Anomaly Detection using GANs in OpenSky Network

Fahrettin Gökgöz
Information Technology for Command and Control
Fraunhofer FKIE
53343 Wachtberg
GERMANY

fahrettin,goekgoez.fkie.fraunhofer.de

ABSTRACT

Generative adversarial networks (GAN) have successfully been applied to the training of deep generative networks. Currently, their usage is being extended to the stabilization of sequential learning and language. In this work, we propose a GAN model that works on Air Surveillance data to detect anomalies. OpenSky Network API is well suited to the task of collecting real air surveillance data to analyze inners of the flight data using current AI techniques to support decisions makers.

Index Terms

Anomaly, Generative Adversarial Networks (GAN), Long Short Term Memory (LSTM), OpenSky Network

I. INTRODUCTION

Transportation has been well integrated to our lives since the times of working with the animals, through road vehicles and rail roads, water and air ways. It provided the means of transporting goods and people in every stages. This led governments to define it as a part of their critical infrastructure [1].Therefore, an advancement in such an important field affects our society drastically. Advancement of technology has shaped transportation, and led to the creation of new types of transportation according to human needs. Air transportation is the youngest mode of all. At the same time, it provides the safest mode of traveling by steadily improving the safety measures, according to Bureau of Transportation statistics[1] [2] [3].Even though aviation technology has improved quite and reduced the probability of accidents, the later still exist. According to plane crash info [4] 58% of accidents can be traced back to pilot errors. Other significant factors in plane crashes are: traffic control, maintenance, preparation, instability of weather conditions, failures in mechanical equipment, sabotage, terrorism, and other rare causes. These measures indicate that actions of pilots are the most critical element in the situation or in the prediction of accidents. Anomaly detection in flight surveillance data is a way of anticipating accidents. It involves the finding of interesting data points in observations. Identifying those points brings the requirement of selecting relevant features. Systematic analysis of the anomalies leads to taking the right decisions during the unexpected events, and reduces the chance of fatalities. Deep learning has become the de facto approach for automating feature engineering in many high dimensional machine learning tasks. It achieves impressive results in experimental performance. It sets new standards in many domains including speech recognition [2], [3], image classification [4], [5], and natural language processing [6], [7], [8]. In this paper, we attempt to address the ability of deep learning to automate the anomaly detection in flight surveillance data. We begin in section II by introducing the OpenSky Network. Then, in section III we provide a brief introduction to Generative Adversarial Networks,

[1] https://www.bts.gov/

[2] https://www.ntsb.gov/investigations/AccidentReports/Pages/AccidentReports.aspx

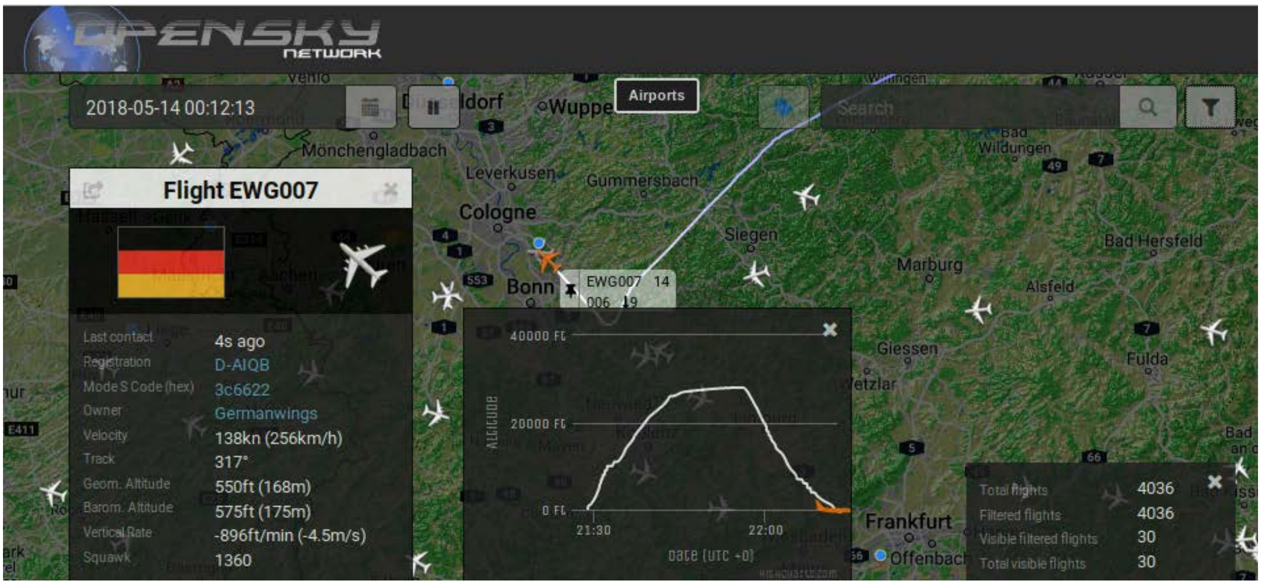
[3]http://www.bbc.com/news/business-42538053

[4] http://www.planecrashinfo.com/cause.htm

and Long Short Term Memory. Then, in section IV, we present the methodology to create and collect the data set. Section V describes the anomaly detection algorithm and section VI concludes our work.

II. BACKGROUND

Here we briefly review the related work on an air surveillance API, called OpenSky Network. OpenSky Network [9] is a live API, which provides air space information for research and noncommercial purposes, is shown in figure 1. The information provided by the API contains Automatic Dependent Surveillance - Broadcast (ADS-B) messages [10]. Airplanes in OpenSky Network are associated with a state vector. The state is a summary of the tracking information like position, velocity, and identity. Detailed information regarding the state vector can be found in table I.



Flight EWG007

Last contact	4s ago
Registration	D-AIQB
Mode S Code (hex)	3c6622
Owner	Germanwings
Velocity	138kn (256km/h)
Track	317°
Geom. Altitude	550ft (168m)
Barom. Altitude	575ft (175m)
Vertical Rate	-896ft/min (-4.5m/s)
Squawk	1360

Total flights	4036
Filtered flights	4036
Visible filtered flights	30
Total visible flights	30

Fig. 1. OpenSky Network

Systematic construction of anomaly detection benchmark from real data [11] introduces four requirements for the process of detecting anomalies: normal data points should be drawn from a real-world generating process, anomalous data points should also be from a real-world process, many benchmark datasets are needed, benchmark datasets should be characterized systematically. In our use case, OpenSky network is used as provider of real data.

III. METHODOLOGY

Deep learning provides diverse selection of algorithms to be applied on particular problems to solve. In this section, the selection of the relevant methodologies GANs and LSTMs will be introduced in respective order.

A) Generative Adversarial Networks

Deep generative models, compared to discriminative models, have received less attention due to the difficulty in approximating complex probability calculations. In the frame of GANs, complexity of this calculations is deferred to competing two neural networks; generative and discriminative [12]. The Training criterion in adversarial networks can be summarized as following. A prior on input noise variables $p_z(Z)$ is defined. Then, with a differentiable function G, the data space is mapped to those variables as $G(z; \theta_g)$.

Then, a discriminator network with a single scalar output can be described as a mapping. Given a sample x, the probability of being from original data rather then generated one;

Index	Property	Description
0	icao24	Unique ICAO 24-bit address of the transponder in hex string representation.
1	callsign	Callsign of the vehicle (8 chars).
2	origin country	Country name inferred from the ICAO 24-bit address.
3	time position	Unix timestamp (seconds) for the last position update within the past 15s.
4	last contact	Unix timestamp (seconds) for the last valid message received from the transponder.
5	longitude	WGS-84 longitude in decimal degrees.
6	latitude	WGS-84 latitude in decimal degrees.
7	geo altitude	Geometric altitude in meters.
8	on ground	Boolean value which indicates whether the position was retrieved from a surface position report.
9	velocity	Velocity over ground in m/s.
10	heading	Heading in decimal degrees clockwise from north (i.e. north=0˚).
11	vertical rate	Vertical rate in m/s. A positive value for climbing, a negative value for descending.
12	sensors	IDs of the receivers which contributed to this state vector.
13	baro altitude	Barometric altitude in meters.
14	squawk	The transponder code aka Squawk
15	spi	Whether flight status indicates special purpose indicator.
16	position source	Origin of this state’s position: 0 = ADS-B, 1 = ASTERIX, 2 = MLAT

Table I: State Vectors

$D(x; \theta_d)$. Both networks are trained simultaneously, D tries to maximize the correct label assignment, and G tries to minimize $\log(1 - D(G(z)))$. Then the value function can be described as following;

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z \sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \qquad (1)$$

Given the value function, formal training of the adversarial network can be formalized as in the algorithm 1.

B). Long Short Term Memory

Air surveillance data is sequential context-dependent data like any other surveillance data. Incorporating the context in sequences is mainly solved by the following two ways in neural networks; overlapping time windows, or recurrent connections in the model. The Time window option has certain problems with finding the optimal window. The optimal window is generally task-dependent. Additionally, the shift operation to allow different elements to consider in the same window cannot be generalized. Standard

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets [12].

```
for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{z^1, ..., z^m\}$ from noise prior $p_g(z)$.
        • Sample minibatch of $m$ examples $\{x^1, ..., x^m\}$ from data generating distribution $p_{data}(x)$.
        • Update the discriminator by ascending its stochastic gradient:
            $$\nabla_{\phi_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(x^{(i)}\right) + \log\left(1 - D(G(z^{(i)})\right) \right] \qquad (2)$$
    end for
    • Sample minibatch of $m$ noise samples $\{z^{(i)}, \cdots, z^{(m)}\}$ for noise prior $p_g(z)$
    • Update the generator by descending its stochastic gradient
        $$\nabla_{\phi_g} \frac{1}{m} \sum_{i=1}^{m} \left[ \log\left(1 - D(G(z^{(i)})\right) \right] \qquad (3)$$
end for
```

Recurrent Neural Networks (RNN) also have their own limitations and problems, such as temporal order and learning difficulties in long sequences with vanishing or exploding gradients [13], [14]. Figure 2 demonstrates the architecture introduced by LSTMs to tackle the difficulties that are occurred in RNNs. Instead of blindly folding inputs to recurrent steps, LSTM introduces state and gate concept. Each block has two states that are transferable to the next one; cell state and hidden state. These states can be modified with three gate operations; forget, input, output.

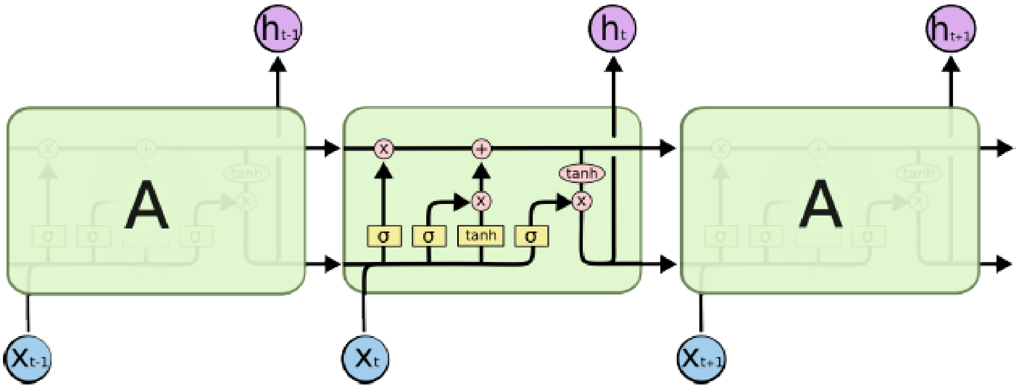


Fig. 2. LSTM Chain

The First thing that happens is the computation of the current cell state. The ”Forget” gate is responsible for this. It provides options from keeping state as it is, to completely ignoring it. They can be expressed as the following formula:

$$f_t = \sigma(W_f.[h_{t-1,x_t}] + b_f) \qquad (4)$$

The following decision is whether or not to accept the new information. The responsible gate is ”Input”. It has two parts:

$$i_t = \sigma(W_i.[h_{t-1,x_t}] + b_i)$$
$$\widetilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C) \quad (5)$$

The current state of the cell depends on the decision based on ”Input” and ”Forget” gate and can further be described as following:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \quad (6)$$

The ”Output” of the cell is decided based on:

$$o_t = \sigma(W_o.[h_{t-1,x_t}] + b_o)$$
$$h_t = o_t * \tanh(C_t) \quad (7)$$

The following section will describe our data collection and enrichment strategy.

IV. DATA COLLECTION

The OpenSky Network provides the API to connect and collect the data in sequential manner as the content described in the state vector. Each individual update is captured and stored per airplane. Further, in our collection we enriched the data as we considered important. Periodicity of the flight data is an additional criterion that is not captured in the provided API. Seasonal flights can be captured with the addition of weakly, monthly, and yearly periodicity to data.

The following section will describe the proposed solution for the anomaly detection with using the combination of LSTM and GAN.

V. ALGORITHM

An anomaly detection algorithm will be introduced in this section using the baseline from section III-A. Baseline algorithm describes the value function for learning. There are two further requirements to describe anomalies with an unsupervised manner; 1. given the test data sample, finding an encoding that describes the latent space mapping, 2. score function for the anomalies [15].

Describing the mapping from new samples to latent space is not introduced in the baseline value function. Therefore, either a post training step is needed to be applied with the smooth transition assumption on the latent space [16], or a modification on the objective function is required to include the data to latent space mapping [17]. The following objective function contains the modification required for learning the encoder $E = G^{-1}$.

$$\min_{G,E} \max_{D} V(D, E, G) = \mathbb{E}_{x \sim p_x}\left[\mathbb{E}_{z \sim p_E(.|x)}[\log D(x,z)]\right] + \mathbb{E}_{z \sim p_z(z)}\left[\mathbb{E}_{x \sim p_G(.|z)} \log\left(1 - D(x,z)\right)\right] \quad (8)$$

With the modification in the objective function, the network learns to map the data points to the latent feature space. Now we can describe an anomaly score based on the given new examples' reconstruction loss.

$$A(x) = \alpha L_G(x) + (1-\alpha)L_D(x) \text{ where } L_G(x) = ||x - G(E(x))|| \text{ and } L_D(x) = \sigma(D(x, E(x)), 1)$$

In this way, the anomaly score will be assigned to each individual test sample. High values will be representing the higher chances of the anomalies

VI. CONCLUSION

This paper discussed the possibility of the extending the base training objective function to capture the input to latent space mappings to describe the anomalies based on the introduced air surveillance data samples with out using particular labeling mechanism. This method enables the anomaly detection on high-dimensional and complex data sets. The plan for the future is to conduct an evaluation of the method with respect to field validity, and explore further possibilities with the anomaly detection.

REFERENCES

[1] J. Moteff, C. Copeland, and J. Fischer, “Report for Congress Critical Infrastructures : What Makes an,” Time, 2003.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[3] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” no. 3, 2013.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” pp. 1–14, 2014.

[5] E. Shelhamer, J. Long, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” 2016.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural Language Processing (Almost) from Scratch,” Journal of Machine Learning Research, vol. 12, pp. 2493–2537, 2011.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12, 2013.

[8] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[9] M. Schafer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, “Bringing up opensky: A large-scale ads-b sensor network for research,” in IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, pp. 83–94, April 2014.

[10] I. G. E. Gore, N. Branch, T. Roman, E. Orange, K. Wayne, and R. A. Gore, “United States Patent

(19),” no. 19, pp. 36–39, 1998.

[11] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, “Systematic construction of anomaly detection benchmarks from real data,” Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description - ODD ’13, pp. 16–21, 2013.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” Advances in Neural Information Processing Systems 27, pp. 2672–2680, 2014.

[13] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, pp. 1735–1780, nov 1997.

[14] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222–2232, 2017.

[15] T. Schlegl, P. Seebock, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10265 LNCS, pp. 146–147, 2017.

[16] J. Donahue, P. Krahenbuhl, and T. Darrell, “Adversarial feature learning,” CoRR, vol. abs/1605.09782, 2016.

[17] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-Based Anomaly Detection,” pp. 1–7, 2018.