proceedings

MDPI

Proceedings

Improved Testing of AI-Based Anomaly Detection Systems Using Synthetic Surveillance Data [†]

Antoine Chevrot [1,*], Alexandre Vernotte [1], Pierre Bernabe [1,2], Aymeric Cretin [1], Fabien Peureux [1] and Bruno Legeard [1,3]

[1] FEMTO-ST Institute, University Bourgogne Franche-Comté, CNRS, 16 Route de Gray, 25030 Besançon, France; alexandre.vernotte@femto-st.fr (A.V.); pierbernabe@simula.no (P.B.); aymeric.cretin@femto-st.fr (A.C.); fabien.peureux@femto-st.fr (F.P.); bruno.legeard@femto-st.fr (B.L.)

[2] Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway

[3] Smartesting Solutions & Services, 25000 Besançon, France

[*] Correspondence: antoine.chevrot@femto-st.fr

[†] Presented at the 8th OpenSky Symposium 2020, Online, 12–13 November 2020.

Published: 1 December 2020

Abstract: Major transportation surveillance protocols have not been specified with cyber security in mind and therefore provide no encryption nor identification. These issues expose air and sea transport to false data injection attacks (FDIAs), in which an attacker modifies, blocks or emits fake surveillance messages to dupe controllers and surveillance systems. There has been growing interest in conducting research on machine learning-based anomaly detection systems that address these new threats. However, significant amounts of data are needed to achieve meaningful results with this type of model. Raw, genuine data can be obtained from existing databases but need to be preprocessed before being fed to a model. Acquiring anomalous data is another challenge: such data is much too scarce for both the Automatic Dependent Surveillance–Broadcast (ADS-B) and the Automatic Identification System (AIS). Crafting anomalous data by hand, which has been the sole method applied to date, is hardly suitable for broad detection model testing. This paper proposes an approach built upon existing libraries and ideas that offers ML researchers the necessary tools to facilitate the access and processing of genuine data as well as to automatically generate synthetic anomalous surveillance data to constitute broad, elaborated test datasets. We demonstrate the usability of the approach by discussing work in progress that includes the reproduction of related work, creation of relevant datasets and design of advanced anomaly detection models for both domains of application.

Keywords: anomaly detection; synthetic data generation; AI testing; ATC; VTS

1. Introduction

Vehicle traffic surveillance has become increasingly reliant on cooperative and dependent technologies for its operations, in order to support the ever-growing traffic load. Whether ADS-B [1] for Air Traffic Control (ATC) or the Automatic Identification System (AIS) [2] for Vessel Traffic Surveillance (VTS), the underlying mechanisms are similar: aircraft/vessels periodically broadcast their position, velocity and other critical data while ground stations pick up the messages, process them and forward the information to the traffic controller.

The reliance on vehicles communicating surveillance information, coupled with alarming cyber security issues [3] (neither ADS-B nor AIS provide encryption or identification), exposes traffic surveillance to new cyber security threats, and especially to false data injection attacks (FDIAs), in which an attacker modifies, blocks or emits fake surveillance messages to dupe controllers and surveillance systems. There have been cases of fishing boats voluntarily turning off their transponders and/or changing their identity before fishing in prohibited areas while remaining unnoticed. Similarly in ADS-B, Strohmeier et al. [4] collected recent incidents in the ATC domain. There is therefore a strong need for detection systems that are capable of consistently dismissing false information and preserving confidence in surveillance technologies such as ADS-B and AIS.

There has been a growing interest in conducting research on anomaly detection systems that address these new threats [5]. Some of these systems are machine learning (ML)-based and thus require significant amounts of data to achieve meaningful results. It is indeed critical for ML researchers to have access to reliable and genuine data sources to train their models. For ADS-B data, the OpenSky Network is one of the most used references; in terms of accessibility and data history, surveillance data can be easily obtained from almost anywhere in the world. This kind of open-source database does not exist for AIS, but the US Coast Guard has made a reasonable amount of data available (https://marinecadastre.gov/ais/).

Access to genuine data is key when building anomaly detection models. Another challenge is to properly evaluate the models' detection capabilities, and this requires enough meaningful anomalous data to be obtained. To date, researchers have crafted their test data sets by modifying recorded surveillance communication flows by hand. Inevitably, the resulting anomalies are quite straightforward (e.g., a sudden altitude drop, or a progressive latitude drift) and hardly measure the efficiency of the detection models against future, more elaborate attacks. Indeed, while the few reported incidents of tampering with surveillance data have been simple attacks, it is undeniably urgent to foresee and prepare for more elaborate FDIA scenarios that involve replayed surveillance data and realistic trajectory modifications.

This paper presents an approach that automatically generates synthetic but realistic anomalous surveillance data to improve the efficiency and effectiveness of ML-based detection systems. Moreover, it builds upon existing libraries and ideas to allow the feeding of ADS-B data from OpenSky to a ML model. The goal is to offer ML researchers the necessary tools to facilitate the access and processing of data and allow them to design the types of attack scenario they require themselves, while leaving the actual generation of synthetic anomalous data to an existing framework. The paper also demonstrates that the approach will also be applicable to the maritime domain, on the condition that an OpenSky-like database emerges.

2. Related Work

We can find an overview of anomaly detection in [6] specifically using deep learning. Anomaly detection is often viewed as an unsupervised task and therefore many applications can be found in the literature. For example, anomaly detection applications can be found in smart grids and water distribution systems [7] or surveillance videos [8]. Fully-supervised approaches for anomaly detection usually ignore data without any labeling [9]. They perform well on datasets with an equal distribution between training and testing sets compared to unsupervised or self-supervised approaches, but fall behind once the testing data introduce new types of outliers [10]. Overall, the advances in neural networks and deep learning for anomaly detection show that these techniques have great advantages over traditional machine learning, mainly due to their increased performance with large amounts of data. This paper focuses on data quality and the quality of its access; thus, we focus on these type of techniques.

Both ADS-B and AIS data fall into the category of multivariate time-series. Some of the challenges listed in [6] to detect anomalies in this type of data using deep learning models data include the following:

1. A lack of a defined pattern in which an anomaly occurs;
2. Noise within the input data;
3. As the length of the time series data increases, the computational complexity also increases.

These three issues need to be taken into consideration for both the data processing and the choice of model architectures. They apply to both ADS-B and AIS protocols, and some answers to these challenges can already be found in published papers. Concerning the ATC domain, a large number of works on anomaly detection using solely ADS-B data revolve around the use of deep learning techniques and specifically auto-encoders. To tackle the temporal nature of the data, a large amount of efforts are being devoted to the use of the recurrent neural network (RNN) and specifically long short-term memory (LSTM) units (please refer to the survey presented in [11] for a more comprehensive overview of advances in anomaly detection in the ATC domain).

Current maritime traffic management and surveillance work has concentrated on predicting destinations and trajectories and detecting irregularities in observed vessels, regardless of the nearby vessels' details [12]. Other related works model vessel activity using probabilistic models from historical AIS knowledge; e.g., Auslander et al. [13] construct hierarchical neural networks, or Markov models, such as Markov chains and Markov logic networks [14]. Nguyen et al. [15] have explored multi-task ML; they used a static embedding method that completes the missing data and regularizes the frequency to learn a probabilistic model of typical vessel trajectories from the AIS data stream.

All these efforts in the domain of anomaly detection in both domains results in a need to have proper access to both training and testing data for deep learning models. The present article builds upon a previous paper by the same authors [16], coupled with the use of a domain-specific language allowing a better control over anomalies as well as labeling and massification. In addition, we present a more general problem formulation including the VTS domain and the approach of data generation for AI training and testing compared to the solely ATC testing perspective defined in previous works.

3. ADS-B Data Architecture

In order for a reasonable environment in which to train and evaluate machine learning models to be developed, it is first necessary to ensure their access to data is guaranteed. In this section, our full ADS-B Raw data architecture is described, from the gathering of the data in a historical database fed by collaborative sensors to the generation of testing data in the context of anomaly detection models.

3.1. Data Gathering

In order to create our large-scale dataset of ADS-B raw data, we used the history database from the online flight tracking network OpenSky [17].These data are collected by cooperative ground stations, and the network is mainly maintained by enthusiastic private individuals. As a result, the quality of the data often depends on the performance of the acquisition material used by the seeders. State vector data are the data from the OpenSky betwork with verifications made and with erroneous messages already filtered out. These data did not really fit our requirement of data that arise directly from sensors with no filters to train models with no bias introduced by uncontrolled data cleaning; nonetheless, some cleaning steps were necessary and are detailed in Section 3.2.

Traffic [18] is an open-source Python-based tool allowing users to query the OpenSky network database. It simplifies the data gathering process by aggregating the different types of ADS-B messages (position, velocity and identification) that we use later for the model. In a recent update of the library to which the authors contributed, the data can be taken from the raw tables of the Impala database, which considerably helps in the building of the data-pipe to create multivariate time-series.

3.2. Preprocessing

This section presents all the steps that we applied to the raw data. To introduce the least bias possible in the data cleaning process, the operations performed on the data, apart from the time-series handling and windowing developed here, were kept minimal and focused on the different discrepancies noticed in the raw data (see Figure 1).

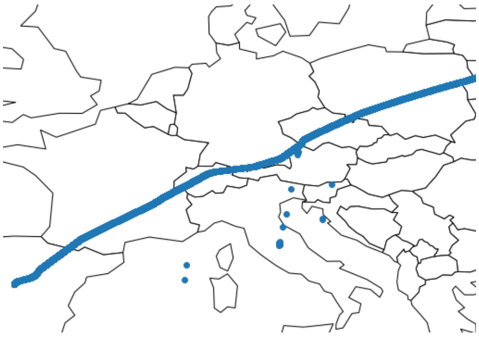

Figure 1. Data discrepancies on a flight between Madrid and Moscow.

The main cleaning part was performed with two local outlier factors (LOF) used back to back and trained on many flights: the first one had a low number of neighbors (∼5–10) to detect isolated local outliers, and the second one had a higher number of neighbors (∼20–30) to filter out consecutive outlier messages, typically the result of a faulty encoding, that were not detected by the first LOF due to their low deviation of density with respect to their closest neighbors (which are also outliers).

3.3. ML Architecture

For the ML architecture we aimed to evaluate for the anomaly detection, we needed to break-down the flights in three different phases. We used the fuzzy logic algorithm first used by Sun et al. [19]. This helped to identify the climbing, cruising and descending phases based on the relationships between the altitude, speed and vertical rate taken from the ADS-B data. From these phases,the B and C flight points used in the model developed by Habler can easily be determined or the flight split to be used directly in a forest model (see Section 6 for more detail).

4. Automatic Identification System (AIS) Data Architecture

The Automatic Identification System (AIS), as a tracking system, supports maritime traffic surveillance and control. Having AIS activated is mandatory for all vessels with 300 GT (gross tonnage) and above, for all ships which are engaged in international journeys and for all passenger ships. For all other vessels, cheaper and less powerful transmitters are suggested but not required. An AIS transmitter broadcasts vessel information, such as its location, speed or heading. Not all AIS messages contain the same information, and static information, such as the vessel type, the vessel name or its size, is transmitted every 6 min. The nearby ships and coastal stations can receive the messages in a range of about 20 to 40 km. This range limitation is due to the Earth's curvature and the height at which the antenna is installed on ships. Furthermore, the AIS messages are detectable with a satellite that has the ability to receive signals from anywhere in the world.

4.1. Data Collection

Creating a large-scale dataset of raw data, such as in ADS-B, is a challenge in AIS. The data are mostly collected by the coastal base stations and satellites operated by companies and coastal guards. As they are considered critical information, states are often reluctant to open this data freely. When the AIS data are available, they have been preprocessed and cleaned. However, to create a model that can be generalized to every part of the world, it is necessary to use raw data. We can cite two organizations that work to make AIS data available: the first is "Global Fishing Watch", which tries to contact governments directly to access their data, but the raw data they use in their research is not public; the second one is "AIS hub", which aims to become a raw AIS data sharing center where any parties can get memberships to share the data they collect on their own AIS receiving station. Although the project is interesting, the coastal coverage is still limited, and data from satellites are necessary to cover the non-coastal areas. In our project, we obtain raw data from Statsat, a company that owns satellites. These data are collected by four satellites, but they do not cover the whole Earth simultaneously. Moreover, the AIS protocol's limited capacity causes a great deal of message loss due to collisions in dense areas such as the EU, China, or the US coast. To avoid this problem, we integrate data from ground stations from Norwegian coastal guards into this dataset. The final dataset consists of over 300 million AIS messages.

4.2. Preprocessing

The data provided by Statsat use the Comment Block (CB) extension of the IEC 61162 standard to exchange AIS data together with additional information such as the complete timestamp, configuration and status messages.

The data were decoded and stored inside a NoSQL database. The database was then completed with the coastal guard data. Furthermore, information such as the distance to the satellite and distance to the nearest port was added. Finally, static information was added such as the vessel type, power engine, length and tonnage obtained from EU Fleet Registration. The datasets used to train the models were extracted from the database based on their timestamps, positions and message types. Incoherent, duplicated and incomplete messages were removed. We then used different algorithms to extract relevant sequences of messages—for example, the last 100 messages of a vessel before losing connection—or random sequences passed to the anomaly generator.

5. Synthetic Anomalous Data Generation Framework

We propose a framework that generates synthetic anomalous data that cover the whole spectrum of seen and anticipated attack scenarios, as compiled in a taxonomy [20] (dedicated to ADS-B but applicable to AIS). This includes vehicle spoofing and disappearance, but also ghost vehicle injection (adding a fake track to the existing flow, either obtained from a previous flow or fully crafted from scratch), vehicle flooding (injecting multiple fake track to overload the surveillance situation picture) and virtual trajectory modification. The framework is built upon an existing FDIA testing framework called FDI-T for False Data Injection Testing [21], designed for surveillance systems testing. Several extensions have been developed for FDI-T in order to turn it into an efficient and intuitive tool for training and testing ML models.

5.1. Background: FDI-T—An FDIA Testing Framework

FDI-T (which was conjointly developed by Smartesting (https://www.smartesting.com) and Kereval (https://www.kereval.com/)) is a testing framework that allows domain experts to design FDIA scenarios by altering (creating, modifying and deleting) recorded legitimate surveillance messages in a fruitful, scalable and productive manner. The altered recordings are then played back (with respect to time requirements) onto real surveillance systems to simulate an attacker tampering with the surveillance communication flow. The approach mainly aims to evaluate the of a system against potential security and safety anomalies, and more precisely to FDIAs.

As depicted in Figure 2, the architecture of the FDI-T framework is composed of five components:

① **Data acquisition**. This component collects legitimate surveillance messages (in Beast or SBS (site base station) formats for ADS-B, and in the NMEA 0183 format for AIS). obtained either from the Internet or a Mode S receiver. Data take the form of a recording—i.e., a sequence of surveillance messages—ordered by reception time.

② **FDIA scenario design**. The domain expert defines FDIA scenarios to be applied on a recording obtained via the data acquisition component. FDIA scenarios have various parameters, such as a time window, a list of targeted aircraft, triggering conditions, etc. Once designed, FDIA scenarios are translated into a set of alteration directives, which is the output of the component (an alteration directive is a small modification of the initial recording, usually doable by hand). The scenario design is specified via a domain-specific language (DSL), which is further specified depending on the targeted domain (ATC or VTS). For example, a trajectory modification scenario can be defined as below:

```
scen:   alter all_vessels satisfying "filt" at 10 minutes triggered_by "trigg"
            with_waypoints [ (2.476,47.69) at 72 minutes, (2.319,48.01) at 320 minutes ]
filt:   F (LATITUDE > 45.18) and G (LATITUDE > 42.475)
trigg:  eval when (VESSEL.SPEED_OVER_GROUND <= 8.2)
```

This scenario ***scen*** aims to alter the AIS messages of vessels that satisfy filter ***filt***, starting 10 min after the recording's first message, and according to the alteration trigger ***trigg***. The filter ***filt*** targets vessels that sailed at least once (i.e., F stands for *eventually*)at a latitude higher than 45.18 and never sailed under latitude 42.475 (i.e., G for always). The trigger ***trigg*** marks vessels for alteration when their reported speed over ground equals 8.2 knots or under. Finally, the alteration consists of modifying the vessels' trajectory to make them pass two waypoints at a certain time (related to the recording starting time).

③ **Alteration engine**. This component takes as its input a set of original sub-recordings, a set of alteration directives and a correspondence matrix that defines which alteration scenario should be applied on which given sub-recording. It then produces altered sub-recordings in the system input format (regarding the ATC domain, this component is thoroughly described in [16]).

④ **Execution engine**. The obtained altered ATC sub-recordings are fed to the surveillance system as if it were receiving live surveillance messages.

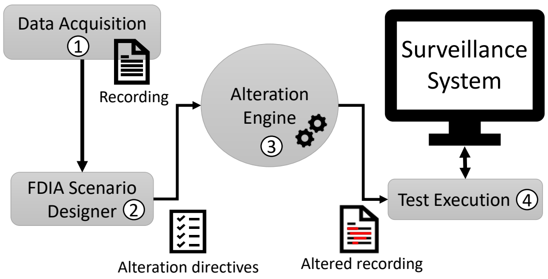

Figure 2. FDI-T Framework Architecture.

5.2. FDI-T for the Training and Testing of Anomaly Detection Models

Although FDI-T was initially designed for surveillance systems testing, it also enables the generation of datasets for training and testing ML-based anomaly detection systems. We describe below the four features that were added to the existing testing framework in order to turn it into an efficient synthetic data generation tool that is capable of generating hundreds of thousands of data samples with very little effort.

Labelling. A positive side-effect of generating synthetic anomalies is the ability to obtain well-balanced datasets, opening the possibility of supervised learning models. Storing alteration information (which messages were tempered with, what properties were altered) is achieved by directly appending an alteration bitmask at the end of each surveillance message. Each bit of this bitmask is associated to a message field: if the field content has been modified by FDI-T, its corresponding bit is flipped to 1, and flipped to 0 otherwise. For example, given an eight-field message in which only the latitude and longitude (fields Nos. 3 and 4 in the message) were tampered with, the bitmask 00110000 wold be appended to the original message.

Massification capabilities. While test engineers usually aim for a sharp test suite that satisfies all test requirements with as few test cases as possible, data scientists need a significant amount of data in order to properly train and test models. FDI-T in its current state allows users to create one altered recording at a time, and creating a dataset is realistically unfeasible in these conditions. Thus, we added massification capabilities to the DSL: users may first define variables containing lists of values, and thereafter reference the variables inside a scenario instead of a value. The framework creates as many single valued scenarios as there are values in the variable. In the case of a scenario with several references to variables, a single valued scenario is created for each combination of variables values. An example of a massified false alarm scenario is presented below:

let *$start* = {0,25,40,112}, let *$end* = {120,215,230},
alter plane from *$start* seconds until *$end* seconds with_values SQUAWK = 7700

It contains two variables, *$start* and *$end* (containing four and three values, respectively), which are used to define the alteration start and end time. The framework therefore generates 4 × 3 = 12 single valued scenarios leading to 12 altered recordings, each having a different start and end time.

Recording-Agnostic scenarios. Initially, the test engineer had to have some knowledge about the recording to be altered. Again, this is not acceptable in an ML model training/testing context, and two additions to the DSL were needed to fix the issue. On the one hand, relevant information (about the recording and vehicle) is gathered and stored as constant in the DSL; on the other hand, it was made possible to use value offsets when altering vehicle properties and defining waypoints. Thus, the initial values can be unknown by the user. In addition, such scenarios are thus made applicable to different vehicles without any change being required. The aforementioned additions are illustrated below:

alter plane at 0.5 * **REC_DURATION seconds with_waypoints [**
(>> 0.5,∼) **with_altitude** 2000 **at** 0.2 * **ALT_TW**, (>> 0.5,∼) **with_altitude** 2000 **at** 0.8 * **ALT_TW]**

The scenario contains two constants representing the recording duration and the alteration time window. Simple arithmetic operations may involve these constants; e.g., 0.5 * **REC_DURATION** equals half the recording duration. Both waypoints have their coordinates defined using offsets: >> 0.5 represents an 0.5 latitude offset while ∼ means preserving the initial longitude. The time of passage is defined as a ratio of the alteration time window **ALT_TW**.

Batch generation. Making FDIA scenarios recording-agnostic means that they can be applied sequentially on a set of recordings without human intervention. A batch generation function was added to FDI-T as it could initially only process one recording at a time. It is now possible to supply FDI-T with a set of recordings, and the framework will iteratively apply FDIA scenarios to each of the recordings. This function, combined with the massification capabilities of the DSL, truly enables data scientists to obtain large and simultaneously rich synthetic data sets with little effort.

6. Work in Progress and Use-Cases

This section is about work in progress that directly stems from the new possibilities offered by the proposed acquisition and synthetic data generation approach. This includes the reproduction of related work (detection models), the creation of relevant datasets and the design of advanced anomaly detection models for both domains of application. Finally, several use-cases permitted by the proposed approach are also discussed.

6.1. ADS-B Datasets and Use-Cases

In Section 3, we presented the full architecture to obtain the data necessary to train and evaluate ML-based anomaly detection models. Using this architecture, the training dataset, along with the testing dataset including the original anomalies given by Habler et al. in [5], has been recreated automatically, avoiding the cumbersome task of modifying all the messages by hand For the sake of completeness, we added a trajectory alteration which modifies the latitude and the longitude of a flight, an anomaly that Habler et al. could not have obtained by hand. This eventually highlighted the shortcomings of the authors' validation methodology since their model could not detect the altered trajectory [16].

From this recreated dataset coupled with the opportunity to gather larger amounts of data, the possibility of training a variational auto-encoder [22] (VAE) emerged. The VAE uses a reconstruction probability instead of a reconstruction error as an anomaly score, which allows it to be more principled and objective [23]. The VAE is also seen as a generation model, and comparisons can be made between the trained VAE and the tool FDI-T. The first experiments have been conducted, and the results are promising.

With the expressiveness given by the DSL, its massification possibilities and the fact that it is recording-agnostic, there are now possibilities to create subtler anomalies; e.g., the trajectory modification that highlights the shortcomings of the validation methodology of Habler et al. For instance, an engine failure can be simulated by specifying waypoints to lead an aircraft toward the closest airport at which to land. In a similar fashion, hijacking scenarios could be simulated. While an instance of such a scenario could be handmade, it would take a tremendous amount of time to constitute a dataset.

Lastly, we identified two additional use-cases, focusing more on the ability of FDI-T to modify messages coming from a particular sensor: sensor disconnection and sensor drift/offset. Instead of targeting a specific aircraft, a sensor disconnection anomaly specifically targets a sensor or sensors to create a service discontinuity that could impact the OpenSky network or any other data gatherer. Sensor drift/offset consists of the creation of a drift or an offset on all messages sent by a particular sensor. This tampers with the data quality sent by a particular sensor by simulating reception or decoding errors. As a result, in the long run, it could modify the trust factor that the OpenSky network has in a seeder.

As the main goal of this paper is to determine the reproducibility and availability of the tools, it is also worth mentioning that most of the work conducted for the ADS-B raw data gathering, cleaning and processing into training data is routinely published in Github (https://github.com/Wirden/scifly). There is also a Kaggle (https://www.kaggle.com/) dataset underway to ease the sharing with the ML community. The data acquisition chain is already linked to the Traffic library and the OpenSky network for quick experimentation. As for FDI-T, it is available for experimentation and beta-testing upon request to the authors.

6.2. AIS Datasets and Associated Learning Objectives

The creation of datasets in the maritime domain is also underway, although it is still in the early phases. We are working on supervised learning techniques using FDI-T to create false data combined with a self-supervised method that uses the data history to generate new datasets. The frequency of AIS messages, the content of the messages, and the historical size of a trajectory are irregular. We decided not to sample or regularize the messages and directly used the raw messages as input. In our opinion, the missing messages also contain information that should be given to the model. Concretely, six datasets are currently in the process of creation. They are described below with their related detection objectives.

- **With self-supervised techniques:**

  AIS intentional shutdown detection: The shutdown of AIS allows a fisher to hide illegal fishing quite easily. The dataset is composed of a model that predicts if a new message should be received soon, or if the connection will be lost for diverse reasons.

  Rendez-vous prediction: A dataset containing instances of two vessels that eventually approach and stay close for a certain period of time—a so-called rendez-vous situation. The model predicts, given a path of two vessels, if they will meet in the future.

  Vessel type identification:A very broad dataset, as the model is trained to determine the type of a vessel given its path.

- **With FDI-T:**

  Trajectory modification: A dataset that contains random trajectory modifications, speed modifications and/or vessel type modifications. The objective is to detect which parts of the messages have been modified.

  Rendez-vous hiding with trajectory modification: A dataset created using FDI-T's trajectory modification feature to mimic real-life scenarios in which two vessels hide a rendez-vous by modifying their trajectory. Indeed, two vessels staying close to each other in the middle of the sea is an easily detectable behavior.

  Spoofing: A dataset in which each sample contains multiple vessels' histories; the objective of the model is to spot fake vessels among genuine ones.

7. Conclusions

This paper presents an end-to-end toolchain that is capable of creating full datasets that can be used to train and test AI-based anomaly detection systems for air and sea transportation quickly, easily and with a high degree of control . First, a surveillance data acquisition and processing module is created using surveillance historical databases and existing extraction libraries. The module helps to collect relevant data, automatically clean it from outlier messages (e.g., faulty sensors) and transform it into datasets that are readily feedable to models. Second, a synthetic anomalous data generation module is proposed that covers the whole spectrum of seen and anticipated attack scenarios. It is built upon an existing FDIA testing tool designed for surveillance systems testing, which relies on a DSL-based scenario design module to automatically generate test cases. Several extensions were developed for the DSL to facilitate the training and testing of ML models. Finally, we show the possibility to create datasets to train or test ML anomaly detection models as well as discuss different use-case scenarios for both ATC and VTS. Future experimentation in the ATC domain includes testing on Olive and Basora [24] to further validate our approach and explore more local flows of trajectories around airports, as well as to refine our own model detection for ADS-B data. Future work in the VTS domain will consist of finalizing the discussed datasets to train and evaluate the associated models.

Acknowledgments: This work is part of an ongoing research initiative toward the generation of FDIA test scenarios, partially carried out in partnership with the French company Kereval. It is partially supported by a UBFC-ISITE-BFC grant, the EIPHI Graduate school (contract *ANR-17-EURE-0002*), and the ANR ASTRID GeLeaD project grant. Most computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

References

1. Smith, A.; Cassell, R.; Breen, T.; Hulstrom, R.; Evers, C. Methods to provide system-wide ADS-B back-up, validation and security. In Proceedings of the 25th Digital Avionics Systems Conference, Portland, OR, USA, 15–19 October 2006; pp. 1–7.
2. *IALA Guideline 1082*; International Association of Marine Aids to Navigation and Lighthouse: Saint-Germain-en-Laye, France, 2016.
3. Balduzzi, M.; Pasta, A.; Wilhoit, K. *A Security Evaluation of AIS Automated Identification System*; Association for Computing Machinery: New York, NY, USA, 2014.
4. Strohmeier, M.; Martinovic, I.; Lenders, V. Securing the Air–Ground Link in Aviation. In *The Security of Critical Infrastructures: Risk, Resilience and Defense*; Keupp, M.M., Ed.; Springer International Publishing: Heidelberg, Germany, 2020.
5. Habler, E.; Shabtai, A. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages. *Comput. Secur.* **2018**, *78*, 155–173.
6. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407.
7. Almalawi, A.; Yu, X.; Tari, Z.; Fahad, A.; Khalil, I. An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Comput. Secur.* **2014**, *46*, 94–110.

8. Kiran, B.R.; Thomas, D.M.; Parakkal, R. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *J. Imaging* **2018**, *4*, 36.
9. Almgren, M.; Jonsson, E. Using active learning in intrusion detection. In Proceedings of the 17th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, USA, 30 June 2004; pp. 88–98.
10. Gornitz, N.; Kloft, M.; Rieck, K.; Brefeld, U. Toward supervised anomaly detection. *J. Artif. Intell. Res.* **2013**, *46*, 235–262.
11. Basora, L.; Olive, X.; Dubot, T. Recent Advances in Anomaly Detection Methods Applied to Aviation. *Aerospace* **2019**, *6*, 117.
12. Filipiak, D.; Stróz, M. *Anomaly Detection in the Maritime Domain: Comparison of Traditional and Big Data Approach*; Technical Report STO-MP-IST-160; NATO Science and Technology Organization: Brussels, Belgium, 2018.
13. Auslander, B.; Gupta, K.M.; Aha, D.W. *Maritime Threat Detection Using Probabilistic Graphical Models*; Naval Research Lab: Washington, DC, USA, 2012.
14. Kim, K.I.; Lee, K.M. Deep Learning-Based Caution Area Traffic Prediction with Automatic Identification System Sensor Data. *Sensors* **2018**, *18*, 3172, doi:10.3390/s18093172.
15. Nguyen.; Vadaine.; Hajduch.; Garello.; Fablet. GeoTrackNet-A Maritime Anomaly Detector Using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection. *arXiv* **2019**, arXiv:1912.00682.
16. Cretin, A.; Vernotte, A.; Chevrot, A.; Peureux, F.; Legeard, B. *Test Data Generation for False Data Injection Attack Testing in Air Traffic Surveillance*; 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto, Portugal, 24–28 October 2020.
17. Schäfer, M.; Strohmeier, M.; Lenders, V.; Martinovic, I.; Wilhelm, M. Bringing Up OpenSky: A Large-scale ADS-B Sensor Network for Research. In Proceedings of the IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014; pp. 83–94.
18. Olive, X. Traffic, a toolbox for processing and analysing air traffic data. *J. Open Source Softw.* **2019**, *4*, 1518, doi:10.21105/joss.01518.
19. Sun, J.; Ellerbroek, J.; Hoekstra, J. Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets. *J. Aerosp. Inf. Syst.* **2017**, *14*, 566–572.
20. Strohmeier, M.; Lenders, V.; Martinovic, I. On the Security of the Automatic Dependent Surveillance-Broadcast Protocol. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1066–1087.
21. Cretin, A.; Legeard, B.; Peureux, F.; Vernotte, A. Increasing the Resilience of ATC systems against False Data Injection Attacks using DSL-based Testing. In Proceeeding of the ICRAT'18, Barcelona, Spain, 26–29 June 2018; pp. 1–4.
22. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2014**, arXiv:1312.6114.
23. An, J.; Cho, S. Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* **2015**, *2*, 1–18.
24. Olive, X.; Basora, L. Detection and identification of significant events in historical aircraft trajectory data. *Transp. Res. Part C Emerg. Technol.* **2020**, *119*, 102737.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.