the following aspects: (1) Prevent eavesdropping and modification by encrypting ADS-B messages [16]. Because this method needs to change the existing ADS-B protocol structure, it is difficult to implement. (2) An aircraft is authenticated through a challenge response [16], and additional sensors are added in the airspace to verify the security of the transmitted data [8, 17, 18]. However, the ADS-B system has been deployed on most aircraft, and software and hardware installations and changes require strict airworthiness certifications, so they are difficult to implement at this stage. (3) *Position-based verification methods* [19–22]: these methods usually perform a secondary check on the position claimed by the aircraft or other ADS-B users. The principle is to establish a mechanism that can find the exact position of the message sender, which is essentially different from the verification of the broadcast source and the message. The advantage of this method is that it can be used as a primary navigation system or even a Global Positioning System (GPS) backup system because it can generate additional position data, which can be combined with ADS-B and radar systems. However, such methods usually require synchronization of multiple ground stations or receiving devices, and the complexity is high. (4) *Methods of antenna verification Direction of Arrival* (*DOA*) [23–25]: these methods can avoid problems such as time synchronization and data fusion and do not need to change the existing ADS-B protocol. However, this approach requires spatial search direction finding, has high computational complexity, and is sensitive to array errors. (5) From the perspective of data, a machine learning method is used to reconstruct the ADS-B message sequence, and the reconstruction error is used to detect anomalous messages. Based on the original features contained in an ADS-B message, Habler et al. calculated the distances from all points on the track to four special nodes and the distances between two adjacent track points, for a total of 5 parameters, as additional training features to perform anomaly detection [14]. Our research group statistically expands the original features based on the strong temporal correlation of ADS-B messages so that the model can better capture the time dependence of the data [15]. Although such methods can detect anomaly data, they cannot further determine the specific cause of the anomaly, that is, which data items (features) in the ADS-B message have been modified. In addition, these methods need to further expand the features of the original data to a certain extent, that is, perform more complicated feature engineering. These data processing steps undoubtedly increase the complexity of the application in the actual process.

2.2 Types of ADS-B attacks

The ADS-B system is a new paradigm of air traffic control and does not require manual operation or inquiries. It can automatically obtain parameters from relevant airborne equipment and broadcast the flight status information of the aircraft to other aircraft or ground stations for controllers. According to the direction of aircraft information transmission, the system functions can be divided into two categories: ADS-B IN and ADS-B OUT [26]. The former is an optional service that enables the aircraft to receive and display detailed information broadcast by other aircraft operating in the same area. The latter is the basic function of the on-board ADS-B equipment. It sends the aircraft's position information and other additional information to other aircraft or controllers at a certain period, mainly including aircraft identification information, speed, heading, and climb rate. Ground stations

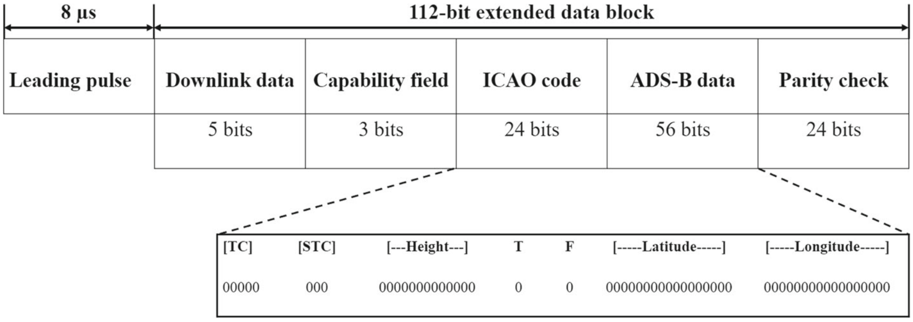


Fig. 1 ADS-B protocol with massage length of 112 bits

Table 1 ADS-B packet attack types

Serial number	Attack type	Purpose of attack	Way of attack
1	Eavesdropping	Eavesdrop operating status information of aircraft (aircraft reconnaissance)	Obtain ADS-B data of corresponding airspace through ADS-B IN device
2	Jamming	Jam the transmission of an ADS-B message in a specific airspace (ground station flood denial, aircraft flood denial)	By using an ADS-B transmitting device with sufficiently high transmit power in the relevant frequency band
3	Message injection	Inject fake aircraft into specific flight scenarios, confusing air traffic control systems (aircraft target ghost injection/flooding)	By using a transmitting device with sufficient high transmit power in the relevant frequency band and capable of generating correct modulation and conforming to the ADS-B message format
4	Message deletion	Delete some or all of the information contained in a message (aircraft disappearance)	By implementation at the physical layer through constructive or destructive interference
5	Message modification	Modify the information contained in a message (virtual trajectory modification)	Realized by overshadowing and bit-flipping at the physical layer of the system and can also be achieved by combining two attack methods: false message injection and message deletion

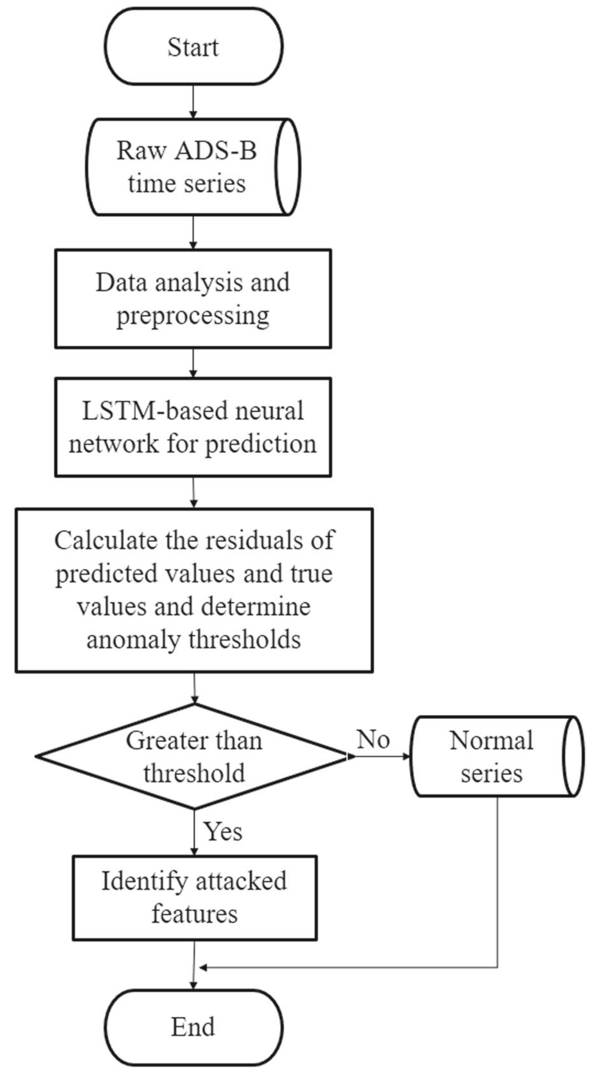

Start
Raw ADS-B time series
Data analysis and preprocessing
LSTM-based neural network for prediction
Calculate the residuals of predicted values and true values and determine anomaly thresholds
Greater than threshold
No
Normal series
Yes
Identify attacked features
End

Fig. 2 Overall process of the method

can monitor air traffic by receiving this information. The ADS-B protocol format is shown in Fig. 1.

The risks faced by the ADS-B system are essentially derived from the broadcast nature of radio frequency communications and the fact that messages are broadcast as unencrypted plain text [27]. The importance and strong attackability of the aircraft operating status information that these messages contain make them the main target for malicious attackers. At present, the types of attacks that exist for the ADS-B system are mainly divided into eavesdropping, jamming, message injection, message deletion, and message modification [28] (Table 1). Among them, eavesdropping will not directly harm the air traffic control system, so the impact is minimal. Message deletion will have an impact on the surveillance system, causing the aircraft to temporarily disappear from the ATC map, but it can be identified by surveillance systems such as radar and multilateration systems. Message modification is a typical spoofing attack. For example, if an attacker continuously changes the aircraft position information in ADS-B messages by small amounts, that is considered a “frog boiling”-type spoofing attack [29]. At this time, other surveillance technologies (such as radar surveillance systems) and positioning technology will have difficulty detecting these small differences due to accuracy issues, resulting in incorrect guidance to air traffic controllers or delaying the response of the collision avoidance system. This has a great impact on the ATC system.

3 ADS-B spoofing attack detection method

3.1 Overall process

Figure 2 shows the overall flowchart of the method proposed in this paper. First, we proceed from the original ADS-B data, process the data into a sliding window composed of ADS-B vectors, and then input the data into a neural network composed of LSTM units for prediction training. After that, additional data (not the training set) is selected and input into the trained model, and the overall anomaly threshold and the threshold corresponding to each feature are determined by calculating the residual of the predicted value and the true value set. When performing anomaly detection, we can first determine whether an anomaly occurs through the overall threshold. If an anomaly occurs, we can further compare whether the anomaly score of each feature exceeds the corresponding threshold. Features with abnormal scores exceeding the threshold may belong to the attacked features.

3.2 Data preprocessing

Before model training, the dataset needs to be preprocessed according to the steps shown in Fig. 3. First, the features related to the aircraft operating status information are extracted from the ADS-B message, including the aircraft’s longitude, latitude, altitude, speed, heading, and climb rate. Then, the data are sorted according to the International Civil Aviation Organization (ICAO) code (the unique identifier of each aircraft) so that the dataset is sorted according to different flights; the form is shown

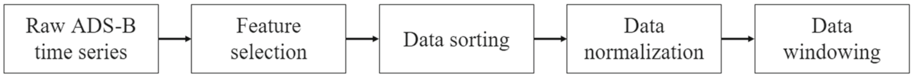


Fig. 3 Data preprocessing steps from raw ADS-B time series

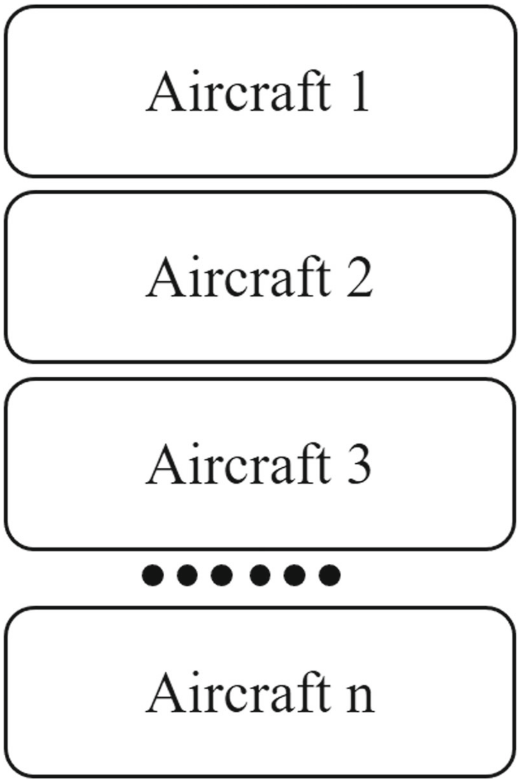


Fig. 4 Dataset arrangement according to different flights

in Fig. 4. Next, the dataset is normalized so that the scaling transformation of different feature dimensions makes the features comparable between different measures without changing the distribution of the original data. Finally, the data are processed into window form according to the time-dependent relationship between ADS-B data features.

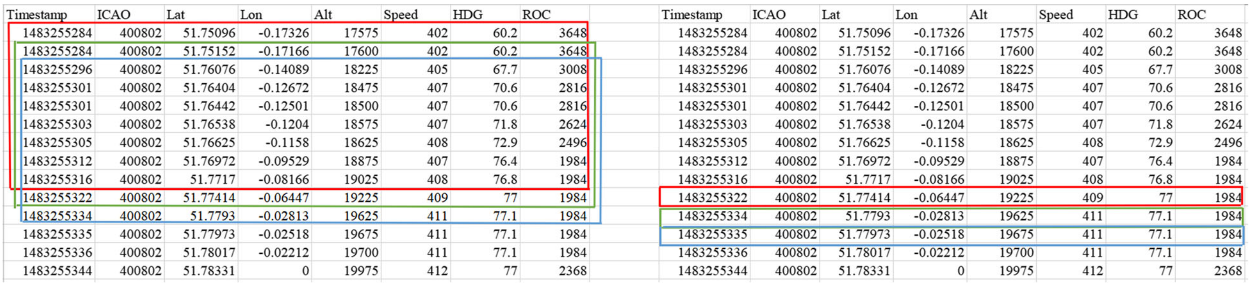
3.3 Sliding window

Define an n-dimensional time series $S = \{S_1, S_2, ..., S_c\}$ to represent the ADS-B sequence window, which is composed of a series of n-dimensional vectors, where C is the length of the time series. $S_i = \{s_1, s_1, ..., s_n\}$ is an n-dimensional vector, and each dimension corresponds to a feature. Specifically, S represents a window composed of continuous C pieces of an ADS-B message, and each vector S_i contains the features extracted from the corresponding ADS-B message, namely the longitude, latitude, altitude, speed, heading, and climb rate. Considering the time correlation of ADS-B data, the data are processed into the form of a sliding window. For example, a window with a length of 10 is selected, and the training phase first uses the data with the serial number [1,10] to predict the 11th data; then, by sliding the window, the data with the serial number [2,11] are used to predict the 12th data, and the rest of the data all follow this pattern. Figure 5 shows a schematic diagram of the sliding window, including the timestamp, ICAO, latitude, longitude, altitude, speed, heading, and climb rate from left to right. From top to bottom, different colored boxes correspond to different sliding windows, with model input on the left and model output on the right. The data are in comma-separated values (CSV) format.

3.4 Model structure and parameter settings

This paper uses an LSTM network to predict an ADS-B sequence. Considering that the input data is not of high dimension and has obvious change rules, the shallow neural network structure can be used to learn the internal connection of the data. The model is built by the keras framework. The specific structure is shown in Fig. 6.

The network is a sequential model consisting of a layer of LSTM units and a fully connected layer. The number of LSTM units is 14, and the number of fully connected layer units is 7, which is the dimension of the ADS-B vector (ICAO is used for flight sequencing and does not participate in model training). In fact, an LSTM unit is a memory unit for learning long-term patterns, including the current state and three nonlinear gates: a forget gate, input gate, and output gate. The forget gate is responsible for determining how much information to remember. It is determined by a nonlinear function



Timestamp	ICAO	Lat	Lon	Alt	Speed	HDG	ROC
1483255284	400802	51.75096	-0.17326	17575	402	60.2	3648
1483255284	400802	51.75152	-0.17166	17600	402	60.2	3648
1483255296	400802	51.76076	-0.14089	18225	405	67.7	3008
1483255301	400802	51.76404	-0.12672	18475	407	70.6	2816
1483255301	400802	51.76442	-0.12501	18500	407	70.6	2816
1483255303	400802	51.76538	-0.1204	18575	407	71.8	2624
1483255305	400802	51.76625	-0.1158	18625	408	72.9	2496
1483255312	400802	51.76972	-0.09529	18875	407	76.4	1984
1483255316	400802	51.7717	-0.08166	19025	408	76.8	1984
1483255322	400802	51.77414	-0.06447	19225	409	77	1984
1483255334	400802	51.7793	-0.02813	19625	411	77.1	1984
1483255335	400802	51.77973	-0.02518	19675	411	77.1	1984
1483255336	400802	51.78017	-0.02212	19700	411	77.1	1984
1483255344	400802	51.78331	0	19975	412	77	2368

Timestamp	ICAO	Lat	Lon	Alt	Speed	HDG	ROC
1483255284	400802	51.75096	-0.17326	17575	402	60.2	3648
1483255284	400802	51.75152	-0.17166	17600	402	60.2	3648
1483255296	400802	51.76076	-0.14089	18225	405	67.7	3008
1483255301	400802	51.76404	-0.12672	18475	407	70.6	2816
1483255301	400802	51.76442	-0.12501	18500	407	70.6	2816
1483255303	400802	51.76538	-0.1204	18575	407	71.8	2624
1483255305	400802	51.76625	-0.1158	18625	408	72.9	2496
1483255312	400802	51.76972	-0.09529	18875	407	76.4	1984
1483255316	400802	51.7717	-0.08166	19025	408	76.8	1984
1483255322	400802	51.77414	-0.06447	19225	409	77	1984
1483255334	400802	51.7793	-0.02813	19625	411	77.1	1984
1483255335	400802	51.77973	-0.02518	19675	411	77.1	1984
1483255336	400802	51.78017	-0.02212	19700	411	77.1	1984
1483255344	400802	51.78331	0	19975	412	77	2368

Fig. 5 Illustration of sliding window. First uses the data with the serial number [1,10] to predict the 11th data; then, by sliding the window, the data with the serial number [2,11] are used to predict the 12th data, and the rest of the data all follow this pattern

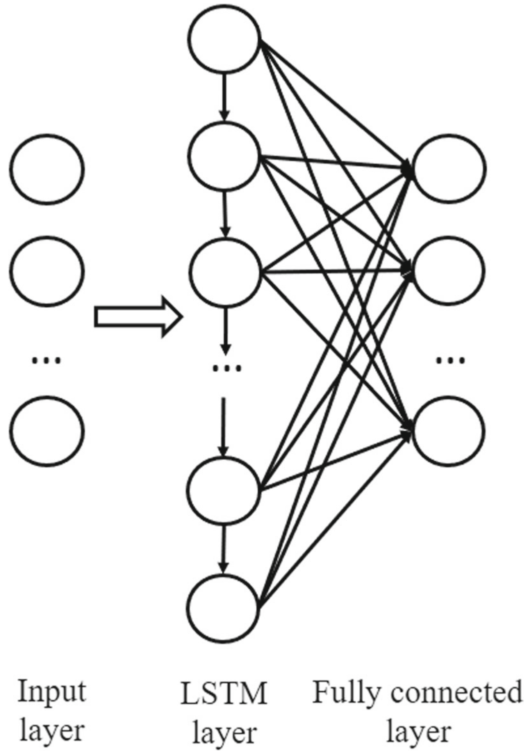


Fig. 6 Model structure diagram. At the LSTM layer, each circle represents an LSTM unit, and the down arrows in this layer represent that the cell status and output of each unit will be passed to the next unit

and outputs a number between 0 and 1, where 0 means forgetting all the information in memory and 1 means keeping all the information in memory. The input gate is responsible for deciding how to update the old unit status; that is, the new information is selectively recorded into the unit status. The output gate is responsible for deciding how much memory information is passed to the next unit.

During the training process, the ADS-B data is input into the neural network one by one in the form of a sliding window, and the training output is the next data in the input window. In addition, the loss function for training uses the mean square error.

3.5 Threshold setting

The total dataset is defined as M, and M is divided into three subsets, M_1, M_2, and M_3, where the ratio is approximately 8:1:1. Among them, M_1 is used for model training, M_2 is used for determination of thresholds, and M_3 is modified according to the descriptions of different attack types; then, the model is tested. After the model is trained, M_2 is input into it to obtain a set of predicted values P. The

Table 2 ADS-B attack data simulation method

Attack type	Simulation data	Simulation method
Jamming	Random noise	Multiply the flight value obtained in the original ADS-B message by a random value between 0 and 2.
Injection	Route replacement	Given certain route information, inject different correct route information to replace the sequence for the selected ADS-B sequence segment.
Modification	Fixed offset (+)	Increase the flight value (except time characteristics) obtained in the ADS-B message by 10%.
	Fixed offset (−)	Decrease the flight value (except time characteristics) obtained in the ADS-B message by 10%.
	Height offset (+)	Use 400 ft as multiples to gradually change the altitude characteristics of ADS-B messages. In the selected ADS-B sequence, increase the altitude feature of the first vector by 400 feet, the second by 800 feet, and so on.
	Height offset (−)	Decrease the altitude feature of the first vector by 400 ft, the second by 800 ft, and so on.
	Speed offset (+)	Use 20 knots as multiples to gradually change the speed characteristics of ADS-B messages. In the selected ADS-B sequence, increase the speed feature of the first vector by 20 knots, the second by 40 knots, and so on.
	Speed offset (−)	Decrease the speed feature of the first vector by 20 knots, the second by 40 knots, and so on.
	Heading change	Change the value of the heading information contained in the ADS-B message to the opposite of the original value.
	Climb rate change	Change the value of the climb rate information contained in the ADS-B message to the opposite of the original value.

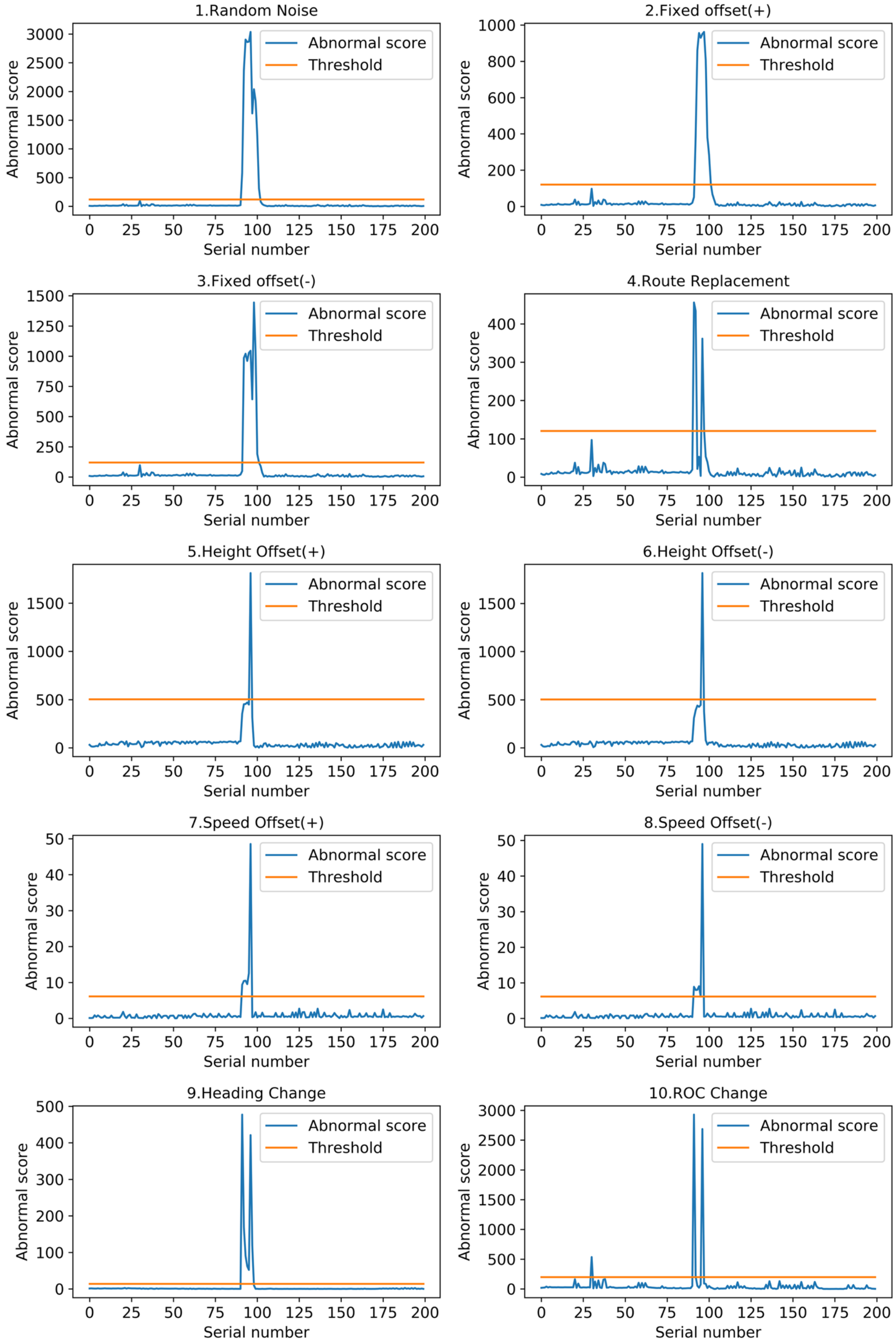


Fig. 7 Abnormal score figures. Abnormal score figures of different kinds of attack data where the abscissa is the data serial number and the ordinate is the abnormal score

set of true values corresponding to P is V, and the residual set of P and V is defined as D. For $d_i \in D$, we have:

$$d_i = |p_i - v_i|$$

where $p_i \in P$, $v_i \in V$, and i is the index coefficient. Then, the mean and standard deviation of set D are calculated and recorded as μ and σ; that is, $E(D) = \mu$ and $D(D) = \sigma^2$. Then, we can define the threshold as follows:

$$t = 3\sigma$$

In the test phase, the corresponding residual set D' is obtained by using the model and the dataset M_3 in the

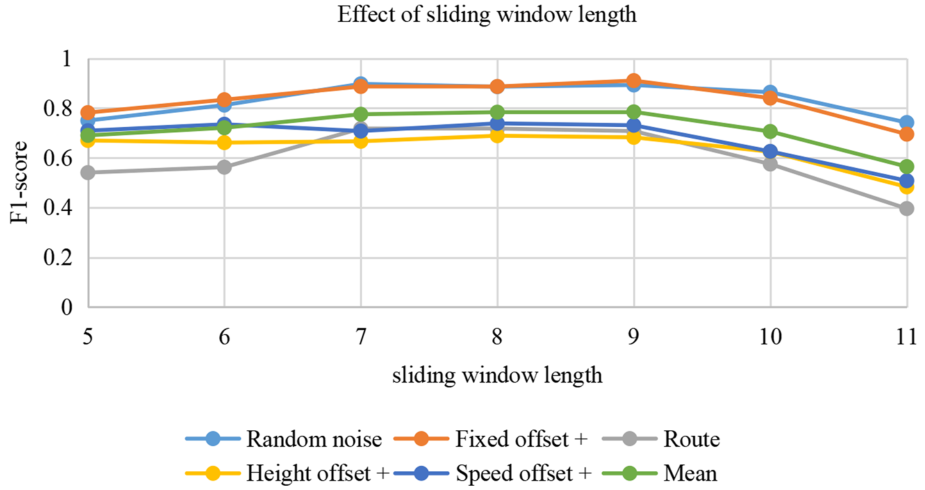


Fig. 8 Effect of sliding window length. Effect of sliding window length on the detection effect for five representative types of anomaly data

same manner as described above. The abnormal score can be defined as:

$$a = \left| d_i' - \mu \right|$$

where $d_i' \in D'$ and μ is the mean of set D.

It is worth noting that the anomaly thresholds of different features are different, which are three times their corresponding standard deviations (it can also be changed according to different needs, if you need to reduce the false alarm rate, you can also set to σ or 2σ). The overall anomaly threshold is the average number of anomaly thresholds for all features. Similarly, the definition of abnormal score also corresponds to the same situation.

In practical applications, the average threshold of all features can be used first to determine whether an attack has happened. Furthermore, if the predicted residual of a certain feature exceeds the corresponding threshold, it can be determined that an anomaly has occurred in this specific feature.

4 Results and discussion

4.1 Attack data simulation

The data used in the experiments in this paper were obtained from a GitHub project [30]. The data were decoded from real ADS-B messages with a total length of approximately 220,000. This paper focuses on 10 different types of attack data for jamming, modification, and injection, as shown in Table 2. The starting point of the anomaly data simulation method is as follows:

1 It can be achieved at the technical level.
2 Try to simulate more realistic data that is not easy to be discovered by the air traffic controller.

The paper [10] gives us a good example of simulation, and our experiment simulates a richer type of anomaly based on it.

Table 3 Average precision, recall, and F1-scores

Attack type	Precision	Recall	F1-score
Random noise	0.9136	0.8902	0.8932
Fixed offset+	0.9667	0.8655	0.9109
Fixed offset−	0.9674	0.9242	0.9415
Route	0.9772	0.5751	0.6844
Height offset+	0.8656	0.5947	0.6824
Height offset−	0.8518	0.5284	0.6316
Speed offset+	0.9768	0.5841	0.7311
Speed offset−	0.9809	0.5530	0.7058
Heading	0.9788	0.4583	0.5914
Climb rate	0.8698	0.1856	0.3032

Table 4 Average precision, recall, and F1-scores (after changing the detection target)

Attack type	Precision	Recall	F1-score
Random noise	0.9136	1.0000	0.9548
Fixed offset+	0.9667	1.0000	0.9830
Fixed offset−	0.9674	1.0000	0.9834
Route	0.9772	1.0000	0.9885
Height offset+	0.8656	1.0000	0.9280
Height offset−	0.8518	1.0000	0.9200
Speed offset+	0.9768	1.0000	0.9883
Speed offset−	0.9809	1.0000	0.7058
Heading	0.9788	1.0000	0.5914
Climb rate	0.8698	0.8958	0.3032

4.2 Results visualization

An independent flight or sequence segment is selected, and the sequence segments with serial numbers [100, 105] are injected with the different types of attacks described in Table 2. Figure 7 shows the abnormal score for a certain flight after modification, where the abscissa is the data serial number and the ordinate is the abnormal score. Different subgraphs represent attacks against different features in the following order: random noise, fixed offset (+), fixed offset (−), route replacement, altitude offset (+), height offset (−), speed offset (+), speed offset (−), heading change, and climb rate change. It can be seen that for different data features, the method can effectively detect the attack using the corresponding threshold.

4.3 Evaluation metrics

To evaluate the method more accurately, this paper uses precision, recall, and the F1-score as metrics. They are defined as follows:

Precision: precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$P = TP/(TP + FP)$$

Recall: recall is the ratio of correctly predicted positive observations to all observations in the actual class.

$$R = TP/(TP + FN)$$

F1-score: the F1-score is the weighted average of precision and recall.

$$F1 = 2 \times (R \times P)/(R + P)$$

TP, *FP*, and *FN* refer to true positive, false positive, and false negative, respectively. We might fail to detect potential anomalies if we only pay attention to precision. However, some false positives might be received when we focus only on recall. The F1-score provides a balance of precision and recall and is therefore used as the main evaluation metric in our experiments.

4.4 Effect of sliding window length

Before statistical analysis of all attack detection results, we first study the effect of different sliding window lengths on the detection effect. We selected five representative types of anomaly data (random noise, fixed offset, route replacement, altitude offset, and speed offset) to test the effect of sliding window length on the detection results. Figure 8 shows the F1-scores of these types of anomalies under different window parameters. For the dataset used in this paper, the detection result is best when the sliding window length is 9. By continuing to increase the window length, the detection effect gradually becomes worse because a longer window will mask the time change in a short time.

4.5 Comprehensive test results

In the test set composition, 20 flight segments are selected, and attacks are injected into two sequence segments, [100,105] and [140,145], for different flight phases. A training model with a window length of 9 is selected to test various attack types, as shown in Table 3 for the comprehensive test results.

As shown in Table 3, this method has a low recall rate in regard to some more difficult attacks (warm-water

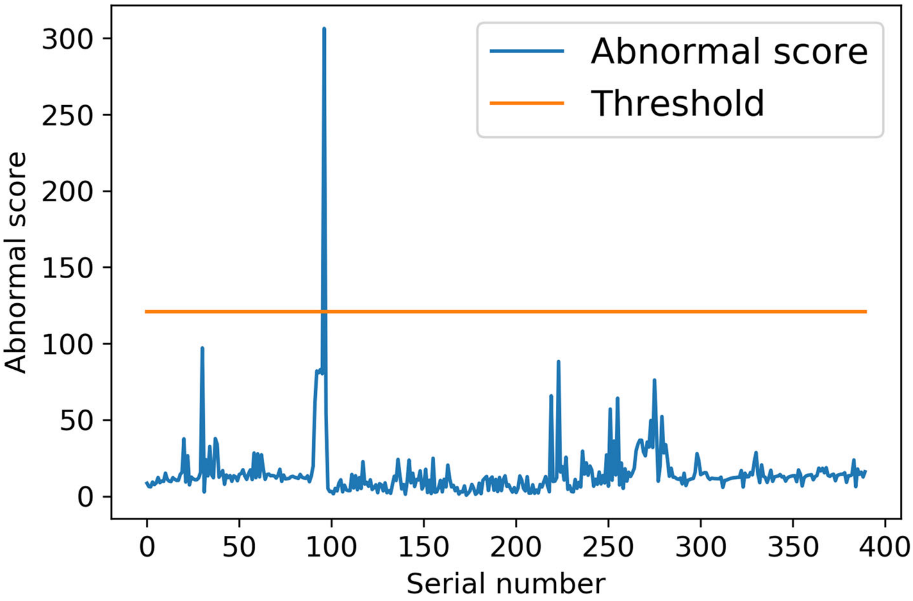


Fig. 9 Triggering effect of latitude change on the average threshold of all features

Table 5 Trigger rates of various anomalies

Attack type	Trigger rates	F1-score
Random noise	1.0000	0.9548
Fixed offset+	1.0000	0.9830
Fixed offset−	1.0000	0.9834
Route	1.0000	0.9885
Height offset+	0.9621	0.8928
Height offset−	1.0000	0.9200
Speed offset+	0.8958	0.8853
Speed offset−	0.9167	0.6470
Heading	0.8542	0.5052
Climb rate	0.9792	0.2969

boiled frog attacks). This is because the results are calculated from separate points when calculating these metrics. For example, in Fig. 6, for attacks such as "height offset," although the attacked data were successfully detected, only one point located in the attacked sequence segment exceeded the threshold. In this case, the recall rate is only 1/5 = 0.2. However, in the actual situation, the data enter the model in the form of sliding windows. Therefore, when an anomaly point is detected, we reasonably suspect that all sliding windows containing that point have the possibility of containing the attacked data. If further analysis is performed, the attacked sequence segment can be accurately detected. The specific method is to change the statistical unit of the recall rate to the number of attacks; that is, the detection target becomes two sequence segments. In this way, the recall rate index is significantly improved. Table 4 shows the results after changing the statistical method.

4.6 Consideration of influencing factors

Figure 9 presents the triggering effect of modified latitude on the average threshold of all features. Similar tests for each feature modification show that a single feature modification will exactly trigger the overall threshold; that is, in practical applications, we can set the overall threshold first, and if anomaly data are detected using this threshold, then the specific feature threshold is used to identify the exact manipulated feature in a further step.

Statistics on the trigger rates of various anomalies are in Table 5, where the F1-score is the result of multiplying the trigger rates.

In addition, this paper also considers whether an attack on one feature will affect other features when attacked, that is, whether it will increase the false alarm rate. Figure 10 shows the latitude abnormal score graph when the altitude is modified. Figure 11 presents a partially enlarged view of the latitude abnormal score.

Figures 10 and 11 show that the latitude anomaly score exhibits only a small fluctuation in the attacked sequence segment ([100,105]), and the magnitude is much smaller than the anomaly threshold. Tests show that when a feature is attacked, it can successfully trigger the overall threshold, and at the same time, it will not affect the abnormal score of other features, which will reduce the complexity of the method for anomaly detection.

4.7 Discussion

In this work, we use public datasets in evaluation. It is possible that they contain a small degree of noise. Furthermore, their data volume is also limited. We will experiment with larger scale datasets in our future work. For the simulated anomaly data, the modified granularity needs

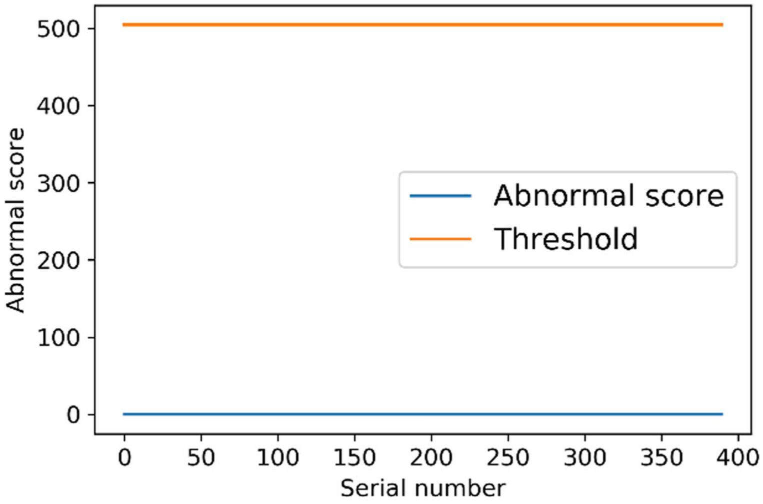


Fig. 10 Latitude abnormal score

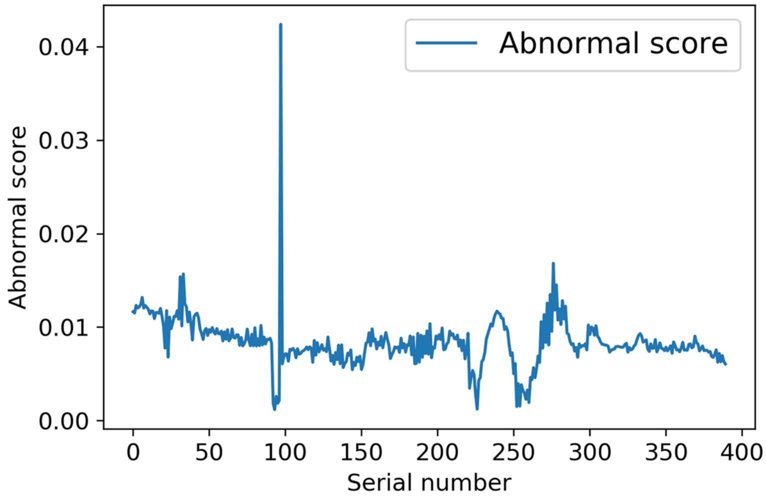


Fig. 11 Partially enlarged view of latitude anomaly

to be further refined. Besides, this method can only find anomalies from a data perspective and cannot further lock the attacker.

Since the ADS-B data changes differently in different flight phases, in the follow-up research, we will consider to divide the dataset according to different flight phases, and train the corresponding models for the data in different phases (parallel process). Then, set the corresponding parameters to improve the correlation between the model and the data to further deal with more complex attack types. In addition, we will also pay attention to the idea of crowdsourcing [31, 32] and further study it on ADS-B system.

On the other hand, although the proposed method cannot completely solve the security problem of the ADS-B system, it will certainly increase the difficulty for attackers to attack the system. Moreover, this method can be easily extended to other aeronautical data, such as GPS signals and radar data.

5 Conclusion

Addressing typical security threats that ADS-B systems may currently suffer, this paper proposes a method for detecting ADS-B spoofing attacks based on LSTM. We use a neural network composed of LSTM units to predict an ADS-B message in the form of a sliding window and set a threshold value by calculating the residual of predicted values and true values to further detect attack data. The detection of 10 kinds of simulated attack data in ADS-B messages shows that this method can effectively detect attack data and further identify the specific features under attack. Since this method does not require complicated feature engineering, the participation of additional nodes, and modification of the existing protocol, it has strong operability in future practical applications.

Abbreviations

ADS-B: Automatic Dependent Surveillance Broadcast; LSTM: Long short-term memory; PSR: Primary surveillance radar; SSR: Secondary surveillance radar; MLAT: Multilateration; TM: Air traffic management; NextGen: Next-generation; ATC: Air traffic controller; GPS: Global Positioning System; ICAO: International Civil Aviation Organization; DOA: Direction of Arrival; CSV: Comma-separated values

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (3122019124).

Authors' contributions

JW conceived of the study, participated in the summary of the types of attacks, and helped to draft the manuscript. YZ participated in its design and coordination, carried out the training and testing of the method, and helped to draft the manuscript. JD participated in the study of related work, and helped to draft the manuscript. All authors read and approved the final manuscript.

Funding

Fundamental Research Funds for the Central Universities (3122019124).

Availability of data and materials

The datasets analyzed during the current study are available in https://github.com/junzis/flight-data-processor/tree/master/data.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Computer Science and Technology, Civil Aviation University of China, Jinbei Road, Tianjin, 300300, China. [2]Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Jinbei Road, Tianjin, 300300, China.

Received: 2 April 2020 Accepted: 28 June 2020
Published online: 12 August 2020

References

1. L. Qi, W. Dou, C. Hu, Y. Zhou, J. Yu, A context-aware service evaluation approach over big data for cloud applications. IEEE Trans. Cloud Comput. **8**(2), 338–348 (2020)
2. W. Zhong, X. Yin, X. Zhang, S. Li, W. Dou, R. Wang, L. Qi, Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment. Comput Commun. **157**, 116–123
3. X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M. Z. A. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled iot. IEEE Internet Things J. **7**(4), 2622–2629 (2020)

4. X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, L. Qi, A computation offloading method over big data for iot-enabled cloud-edge computing. Future Gener. Comput. Syst. **95**, 522–533 (2019)
5. Y. Zhang, C. Yin, Q. Wu, Q. He, H. Zhu, Location-aware deep collaborative filtering for service recommendation. IEEE Trans. Syst. Man Cybern. Syst. (TSMC), 1–12 (2019). https://doi.org/10.1109/TSMC.2019.2931723
6. Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, Q. He, Efficient query of quality correlation for service composition. IEEE Trans. Serv. Comput., 1–14 (2018). https://doi.org/10.1109/TSC.2018.2830773
7. X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, W. Dou, Become: blockchain-enabled computation offloading for iot in mobile edge computing. IEEE Trans. Ind. Inform. **16**(6), 4187–4195 (2020)
8. A. Costin, A. Francillon, in *BLACKHAT 2012, July 21-26, 2012, Las Vegas, NV, USA*. Ghost in the air (traffic): on insecurity of ads-b protocol and practical attacks on ads-b devices, (Las Vegas, ETATS-UNIS, 2012), pp. 1–12. http://www.eurecom.fr/publication/3788
9. M. Schäfer, V. Lenders, I. Martinovic, in *International Conference on Applied Cryptography and Network Security*. Experimental analysis of attacks on next generation air traffic communication (Springer, Berlin, 2013), pp. 253–271
10. A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw. **18**(5-6), 602–610 (2005)
11. Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, Y. Yang, Covering-based web service quality prediction via neighborhood-aware matrix factorization. IEEE Trans. Serv. Comput., 1–12 (2019). https://doi.org/10.1109/TSC.2019.2891517
12. H. Liu, H. Kou, C. Yan, L. Qi, Link prediction in paper citation network to construct paper correlation graph. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–12 (2019)
13. T.-Y. Kim, S.-B. Cho, Web traffic anomaly detection using c-lstm neural networks. Expert Syst. Appl. **106**, 66–76 (2018)
14. E. Habler, A. Shabtai, Using lstm encoder-decoder algorithm for detecting anomalous ads-b messages. Comput. Secur. **78**, 155–173 (2018)
15. J. Ding, Y. Zou, J. Wang, H. Wang, Ads-b anomaly data detection model based on deep learning. Acta Aeronaut. et Astronaut. Sin. **40**(11), 167–177 (2019)
16. C. Finke, J. Butts, R. Mills, M. Grimaila, Enhancing the security of aircraft surveillance in the next generation air traffic control system. Int. J. Crit. Infrastruct. Protect. **6**(1), 3–11 (2013)
17. J. Baek, E. Hableel, Y.-J. Byon, D. S. Wong, K. Jang, H. Yeo, How to protect ads-b: confidentiality framework and efficient realization based on staged identity-based encryption. IEEE Trans. Intell. Transp. Syst. **18**(3), 690–700 (2016)
18. T. Kacem, D. Wijesekera, P. Costa, J. Carvalho, M. Monteiro, A. Barreto, in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. Key distribution mechanism in secure ads-b networks (IEEE, Herndon, 2015), pp. 1–13
19. J. Johnson, H. Neufeldt, J. Beyer, in *2012 Integrated Communications, Navigation and Surveillance Conference*. Wide area multilateration and ads-b proves resilient in Afghanistan (IEEE, Herndon, 2012), pp. 1–8
20. W. Wang, W. Li, D. Lu, Ads-b spoofing detection method using tdoa correlation coefficient. J. Sig. Process. **35**(11), 1784–1790 (2019)
21. R. Kaune, C. Steffes, S. Rau, W. Konle, J. Pagel, in *2012 15th International Conference on Information Fusion*. Wide area multilateration using ads-b transponder signals (IEEE, Singapore, 2012), pp. 727–734
22. M. Strohmeier, I. Martinovic, V. Lenders, A k-nn-based localization approach for crowdsourced air traffic communication networks. IEEE Trans. Aerosp. Electron. Syst. **54**(3), 1519–1529 (2018)
23. J. Naganawa, H. Tajima, H. Miyazaki, T. Koga, C. Chomel, in *2017 IEEE Conference on Antenna Measurements & Applications (CAMA)*. Ads-b anti-spoofing performance of monopulse technique with sector antennas (IEEE, Tsukuba, 2017), pp. 87–90
24. L. Chen, R. Wu, Ads-b spoofing detection method using Doppler effect. J. Sig. Process. **34**(6), 722–728 (2018)
25. W. Wang, G. Chen, R. Wu, D. Lu, L. Wang, in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. A low-complexity spoofing detection and suppression approach for ads-b (IEEE, Herndon, 2015), pp. 1–8
26. R. S. Committee, et al., Minimum aviation system performance standards for automatic dependent surveillance broadcast (ads-b). Technical report, RTCA, Washington DC (1998)
27. M. Strohmeier, V. Lenders, I. Martinovic, On the security of the automatic dependent surveillance-broadcast protocol. IEEE Comput. Surv. Tutorials. **17**(2), 1066–1087 (2014)
28. M. R. Manesh, N. Kaabouch, Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ads-b) system. Int. J. Crit. Infrastruct. Protect. **19**, 16–31 (2017)
29. E. Chan-Tin, V. Heorhiadi, N. Hopper, Y. Kim, The frog-boiling attack: Limitations of secure network coordinate systems. ACM Trans. Inf. Syst. Secur. (TISSEC). **14**(3), 1–23 (2011)
30. J. Sun, H. Vû, J. Ellerbroek, J. M. Hoekstra, Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model. PloS ONE. **13**(10), e0205029 (2018). https://doi.org/10.1371/journal.pone.0205029
31. L. Qi, W. Dou, W. Wang, G. Li, H. Yu, S. Wan, Dynamic mobile crowdsourcing selection for electricity load forecasting. IEEE Access. **6**, 46926–46937 (2018)
32. M. Strohmeier, V. Lenders, I. Martinovic, A k-NN-Based localization approach for Crowdsourced Air Traffic Communication Networks. IEEE Trans. Aerosp. Electron. Syst. **54**(3), 1519–1529 (2016). https://doi.org/10.1109/TAES.2018.2797760.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.