In Pursuit of Aviation Cybersecurity

Experiences and Lessons From a Competitive Approach

Martin Strohmeier | Cyber-Defence Campus, Armasuisse Science and Technology
Mauro Leonardi | University of Rome Tor Vergata
Sergei Markochev | Independent Researcher
Fabio Ricciato | The OpenSky Network
Matthias Schäfer | University of Kaiserslautern
Vincent Lenders | Cyber-Defence Campus, Armasuisse Science and Technology

The passive and independent localization of aircraft has been the subject of much cyberphysical security research. We designed a multistage open competition focusing on the offline batch localization problem using opportunistic data sources. We discuss setup, results, and lessons learned.

The passive and independent localization of aircraft based on their wireless communication with other endpoints has been a long-standing problem and subject of much research. While modern aircraft often broadcast their (unauthenticated) position as obtained via global navigation satellite systems (GNSS), there are a multitude of reasons to calculate or verify aircrafts' positions independently on the ground. Among other applications, the localization of aircraft based on their communication signals can be used to track noncooperative aircraft, improve the security of unencrypted air traffic control (ATC) protocols, or act as redundant backup and safety system in case of outages of critical aviation infrastructures.[1] In recent events, aircraft tracking has become an important factor in independently observing military movements in conflict zones, including the war in Ukraine.[2] This illustrates the true nature of aviation as a large-scale cyberphysical system, with many facets touching on security and privacy.

Originating from military settings, the capability to localize aircraft opportunistically first expanded into the civil aviation domain, and is now available to any actor who controls multiple inexpensive software-defined radios (SDRs). The widespread proliferation of such SDRs has at the same time given rise to globally oriented, crowdsourced flight information websites, such as Flightradar24 (https://flightradar24.com) and the OpenSky Network (https://opensky-network.org). These organizations use the information gathered from many distributed SDR-based receivers of ATC data to display and share the tracks of aircraft around the

Digital Object Identifier 10.1109/MSEC.2023.3265523
Date of current version: 26 April 2023

world. The data of these networks are used for many critical applications, from air traffic management to climate research and open source intelligence in times of conflict.[2]

Consequently, the practical aircraft localization problem (ALP) has expanded: from solving localized, controlled, homogeneous receiver environments to extremely heterogeneous, uncontrolled, and global-scale crowdsourced cyberphysical systems. However, the theoretical algorithms and solutions for use with the ALP have not been adapted to these developments.

Many classical solutions for the ALP have been proposed in the literature, with different theoretical underpinnings and typically focusing on the core multilateration (MLAT) algorithm. However, all suffer from two critical flaws for modern deployments. First, they were principally not designed for the crowdsourced case with its typical organic, noncontrolled receiver growth and placement but instead rely on the ability to place receivers at will and in a proactive, near-optimal fashion. Second, until recently, there has been no scientific, standardized way to compare different methods of solving the different shapes and forms of the ALP.

These issues have recently been addressed with the Localization Reference Data Set (LocaRDS), which was developed specifically by scientists of the OpenSky Network for competitive ALP comparisons, including the present competition.[3] Based on LocaRDS, we conducted an open competition in order to measure and improve the state-of-the-art in ALP research and put it on a solid scientific grounding for securing the localization of aircraft. We seek to reduce the existing fragmentation of research on the ALP, where authors have to build their own test sets from real or simulated data in order to compare their novel methods. Naturally, as previously used data and metrics were generally not available and documentation is sparse, the reproducibility and comparability of results has been very limited until now. By using LocaRDS, we exploit the availability of open real-world crowdsourced flight data, which fulfills the requirements of different localization methods. We hope that through the use of a comparable and standardized source, it will become clear which are the best solutions to the ALP in different scenarios.

Competitions have been a popular method in several areas of computer science, particularly in machine learning. They have proven to engage the community and include stakeholders from outside academia. Competitions seek to foster a specific, often underdeveloped cause and have been applied successfully in (indoor) localization before, both online and in person.[4] We argue that they are also useful in cyberphysical systems research, including security.

Contributions

- We use LocaRDS,[3] a reference dataset for scientific comparability in localization research based on crowdsourced real-world air traffic data, in order to derive an effective benchmark for the ALP.
- We report on the design and execution of a year-long public competition built to find novel and improved solutions to the ALP, in particular for the important crowdsourced setting.
- We analyze the impressive results of the participants and their technical design choices and distill lessons learned from our long-term efforts.

Application of Localization to Security and Privacy in Aviation

Due to their legacy nature, all technologies used in a commercial aircraft today are unauthenticated by nature, opening them up, among other things, to spoofing attacks. Since the rise of cheap commercial off-the-shelf SDRs in the 2000s, published analyses of attacks on wireless communication protocols in critical infrastructure abound. Of interest to us in particular are spoofing attacks on ATC technologies, such as secondary radar systems and the widely supported automatic dependent surveillance-broadcast (ADS-B) technology. ADS-B was mandated in most developed airspaces from 2020 onward and forms the heart of the next generation of ATC, a truly cyberphysical critical infrastructure system. The earliest security analyses were published by Costin and Francillon,[5] McCallie et al.,[6] and Haines.[7] In these analyses, it was already suggested that the use of independent localization could be a suitable way to improve the practical security of the ADS-B system and ATC as a whole, but concrete public research improvements in this area have remained limited.

At the same time, the tracking of aircraft has enjoyed rising importance in recent years, in particular outside of the academic literature. Public websites display aircraft based on crowdsourced signals, both in real time and historically. These open source data are used by journalists, hobbyists, and nongovernmental organizations to investigate the climate emergency, crime, government corruption, human trafficking, and military movements in conflict areas. As not all aircraft send their own location, in particular military aircraft and aircraft operating outside industrialized countries, independent localization of such aircraft helps to improve the necessary data in such cases.

Localization in Crowdsourced Networks

We consider the long-distance outdoor positioning problem to find the 3D position of an aircraft based on the signal characteristics of the communication. Figure 1 shows the abstract process.

This type of localization, or positioning, can in principle be conducted with any communications signal sent out by an aircraft. Without loss of generality, the LocaRDS competition dataset uses the ADS-B system, which is readily collected by many web trackers including, as mentioned before, OpenSky or Flightradar24. Thus, it offers not only sufficient data but based on its popularity, also many target users that would benefit from improved ALP solutions, in particular in a crowdsourced setting. As has been discussed widely in the literature, ADS-B and other ATC protocols are not secure and their verification using independent localization is the current method of choice to improve the security of the system as a whole.[1]

Time Difference of Arrival

The most popular approach to aircraft localization is to use the time differences of arrival (TDoA) concept, where $n > 1$ ground sensors receive and match the same signal sent by an aircraft. At reception, every receiver timestamps the signal. The time of arrival (ToA), measurements are then joined and the differences of all arrival timestamps $t_1, \ldots t_n$ between all n involved receivers are calculated. This is done, for example, by subtracting the earliest timestamp t_{min} or using a fixed receiver of the set as anchor. These data then form the basis for the TDoA approach, which as a surveillance technique is best known as *MLAT*.

MLAT is a proven and well-understood concept used in civil and military surveillance. It serves as an operational method for ATC around airports and even smaller countries (e.g., Austria or Czech Republic). Academic works and aviation regulatory bodies have argued for MLAT being an ideal backup for primary radar systems, which are slowly being phased out due to cost, accuracy, and reliability issues.[8]

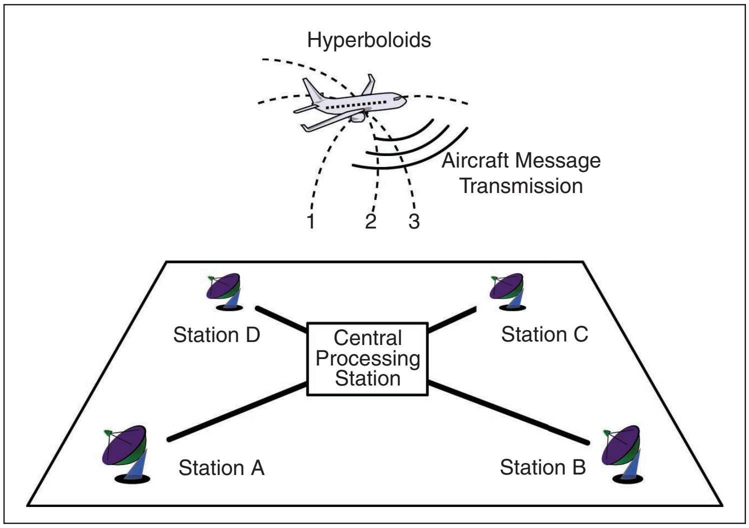

Figure 1. Representation of the ALP.

However, classic MLAT solutions suffer from drawbacks, most notably expensive hardware to enable highly accurate timestamps and tight synchronization. Both are a strict necessity for MLAT algorithms, as they are highly sensitive to noise, in particular in uncontrolled receiver placements where the geometric characteristics are not optimized.[9]

We can assume that the localization accuracy depends on three elements:

1. the measurement accuracy
2. the spatial distribution of the stations
3. the algorithm used to solve the underlying geometric problem.

For the first two elements (measurement noise and sensor distribution), nothing can be improved within a crowdsourced network: it organically grows without any particular optimization of sensor positions and exploits receiver hardware of a given, highly variant, performance. Thus, we can only preprocess the data to statistically characterize the measurement noise, to discard outliers, or to select the optimum subset of sensors.

Concerning the localization algorithm, a number of well-known approaches are available:

- It can be treated as a regression problem, which can be linearized and solved using classical least squares (LS). This is the most common approach in literature and practice but requires an initial guess of the position (iterative solution).
- It can be treated as a statistical estimation problem and any classical estimator (e.g., the maximum-likelihood linear estimator or MLLE) can be applied. If the error model is well suited, this approach usually gives a nonbiased solution close to the optimal Cramer–Rao lower bound. Under some simplification, this approach leads to the LS solution. In any case, an initial guess of the solution is needed (iterative solution).
- Some numerical methods set a new mathematical function that relates the unknown target position, the measurements, and a new parameter derived from the target position (e.g., the target range). These methods do not require any initial guess to work and are commonly referred to as *closed-form algorithms*. On the other hand, they usually introduce quadratic noise terms and usually the solutions are biased. An example of this approach is shown in Chan and Ho.[10]
- Another approach uses geometrical methods that algebraically manipulate the hyperbolic equations until they directly set an inverse problem relating the target position with the measurements. These models usually require more measurements, introducing quadratic and cubic noise as well. Like the numerical methods, these do not

require any guess for the solution (closed form) but typically are biased. An example is given in Schmidt.[11]

- Recently, machine learning approaches have been proposed. A dataset with all possible measurements, computed on a grid of points, is generated and then the best match between the incoming measurement vector and the dataset is found by the use of a k-nearest neighbors algorithm. This approach can be classified as *fingerprinting* and similar approaches were already tested in other application fields, such as indoor localization.

Methods requiring an initial guess can be very sensitive to its choice. The solution can result in large errors if it is far away from the true position. This problem is even more important when the considered scenario is global, contrary to typical localized MLAT deployments around airports. Thus, for any iterative method, the strategy to select the initial guess must be well-defined.

Preprocessing and Synchronization

Classical approaches assume that any station is synchronized with the reference station. Usually, time synchronization of the MLAT systems is achieved in two different ways: either by integrating a GPS receiver on each sensor or via a reference transponder in a known position that transmits radio frequency messages to all sensors.

Both methods are impractical in crowdsourced networks; there are only a few sensors using costly GPS synchronization, while the largest part feeds data without any synchronization mechanism.

A common, suboptimal, solution to overcome this fragmentation is the use of opportunity traffic: airplanes transmit their position encoded in the ADS-B messages. This means that if an airplane is in view of more sensors, the time biases between the stations can be easily estimated by inverting the equation:

$$m_i = \frac{\|\theta - \vartheta_i\| - \|\theta - \vartheta_1\|}{c} + b_{i,1} - b_1 + n_{i,1} \qquad (1)$$

where $\theta = (x, y, z)^T$ is the target position, $\vartheta_i = (x_i, y_i, z_i)^T$ represents the sensor position, c is the velocity of light, $n_{i,1}$ represents the difference between the realization of the noise of the sensor n and the sensor 1, $b_{i,1}$ represents the bias of the station i with respect to the reference station 1, and all positions in the equations are known.

This method has some limitations: it is difficult to achieve the synchronization of the complete network and the synchronization performance depends on the sensor measurement noise, the sensor position error, the aircraft ADS-B position accuracy, and the system geometry.

Moreover, the estimation of the clock offset at one moment usually is not sufficient, due to clock drift over time.

Clock drift comes from two main components of error: systematic fluctuations and random fluctuations.[12] Systematic deviation over time can be written as follows (approximating to the second-order term)[12]:

$$b_{\text{sys}}(t) = b(0) + f(0)t + 0.5Dt^2 \qquad (2)$$

where t is the time, $b(0)$ is the initial time offset of the clock, $f(0)$ is the initial frequency offset of the clock, and D is the frequency drift of the oscillator (it represents the systematic change of frequency due to a combination of internal factors, such as aging or production tolerances).

There are several solutions to this problem:

- *Sequential estimation* using a priori assumptions about the clock dynamic model and its noise characteristics, for example using Kalman filtering or a simple alpha–beta filter. The clock model and statistical properties or the errors are required.
- *Regression* to compute the parameters in (2). Only the regression formulation is required.
- *Fitting/smoothing* the sequence of measurements. No a priori information about the clock is required.

The first two methods use past data to extrapolate the future and can be used also in real-time applications. The third method is suitable only for offline batch localization.

Postprocessing and Aircraft Tracking

Having computed the aircraft positions for each received message individually, it is possible to significantly improve the accuracy and completeness of the results by applying postprocessing techniques. Considering that generally the trajectories of airplanes are smooth, or in accordance with dynamic constraints, it is possible to again use estimators to improve the quality of the final localization output. In particular, we can detect outliers, smooth the trajectory, interpolate the trajectory, or produce an initial guess for the following measurements. Known methods include sequential estimation by using a priori statistical information of the aircraft and filtering as proposed by Kalman, or fitting and smoothing methods, such as regression that can be used without a specific knowledge of the aircraft dynamic.

Competition Design

Design Goals

Our competition had four main design goals:

4. *Batch localization:* Our competition mimicked an offline, batch localization problem, where a flight has fully finished and complete knowledge is

available. Thus, exploiting the fact that aircraft move in predictable trajectories is explicitly allowed, as is the inclusion of "future data." This is in contrast to live online MLAT, where only knowledge up to the present point can be included. We plan to examine this problem in a future competition.

5. *Target metric:* The target metric was exclusively the localization accuracy, here loosely defined as difference between the ADS-B ground truth and the localization data.
6. *Integration of LocaRDS:* Use of a comparable dataset, developed specifically for comparison and competition, that offered significant amounts of training data for the participants, including TDoA, but also the received signal strength (RSS) as optional localization primitive.
7. *No computational requirements:* There were no requirements on execution time placed on the participants, in line with the batch localization goal.

Implicit goal: Sensor synchronization. Calibration and synchronization of the receiving sensors is effectively a prerequirement for all practical localization methods, and it is of particularly crucial importance in the uncontrolled crowdsourced setting. It can thus be considered a separate, implicit subgoal of our competition. Many of the existing localization solutions require very tight time synchronization, in particular those based on TDoA measurements. This is costly even in controlled industrial deployments but impossible to achieve consistently with the variety of modern crowdsourced sensors used by enthusiasts to feed OpenSky and similar networks. While some algorithms may be more or less robust against noise in the TDoA data, the better the synchronization, the better the end results will be.

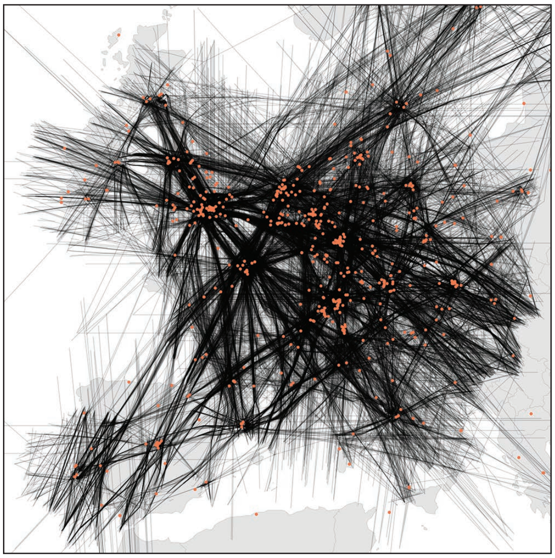

Figure 2. Illustration of the full LocaRDS dataset, with 50,865,291 aircraft positions (black lines) and 323 sensor positions (orange dots). In addition to geographic information, the dataset contains ToA and signal strength measurements for each position reported by an aircraft.

Competition Datasets

The offline competition and the first round of the online competition shared the same datasets (training and test) provided by the nonprofit research network OpenSky, with exclusively GPS-synchronized receivers, a subset of the LocaRDS dataset.[3] For the second round of competition, we used the full LocaRDS dataset, which included both synchronized and nonsynchronized receivers from the crowdsourced OpenSky Network. The competition ended before the full release of LocaRDS, including the test data, in March 2021.

Figure 2 illustrates the positions of the sensors and the measured aircraft trajectories. As can be seen, the focus of the dataset is on Europe, where the underlying data provider OpenSky has the best coverage with a sufficient number of sensors to conduct practical localization. Europe is also an interesting test case for cyberphysical research as it comprises many developed countries, typically with their own sovereign critical aviation infrastructure. Tables 1 and 2 provide the statistics for the different test and training datasets. For the full detailed description of the features and an in-depth discussion of the design of LocaRDS, see Schäfer et al.[3]

Sensor dataset. For all rounds, the relevant sensor information was provided in a separate CSV file. These covered a subset of 514 (offline/R1) and 716 (R2) receivers, respectively, which were feeding aircraft data to the OpenSky Network in the relevant time period in 2018. The sensor data comprised their type, capabilities (GPS-synchronized or not), and the precise location as provided by OpenSky. It is worth noting here that the

Table 1. Synchronized competition datasets.

Offline/R1	Positions	Flights	Size (MiB)
Training set 1	2,074,194	2,769	307.8
Training set 2	1,887,990	3,076	277.7
Training set 3	2,002,847	2,809	296.4
Training set 4	1,994,590	2,585	300.0
Training set 5	1,951,877	2,319	295.1
Training set 6	1,930,138	2,347	296.9
Training set 7	1,869,587	2,144	283.8
Test set	1,836,730	2,888	272.9

positions of the sensors are of varying accuracy. The sensor positions have only been entered by the user when the sensor was added to the network and there is no guarantee for correctness or accuracy. While some users report accurate positions for their antennas (e.g., measured with their smartphone), others just provide a rough estimate based on services like Google Maps. Some may even report wrong locations for privacy reasons.

Training datasets. Each dataset (a CSV file) contained the data recorded by the OpenSky Network over a duration of 1 h and has a (uncompressed) size of about 300 MB (only synchronized receivers) or 1 GB (all receivers). Each row represents the reception of one aircraft position report and contains the following information: a unique aircraft identifier, the Unix timestamp indicated when the message was received by OpenSky, unique identifiers of all sensors that received this signal, nanosecond timestamps from each of the sensors, signal strength measurements from each of the sensors, the position of the aircraft (latitude, longitude, height), and the barometric altitude of the aircraft.

Test datasets. The test or evaluation CSV was constructed in the same way as the training datasets. We then excluded the longitude, latitude, and geometric altitude of an arbitrarily chosen 10% of the flights, which the participants had to predict for the competition. This means the full position is present for all other aircraft in the dataset and can be used to synchronize the receiver clocks. Furthermore, the rough geometric height of the aircraft can be estimated based on the barometric altitude provided.

Evaluation Metrics

The metrics chosen for the scientific evaluation of the ALP should be as broadly applicable to the different scenarios and approaches as possible. In particular, in case of a formal competition, they should further have as low a complexity as possible so the users can easily understand how they are calculated. Finally, they should be robust against cheating.

Table 2. Nonsynchronized competition datasets.

Round 2	Positions	Flights	Size (GiB)
Training set 1	6,535,444	2,888	1.20
Training set 2	6,569,830	2,818	1.20
Training set 3	6,348,679	2,680	1.17
Training set 4	6,111,569	2,932	1.11
Training set 5	6,309,260	2,854	1.15
Training set 6	6,345,589	2,812	1.16
Training set 7	6,187,378	2,695	1.14
Test set	6,457,542	2,929	1.18

Localization accuracy. The key metric for research in localization in general is the accuracy with which the position of the target is predicted. While the utility of aircraft localization depends on the context and the use case, more accuracy is strictly better.

The root-mean-square error (RMSE) has been widely included as a standard metric to compare the predictive performance of different localization models (see Lymberopoulos et al.[4]). However, as a basic metric, we chose the truncated RMSE (TRMSE) between the real aircraft position as reported by the ADS-B ground truth and the contestants' predictions for our main ranking metric. This makes the metrics more robust against a small number of outliers with large position errors.

Dataset coverage. The second consideration concerns the coverage of the evaluation datasets, i.e., how many of the data points were chosen to be predicted. While ideally all samples would have a prediction, this is not practical for several reasons. For example, some methods may need initial samples to calibrate and also regularly recalibrate. Furthermore, there is also value in correctly choosing to not predict bad or uncertain samples in order to minimize outliers and improve the average localization performance. However, it is obvious that with equal localization accuracy, higher coverage is strictly better.

Concretely, we first required a minimum sample coverage of 50%, which should on average satisfy any nontactical applications of the ALP, i.e., those where update rates of aircraft positional information of more than 1 s are allowed. However, other values can sensibly be chosen based on the application requirements and also depending on the sensor coverage in a given geographical region.

Further considerations. Due to the variation in the distribution of uncertainty and quality of measurements in OpenSky, it is clear that there can be tradeoffs between coverage and accuracy, which we might want to capture to enable truly comparable scientific research. Besides requiring a minimum coverage, this tradeoff could also be quantified for a provided solution through applying a penalty directly toward the accuracy scoring. By assuming a fixed high localization error for any missing observation, the TRMSE is increased, incentivizing the contestants to provide a higher number of observations. However, the effectiveness of the penalty is highly dependent on the quality of the provided solution:

if the penalty is set below the TRMSE, it will actually improve the quality score and thus set a false incentive to leave out observations. As we were not aware of the quality of these solutions, we dropped the application of such a penalty and do not report it.

A second consideration is centered around the runtimes of the provided solutions. While the speed of localization algorithms is not crucial in our batch localization scenario, it may still be insightful to analyze. Variations in training times for ML-based solutions may impact the choice of algorithms in situations where regular retraining is required. Similarly, lightweight algorithms for distributed resource-constraint edge computing are a relevant application, for example for crowdsourced flight tracking networks. However, we decided to exclude the runtime from the initial scoring of the competition to enable participants to work on their own environments, making direct comparisons difficult.

Competition Execution

We briefly discuss the first, offline, stage of the competition before providing more detail about the significantly more successful online stages.

Offline Competition

Format. We first decided to conduct this competition offline and in person, in conjunction with a leading academic conference on sensor networks. As the chosen venue regularly hosts on-site competitions of varying nature, this provided us with several advantages: first, a fixed framework with a prespecified day and location; and second, embedding into a major conference would give additional awareness among a relevant academic community.

Participants from all backgrounds were free to join the contest, whether coming from academia, industry, government, or out of private interest. Two months before the on-site meeting, the competition attracted preregistrations from 42 contestants with 33 different affiliations.

As an additional incentive beyond the scientific challenge, there was prize money available.

Rules. The competition required every solution considered for the awards to be open sourced and their integrity and veracity subsequently verified by the organizers. Concretely, all source codes and additional datasets used to generate the results from the measurement data needed to be published under the GNU General Public License version 3 license. In addition, sufficient documentation needed to be provided to understand and reproduce the results.

Usage of any external datasets (e.g., weather data or tracking data from other sources) required explicit permission by the organizers one month prior to the on-site competition day and sharing with all other contestants. Contestants were only allowed to use their own original implementations. The simple reuse of existing code was explicitly disallowed. We encouraged individuals and teams of up to five persons from all backgrounds to register and participate. No affiliation with any of the organizers or their institutions was allowed.

Execution. Three months before the day of the on-site competition, we provided the training datasets to all competitors in form of CSV files, which included the ground truth of all aircraft locations. These datasets could be used by the participating teams to train their models in advance. For participating in the on-site competition, each team had to send at least one team member to the conference, where they received access to a nonlabeled evaluation dataset. They could be supported by their team members remotely and a voice channel to the competition site was constantly available. Overall, six teams with 11 members attended the evaluation day, with several thwarted last-minute due to visa and flight issues.

The teams had 9 h to find all locations of aircraft that were missing location information in the datasets. Every 3 h, the teams had to submit their intermediate results (as a CSV file) to the organizers present. The organizers calculated an indicator of the accuracy of their solution and provided an intermediate ranking. After 9 h, the teams submitted their final results and the final ranking was determined. It was possible to submit multiple times in each 3-h slot.

Online Competition

Format. We provided the labeled training datasets and the test datasets at the start of the competition period. The task was again to predict all locations of the aircraft flights that were missing location information. Each team (or individual) submitted their results for both rounds during the competition periods as a CSV file of a defined number of rows, which was uploaded to the AICrowd website. Afterward, an indicator of the accuracy (the 90% TRMSE) of their solution was immediately calculated and an intermediate ranking provided. When the competition time ended, the final ranking was determined using this leaderboard.

Rules. The rules with regards to open sourcing, licensing, external data used, and eligibility remained the same as in the offline competition. For full award eligibility, the quality of the solutions was required to be below 1,000 m TRSME. Between 1,000 m and 5,000 m, still half of the award money for a top five finish would be distributed, and none distributed for those above 5,000 m. For each round, the full awards were set at 4,000, 3,000, 2,000, 1,000, and 500 Swiss Francs (CHF), respectively.

Execution of first round: Synchronized. The first round ran from 15 June to 31 July 2020. In this round, all provided data were from GPS-equipped sensors, which simplified things significantly as the competitors did not necessarily have to put any effort into sensor time synchronization in order to achieve practical results.

For this round, we instituted a minimum coverage requirement of 50%. Thus, to be ranked, at least 50% of the missing aircraft positions had to be provided. Based on the rankings at the deadline, the underlying code was shared with the organizers by the top five teams/participants. We verified and ran the code independently in order to ensure that the entries were in accordance with the competition rules. There were no issues; thus, the winners were confirmed and the awards distributed. Overall, 46 teams with 75 participants contested this round.

Finally, we solicited feedback on the AICrowd forum that was set up for this competition. This resulted in several helpful comments by the participants in how to make the second round more engaging and remove some frustrations.

Execution of second round: Nonsynchronized. The second round ran originally from 15 September to 31 October 2020. We made three main changes compared to the first round:

- We instituted a minimum coverage requirement of 70%, since the results were better than expected in the first round and did not suffer much when requiring higher coverage.
- We restricted submissions to five per day in order to reduce submission spamming.
- We implemented a separate public and a hidden score in order to reduce overfitting. During the competition, feedback was provided on the scoreboard based on a fixed, arbitrarily chosen, 30% of all aircraft trajectories that needed to be predicted. Only after the end of the competition were the scores on the full test dataset calculated and shown. The winners of the second round were determined by this full ranking on the whole-test dataset.

However, we found that participation was lower, as was the eventual localization success, thus no awards were distributed. Building on our efforts, we again listened to the participants' feedback and refined the second round. Most notably, we released the open source code of the winning entries of round 1, as well as additional description of the training data. We reran the second round from 1 December 2020 until 31 January 2021, where it saw a good uptake of 19 teams with 26 participants and an excellent localization success comparable to the first round in orders of magnitude, despite the significantly more complex setting. After the same verification procedure conducted in the first round, the same number of prizes were awarded and the new solutions also open sourced on GitHub.

Competition Results

Offline Competition

Table 3 shows the on-site results of the top contestants, as well as OpenSky's previous reference implementation. The teams picked one, or a combination of several fundamental solution techniques: machine learning, traditional MLAT, ranging using the RSS, statistical regression, and analyzing the distribution of the data and deducting the position based on the historical data gleaned from the training sets.

Coverage targets varied significantly between 50% and 100%. While the two (at least partly) machine learning-based solutions targeted the whole dataset, all other teams chose to stay close to the 50% requirement, which is likely reflecting the difficulty of localizing a significant part of the real-world dataset measurements, even though they were from synchronized receivers. Yet, a combined ML/MLAT algorithm provided the best solution after 9 h, with 100% coverage and a 90% truncated RMSE of 11,915.81 m. Overall, we noted significant improvements throughout the evaluation day for all participants, making it likely that all approaches can be made more accurate.

In comparison, OpenSky's reference implementation based on traditional MLAT showed that good aircraft

Table 3. Localization results (TRMSE and coverage) of the offline competition (synchronized), compared to the MLAT reference implementation of OpenSky.

Rank	Solution type	Coverage (%)	TRMSE (m)
1	ML/MLAT	100	11,915.81
2	RSS ranging	62	22,654.32
3	Distribution analysis	52	36,505.45
4	Distribution analysis	51	44,818.18
5	Regression	50	50,708.14
Ref.	MLAT	45	682.38

localization results can be achieved with crowdsourced measurements based on cheap off-the-shelf hardware. It targeted measurements with at least three receivers, as is geometrically required for pure MLAT, and thus achieved a coverage of 45% and a TRMSE of 682.38 m.

Despite having three months preparation time with the training datasets, all on-site contestants were significantly less accurate than the traditional reference implementation based on MLAT used by OpenSky. This disappointing result shifted our approach toward an online competition to increase ease of participation and widen accessibility.

Online Competition

As discussed in the previous section, the offline competition results were not able to beat the existing reference implementations, partly by a wide margin. This changed significantly in the online competition, both in the first round (GPS-synchronized receivers only) as well as in the more difficult second round (all receivers, including nonsynchronized ones). Table 4 summarizes the results of both rounds.

First round. The results in the first round significantly beat our expectations in terms of accuracy. The three top results clustered within around 1 m of a 25 m TRMSE, with the fourth and fifth place still below 60 m.

All solutions were provided in Python/Cython and used fundamentally a variation of a classical MLAT approach. They differed, however, in their pre- and postprocessing (see Table 5). Common themes included the identification of the most reliable receivers and the most accurate measurements and tracks. These were then filtered and smoothed using a wide variety of methods from DBSCAN to Hooke Jeeves.

Table 4. Winning entries of the online competition.

Rank	Team type	Background	Coverage (%)	TRMSE (m) (Public/Full)	
Round 1 (synchronized, 50% minimum coverage)					
1	Solo	Academic	50	25.020	—
2	Team	Academic	50.2	25.817	—
3	Team	Independent	50	26.214	—
4	Team	Independent	50.2	33.544	—
5	Solo	Academic	50	59.467	—
Round 2 (unsynchronized, 70% minimum coverage)					
1	Solo	Independent	70	78.14	81.89
2	Team	Independent	70	90.13	98.37
3	Team	Academic	70	141.07	154.57
4	Team	Academic	72.3	157.32	171.66
5	Solo	Academic	72.3	1,497.99	2,392.53

We further observed that all teams are exactly at, or very close to the minimum coverage requirement of 50%, selecting only the most reliable data points for grading and ranking. This is an intended feature that is also relevant in real-world localization systems and illustrates the effort put into the optimal data selection.

“Sniping” was a practical issue we observed during the execution. As the deadline drew closer, more teams entered the competition and significantly more results were entered for grading, leading to a frantic final day and deadline experience for everyone. In terms of team types, we had a mix of participants with academic affiliation and independent competitors; most were teams of two or three but the top contestant was a single individual.

Second round. The second round also exceeded our expectations significantly. As only about 15% of the available sensors were GPS-synchronized, the problem set posed a much harder challenge, which the participants solved with two excellent results below 100 m and two very good ones below 200 m. Round 2 was also won by a solo participant, this time without academic affiliation.

Participants again approached the 70% coverage requirement. The difference between the public and hidden score was between 4.8% and 9.6% for the top four. This means no significant overfitting and importantly it did not affect the final ranking order.

Methodologically, the focus was on accurate sensor synchronization, which the participants attempted to do either locally or globally and with different choices of good sensors. Interestingly, the winning solution incorporated open sourced ideas from round 1, illustrating the power of an iterative and open process.

Technical Evaluation

In this section, we discuss the technical outcomes of the competition. Table 5 shows the methods used both in round 1 and round 2 of the online competition. Notably, all teams focused exclusively on the TDoA option and excluded the RSS that was also available in the dataset. Regarding the applied MLAT solutions, teams used closed-form approaches, which are generally lighter and simpler than LS or MLLE, with all refinement outsourced to the postprocessing.

One of the goals of the competition was to find new strategies (or new combinations of strategies) to solve the MLAT problem in a demanding unsynchronized real environment, as required in round 2. This meant an unprecedented focus on effective pre- and postprocessing.

The winning solution of round 2, discussed in detail in Markochev,[13] uses the training data to estimate the fixed measurement offset in the sensors, and to estimate the speed of the signal in the troposphere (instead of using the classical approximation of the speed of light). In postprocessing, the trajectories were smoothed with a spline algorithm. This final step is the major contribution to improve accuracy and availability of the solutions. It is noted that the proposed method is not suitable for any real-time application but can nonetheless provide accurate localization for less time-sensitive security applications.

Another interesting approach is a different way of obtaining local synchronization used by the second-place team of round 2: instead of classical sequential algorithms, they used a heavier linear regression of neighbors points to compute offset and drift terms. This approach is agnostic to the system model and can be also used in real-time applications.

An important different approach was to first use linear regression for time synchronization, then solving the MLAT problem with LS for subsets of four stations. This involved selecting the best subset of sensors to calculate the solution (this is a sanity check of the measurements similar to integrity monitoring processes in the GNSS field). Additionally, it computes the aircraft altitude using light gradient boosting machines because the MLAT problem is usually ill-conditioned in the vertical dimension and the inversion usually produces large errors in altitude.

Finally, while the obtained results are an order of magnitude worse than the top four, a radically different ML approach was proposed by the fifth-placed team. After global synchronization using linear regression, a grid of possible positions is defined and a cost function on this grid is minimized. This circumvents inverting the problem.

The winning solution showed 78.14 m and 81.89 m TRMSE on the public and the full LocaRDS datasets, correspondingly, both with 70% of coverage. These are very impressive results, which can aid the development of accurate cyberphysical security systems in this space. Distribution of location errors for the full dataset is shown Figure 3.

Table 5. Pre- and postprocessing methods used by the winning entries.

R1/R2	Preprocessing	Postprocessing
R1		
P1	Minimize offsets with training data.	Fit spline to trajectory, identify good quality timings.
P2	Correct offsets, exclude bad sensors, first guess using ML. (gradient boosting trees).	Filter for aircraft with sufficient sensor data (nine triplets), smooth trajectories.
P3	Calculate offsets.	Filter outliers, Hooke Jeeves, linear interpolation, fit a/c track.
P4	Filter outliers.	Filter for direction and density with DBSCAN. Localized extrapolation with Huber regression.
P5	Identify good sensor combinations from training data.	Identify trajectories, filter outliers.
R2		
P1	Estimation of effective signal wave velocity, including altitude dependency. Iterative synchronization of good stations first.	Huber regression plus graph-based filter.
P2	Local adaptive sensor synchronization. Sensor outlier filtering.	Filtering of predicted outliers. Track smoothing with low-pass filter. Interpolation of points with <4 measurements.
P3	Global sensor synchronization. Calculate measurements with four or more sensors.	Interpolate missing points.
P4	Barometric altitude estimation. Sensor synchronization plus error minimization.	Basic filtering/trajectory smoothing.
P5	Models clock drifts of sensors from training data.	Interpolates between predictions.

Lessons Learned

There are several insights from our process of designing and executing a large-scale competition on aircraft localization and we believe these can be applied to other potential competitions in the (cyberphysical) security context:

- *Offline versus online competition:* There are several practical drawbacks of running on-site data science competitions, which from our experience outweigh the advantages even in the prepandemic world. These range from the costs and environmental footprint of experts traveling from around the world to issues with visas and last-minute weather-related cancelations. While the appeal of a live competition is tempting and face-to-face exchanges facilitate creativity and foster competition, these features do not compensate for the sheer efficiency of online contests. Our online competitions saw both a significant increase in contestants and improved results in orders of magnitude.
- *Obstacles for industry participation:* We overestimated the interest of industry entities involved with MLAT or other localization approaches in an open competition. While in theory, competitions can be attractive to companies as they can show off the quality of their work, it is also a risk factor in the marketplace in case their solution performs worse. Another significant factor that prevents industry is the protection of intellectual property surrounding the solution or the implementation, even when the source code does not need to be opened.
- *Value of open sourced localization code:* Our goal of advancing practical aircraft localization and making it accessible beyond the proprietary industry systems, where it is currently prevalent, included the open sourcing of the code from the very beginning. Our experiences with the extension of round 2 illustrate the power of this open and collaborative approach; all top results improved significantly, often by learning from the published approaches (code, documentation, and papers, such as Figuet et al.[14]) of round 1. In the meantime, new projects have popped up on Github (for example, https://github.com/radoslawkrolikowski/adsb-flight-localization), illustrating their own methods and building on the code and data published after the conclusion of round 2. In one development, the data have been used for a secure multiparty computation implementation of MLAT using the Python package MyPC.[15]
- *Improving existing algorithms:* Analyzing the different solutions paved the way for new localization research, merging insights from different fields of research, such as machine learning and geometrical solutions. Moreover, it was clearly shown that a big improvement of the ALP solution can be obtained applying well-tailored postprocessing and data smoothing.
- *Underutilization of RSS:* By measuring the strength of an incoming signal and by knowing or estimating the transmit power of the aircraft, the distance between sender and receiver can be estimated based on their difference, i.e., the *path loss*. One notable drawback of the RSS is, however, that its accuracy depends on many potentially unknown factors, the radio environment, and (analogous to TDoA) the measurement resolution. Besides direct ranging measurements, RSS-based localization approaches often use indoor radio maps. This is intuitively more difficult to recreate with fast-moving aircraft spread out over long distances in highly dynamic environments (due to weather, buildings, and other influences). Building radio maps further requires a setup phase and separate infrastructure, which cannot be offered through a reference dataset. These are likely reasons why the RSS has been underutilized in our competition.
- *Ongoing evaluation:* The widely varying approaches both in the literature and our competition show that there is still much room for improvement in the theoretical development and practical implementation of solutions for the ALP. The results show that the contestants have significantly improved on the OpenSky reference implementation. We expect further improvements through open research with the released data, code, and scientific reports to a wider audience.

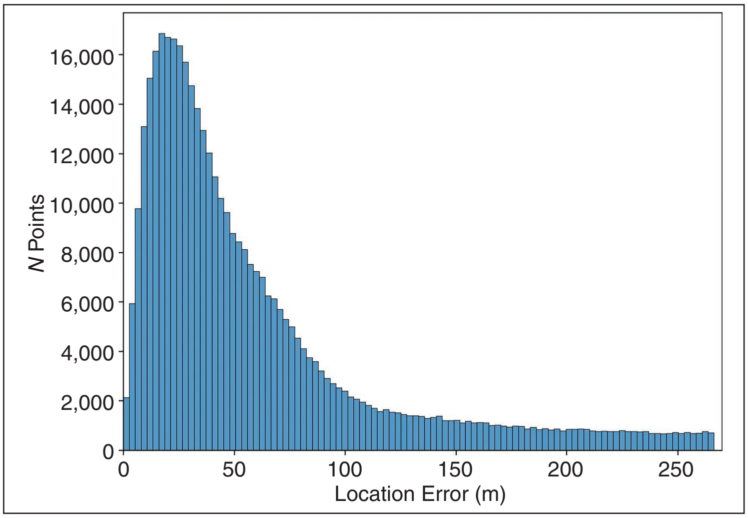

Figure 3. Distribution of aircraft location errors obtained for the full LocaRDS dataset in the winning solution.

Crowdsourced air traffic trajectories are used in many areas of science and commerce. Their veracity strongly depends on the quality and accuracy of the data, which is derived from an unauthenticated ATC system. Using cyberphysical features and physical-layer security, such as MLAT and localization, can verify the data and consequently identify attacks and anomalies. Thus, improving the state-of-the-art of aircraft localization is crucial in order to improve the underlying cyber-physical system.

In this article, we have presented the design and execution of a multistage open competition on solving the ALP in this context. The 72 participating teams reached a highly practical localization accuracy of up to 25 m in a fully GPS-synchronized setting and 78 m in a largely unsynchronized setting, with the cheapest possible receiver hardware (USD$50 and less). By comparing online and offline competitions, many novel lessons were learned for future scientific challenges, including real-time localization. ■

Code and Data Availability

The complete code of the aircraft localization competition, including the winning entries, has been made available at https://github.com/openskynetwork/aircraft-localization. The training and test data have been made permanently available on Zenodo: https://zenodo.org/record/4739276. The competition websites are available at AICrowd (https://www.aicrowd.com/challenges/cyd-campus-aircraft-localization-competition/) and the OpenSky Network Association (https://competition.opensky-network.org/).

References

1. M. Strohmeier, I. Martinovic, and V. Lenders, “Securing the air–ground link in aviation,” in *The Security of Critical Infrastructures*, M. Keupp, Ed. Cham, Switzerland: Springer-Verlag, 2020, pp. 131–154.
2. P. Aldhous and C. Miller, “How open-source intelligence is helping clear the fog of war in Ukraine,” *Buzzfeed News*, Mar. 2022. [Online]. Available: https://www.buzzfeednews.com/article/peteraldhous/osint-ukraine-war-satellite-images-plane-tracking-social
3. M. Schäfer, M. Strohmeier, M. Leonardi, and V. Lenders, “LocaRDS: A localization reference data set,” *Sensors*, vol. 21, no. 16, Aug. 2021, Art. no. 5516, doi: 10.3390/s21165516.
4. D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned,” in *Proc. ACM 14th Int. Conf. Inf. Process. Sensor Netw.*, 2015, pp. 178–189, doi: 10.1145/2737095.2737726.
5. A. Costin and A. Francillon, “Ghost in the air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices,” in *Proc. Black Hat USA*, Jul. 2012, pp. 1–10.
6. D. McCallie, J. Butts, and R. Mills, “Security analysis of the ADS-B implementation in the next generation air transportation system,” *Int. J. Crit. Infrastructure Protection*, vol. 4, no. 2, pp. 78–87, Aug. 2011, doi: 10.1016/j.ijcip.2011.06.001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1874548211000229
7. B. R. Haines, “Hacker + Airplanes = No good can come of this,” in *Presentation*, Las Vegas, NV, USA, 2013.
8. G. Galati, M. Leonardi, P. Magarò, and V. Paciucci, “Wide area surveillance using SSR mode S multilateration: Advantages and limitations,” in *Proc. IEEE Eur. Radar Conf. (EURAD)*, 2005, pp. 225–229, doi: 10.1109/EURAD.2005.1605606.
9. M. Strohmeier, I. Martinovic, and V. Lenders, “A k-NN-based localization approach for crowdsourced air traffic communication networks,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1519–1529, Jun. 2018, doi: 10.1109/TAES.2018.2797760.
10. Y. T. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location,” *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994, doi: 10.1109/78.301830.
11. R. O. Schmidt, “A new approach to geometry of range difference location,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-8, no. 6, pp. 821–835, Nov. 1972, doi: 10.1109/TAES.1972.309614.
12. G. Lorenzo and T. Patrizia, “Detection of atomic clock frequency jumps with the Kalman filter,” *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 59, no. 3, pp. 504–509, Mar. 2012, doi: 10.1109/TUFFC.2012.2221.
13. S. Markochev, “Aircraft localization using atc data with nanosecond precision from distributed crowdsourced receivers,” *Eng. Proc.*, vol. 13, no. 1, Jan. 2022, Art. no. 12, doi: 10.3390/engproc2021013012.
14. B. Figuet, R. Monstein, and M. Felux, “Combined multilateration with machine learning for enhanced aircraft localization,” *Eng. Proc.*, vol. 60, no. 1, Dec. 2020, Art. no. 2, doi: 10.3390/proceedings2020059002.
15. B. Schoenmakers, “MPyC – Python package for secure multiparty computation,” in *Proc. Workshop Theory Pract. MPC*, 2018. [Online]. Available: https://github.com/lschoe/mpyc

Martin Strohmeier is a senior scientific project manager at the Cyber-Defence Campus, 8005 Zurich, Switzerland and a visiting Fellow at the University of Oxford, OX1 2JD Oxford, United Kingdom. He is interested in wireless security, critical infrastructures, and adversarial machine learning. Before coming to Oxford for his Ph.D., Strohmeier received an M.Sc. from Technical University of Kaiserslautern, Germany and worked as a researcher at Lancaster University’s InfoLab21 and Lufthansa. Martin is also a cofounder and board member of the OpenSky Network. Contact him at martin.strohmeier@armasuisse.ch.

Mauro Leonardi is an assistant professor at the University of Rome Tor Vergata, 00133 Rome, Italy, teaching "Satellite Navigation and Surveillance Systems" and "Radar and Localization." His main research activities are focused on radar, satellite navigation, signal processing, positioning, and localization algorithms. Leonardi received a Ph.D. from the University of Rome Tor Vergata. Contact him at mauro.leonardi@uniroma2.it.

Sergei Markochev is an independent researcher in the commercial sector, London, UK. His main research interests include machine learning, applied mathematical modeling, and cloud computing. Markochev received a Ph.D. in physics obtained from the Moscow Institute of Physics and Technology. Contact him at sergey.markochev@gmail.com.

Fabio Ricciato works for the European Commission, DG Eurostat, L-2721 Luxembourg City, Luxembourg, working on various projects around the modernization of official statistics, exploitation of novel data sources, privacy-preserving approaches to cross-institutional data analytics, and analysis of mobile network operator data. Among his interests, since 2015 he has contributed (in his personal capacity) to the OpenSky Network. Ricciato received a Laurea in electrical engineering and Ph.D. in information and communication technologies from University La Sapienza, Italy. Contact him at Fabio.Ricciato@tutanota.com

Matthias Schäfer is lecturer and researcher at the Distributed Computer Systems Lab at the University of Kaiserslautern, 67663 Kaiserslautern, Germany. He is cofounder of the OpenSky Network and founded SeRo Systems in 2014. Schäfer received a Ph.D. in security of mobile systems from the University of Kaiserslautern, Germany. Contact him at schaefer@cs.uni-kl.de.

Vincent Lenders is director of the Cyber-Defence Campus, 8005 Zurich, Switzerland and head of the Cyber Security and Data Science Business Unit at the Science and Technology branch of the Swiss Federal Department of Defence, 603 Thun, Switzerland. He is a cofounder of the OpenSky Network. His research interests lie at the intersection between cybersecurity, data science, networking, and crowdsourcing. Lenders received a Ph.D. in electrical engineering and information technologies from ETH Zurich and was a postdoctoral research Fellow at Princeton University. Contact him at vincent.lenders@armasuisse.ch.