Anomaly Detection for ICS Based on Deep Learning: A Use Case for Aeronautical Radar Data

Theobald de Riberolles, Yunkai Zou, Guthemberg Silvestre, Emmanuel Lochin, Jiefu Song

Anomaly detection for ICS based on deep learning: a use case for aeronautical radar data

Théobald de Riberolles[1] · Yunkai Zou[2] · Guthemberg Silvestre[2] · Emmanuel Lochin[2] · Jiefu Song[1]

Abstract

Industrial control systems (ICS) are no longer restricted to industrial production. They are also at the heart of safety critical systems and carry out key information that require strong need in terms of availability and integrity. Furthermore, they are gradually connected with the Internet. In the context of Air Traffic Management, safety critical data are generally time series which contain periodic events. Anomalies can hardly be detected as we only have a little knowledge of the traffic characteristic and the kind of anomalies we might encounter. Consequently, detecting them is challenging as it requires high detection accuracy currently unfeasible with traditional methods based on anomaly signatures or predictions. To cope with this issue, we introduce an anomaly detection method for ICS based on Long Short Term Memory (LSTM) that outperforms the accuracy of traditional ones. We experiment and develop our method with one major dataset containing French civil radar aviation data. We then evaluate our scheme with different datasets containing ICS monitoring data (publicly available predictable time series data) and show that our autoencoder can detect anomalies from predictable times series and present a higher detection rate on average than traditional detection methods.

Keywords Anomaly detection · Industrial control systems · Deep learning · Cybersecurity

1 Introduction

Industrial control systems provide the building blocks to perform tasks such as the provision of utilities and the execution of complex manufacturing processes. Industrial control system (ICS) is, by definition, a combination of control components (electrical, mechanical, hydraulic, pneumatic, etc.) that act together to achieve an industrial objective (manufacturing, transportation of matter and energy, etc.). Actually, ICS encompasses a broad denomination which includes several types of systems such as SCADA (Supervision Control And Data Acquisition), DCS (Distributed Control System), IACS (Industrial Automation and Control Systems) or PCS (Process Control System).

Traditionally, ICS were considered to be well protected by a so-called air-gapped separation. Because of their criticality and importance, ICS are usually set on private networks, that are (supposed to be) isolated from the Internet. There is therefore no need to develop dedicated Intrusion Detection Systems for them. However, ICS networks are increasingly connected to open networks and can be targeted by external incoming attacks, e.g., using a USB key or smartphone. As an example, we recall two famous incidents that occurred severe damage in the past: the attack on a power grid in Ukraine in 2015 [23] and the Stuxnet virus, which targeted Siemens industrial control systems [6]. This evolution towards a better interconnection makes them vulnerable to cyber-attacks.

According to ICS-CERT (the Industrial Control Systems Cyber Emergency Response Team, an infrastructure of the Cybersecurity and Infrastructure Security Agency (CISA)),

Théobald de Riberolles
theobald.deriberolles@activus-group.fr

Yunkai Zou
zouyunkaicauc@qq.com

Guthemberg Silvestre
guthemberg.da-silva-silvestre@enac.fr

Emmanuel Lochin
emmanuel.lochin@enac.fr

Jiefu Song
jiefu.song@activus-group.fr

[1] Activus Group, 1 chemin du Pigeonnier de la Cépière, 31100 Toulouse, France

[2] ENAC, 7 avenue Edouard Belin, 31400 Toulouse, France

targeted attacks on ICS have increased in recent years. In 2015,[1] 295 incidents were reported to the ICS-CERT compared to 73 in 2013,[2] as a matter of fact, this trend must be thwart. NIST (the National Institute of Standards and Technology) summarizes the main security concerns of modern ICS.[3] Some of them are:

- Design-insecure communication protocols [9];
- Insecure network separation and access controls [19];
- Lack of specific ICS firewalls and anomaly detection systems [24].

This last vulnerability motivates us to develop an anomaly detection method specific for ICS. We propose to monitor network activities from data logs or network traffic and generate alarms when anomalies are detected. The development and use of anomaly detection has been widely discussed for traditional systems; however, little work has been done to ensure an efficient Anomaly Detection System for ICS. The main challenge for this type of anomaly detection lies in the fact that most of ICS are based on specific communication protocols and which are not considered by traditional anomaly detection; then, there is a need to take into account the behavior of the data transmitted by the specific protocol and to explore the specific aspects of the data flows to develop the mechanism of detection. There is also a lack of actual ICS datasets to assess anomaly detection. Furthermore, for this type of anomaly detection, the temporal character of the data which are time series, must be inspected to assess whether or not we are front to an anomaly in comparison with a normal behavior, to prevent false positives and obtain an exhaustive detection system. We contribute by developing an anomaly detection for ICS. Based on the data exchanges at the heart of these systems, we detect anomalies with a higher accuracy than the state of the art. Our method is based on machine learning techniques using an autoencoder based on Long Short Term Memory (LSTM) network cells [12] which are coupled with signatures of normal network behavior at the time series analysis level allowing to detect anomalies accurately and fastly.

The article is organized as follows. A background linked to the detection of anomalies, the ICS, the critical systems security measures in Section 2. Section 3 describes the datasets and formulates the paper problem. The deep learning model for the detection of outliers is explained in Section 4. The use of our method to detect anomaly in an aviation dataset is illustrated in Section 5. An example of detection of spoofing attack in this dataset is reported in Section 6 with a comparison with other detection methods. An opening by using our method with other ICS is proposed in Section 7. Finally, Section 8 concludes the article.

2 Background

This section defines what is an anomaly and provides an anomaly detection background. We then highlight the challenge faced with ICS anomaly detection and show existing solutions. The last part of this section focus on a method that we use for our detection.

When analyzing real-world datasets, a common need is to determine which instances stand out as being different from all the others. Such instances are called anomalies deviants or outliers in the data mining and statistics literature [1]. Anomalies can be caused by data errors but are sometimes indicative of a new underlying process, previously unknown. They are usually the cause of an attack that can take place on the system itself. One of the best-known attacks illustrating this is a Spoofing attack through a Man In The Middle: an attacker is placed between the transmitter and the receiver with the objective to alter data, thus impacting the system thereafter. Taking the example of radar data, an attacker can continually change the aircraft position information given by the radar message, that is, the "bubbling" spoofing attack [4], the radar monitoring system will have difficulty detecting these subtle differences, resulting in improper guidance from air traffic controllers or delays in responding to the collision avoidance system, thus a potential danger to human lives.

The purpose of anomaly detection is to assess the distance between the collected data and a reference standard behavior. This detection is a defense applied for decades as in abnormal program behavior, botnet, and IoT intrusion detection.

Regarding air traffic data itself, there is few work about anomaly detection. In his thesis, Nanduri [18] deals with this type of anomaly detection, as detecting atypical flights and anomalies based on statistical signatures or detecting anomalies in the data in the vector space.

In their document "Using ASTERIX in accident investigation" [10] Farrel and Schuurman explain that radar data are often used for investigation of air accidents, and discuss ASTERIX data,[4] for safety use.

[1] https://us-cert.cisa.gov/sites/default/files/Monitors/ICS-CERT_Monitor_Nov-Dec2015_S508C.pdf

[2] https://us-cert.cisa.gov/sites/default/files/Annual_Reports/Year_In_Review_FY2013_Final.pdf

[3] https://us-cert.cisa.gov/sites/default/files/recommended_practices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf

[4] ASTERIX stands for STructured Eurocontrol suRveillance Information eXchange and is the EUROCONTROL (European Organisation for the Safety of Air Navigation) standard for the exchange of surveillance-related data.

Nevertheless, Casanovas et al. [3] present a proof of concept about the vulnerability of the ASTERIX protocol. They were able to set a Man In the Middle attack dedicated to this traffic with the objective to delete, modify or add aircraft inside the traffic. This study highlights the fact that there is a need to have an additional security level against an attack from inside the network.

Furthermore, the survey document [25] presents the difficulties in developing anomaly detection specific to system as the ATM one, the ICS and in particular SCADA because they can have limited resources, components that are not secure or old, availability requirements. They define existing approaches such as knowledge-based, behavioral and hybrid. Steven Chung presents in his article [5] one of the first IDS for SCADA that builds models used to capture the normal behavior of the system based on statistical measurements for a specific communication protocol. Another approach, close to a conventional approach, is based on a supervised method with known anomalies [21] and thus builds up normal behaviors over time by correlating the various anomalies. These methods are therefore based both on the system and their knowledge. Note we often have little knowledge of the behavior of ICS.

To improve these approaches, techniques based on machine learning methods have been developed in various fields. Studies such as Javaid [13] have shown that deep learning methods overtake standard ones. They constructed self-evolving models for further classification. Based on the available data and the anomalies identified or not, these algorithms create profiles of the ICS network normal behavior. Then, the anomaly detection is made by calculating the distance of the traffic with the normal profiles. These methods use classification techniques to a class as presented in the article [17], but also statistical Bayesian networks [2] which improve the accuracy of anomaly detection and make easier to detect anomalies or new attacks. This reduces the false positive rate by combining several abnormal detection mechanisms such as n-grams and invariant induction. A light and fast IDS for a SCADA system was also based on a Bloom filter-based intrusion detection for smart grid SCADA using the regular communication models of the system. There are many choices for the predictor, such as the ARIMA auto-regressive integrated moving average predictor, the predictor based on SVM (Support Vector Machine), and the predictor based on the long-term memory network (LSTM).

Time series prediction models have proven effective in detecting anomalies using the prediction error or a function of the prediction error to measure the severity of the anomaly [15].

2.1 Why choosing LSTM network

The authors in [12] present recurring methods used for many sequences learning tasks such as handwriting recognition, speech recognition, and sentiment analysis, and more recently to perform data prediction on times series.

In particular, LSTM-AD (Long Short Term Memory networks for Anomaly Detection) is used as prediction models:

- In time series;
- In anomaly detection in EEG (electroencephalogram) time signals via deep-long-term memory networks where the probability of prediction error is used to measure anomalies.

Given the long-term learning capacity of LSTMs, their frequent use and their abilities to learn from the unknown makes them good candidates for solving anomaly detection problems for ICS. Furthermore, a recent detection mechanism with a time series level detection model based on an LSTM network combined with a Bloom filter was proposed by [11] to develop an IDS specific to ICS which have better ability in detecting anomalies than other techniques for ICS networks.

LSTM Encoder-Decoder models are presented as a natural extension of LSTM models for time series with better performance. They have recently been proposed for sequence-sequence learning tasks such as machine translation [22]. Based on the success of LSTM for anomaly detection dedicated to the ICS and the need for the most precise detection possible in an ICS given the criticality of the data transported, we decided to based our detection method on the LSTM Encoder-Decoder models.

3 Dataset and problem description

This section presents the method chosen for measurement, describes our problem and the metric of abnormal score. We use aviation radar data encapsulated in a specific protocol named EUROCONTROL ASTERIX (previously introduced in Section 2), also encapsulated in Ethernet data. Our dataset is a set of real radar data collected in the French ATM/ATC System. These data consist in twenty-three Secondary Surveillance Radar (SSR) and nine Primary Surveillance Radar (PSR) collected before processed by the calculators of the control center. We capture these data, in PCAP format, during the year 2019. With the ASTERIX protocol, radars identifiers are represented by destination addresses, and source addresses are the last SIR (Server

of Radar Information identifier), so we used Ethernet destination address to distinguish the different radars of our collection.

3.1 Data used

We use the ASTERIX Python module developed by Damir Salantic for Croatia Control Ltd. to parse EUROCONTROL ASTERIX protocol data, and an internal Scapy[5] module to manipulate and visualize data radar packets.[6]

From the ASTERIX packet of CAT48,[7] the main category used in this study, we decode the time, position, velocity, and flight level information from the radar network packet, where polar coordinates represent the position information while the speed information includes the magnitude and heading of aircraft. A detailed description of the relevant information is shown Table 1.

3.2 Problem description

An n-dimensional time series $S = \{S_1, S_2, ..., S_C\}$ is defined, which represents a radar sequence window, where C is the length of the time series. $S_i = \left\{s_i^1, s_i^2, \ldots s_i^n\right\} (1 \leq i \leq C)$ is an n-dimensional vector, each dimension corresponding to one feature. Specifically, S represents a window composed of continuous C radar message information, and each vector S_i contains characteristic information obtained from the corresponding radar message, namely, position, altitude and speed. During the training phase, the correct radar time series is used as training data input into the anomaly detection model, which forces the reconstruction of the sequence. After the training is completed, when the correct radar time series is input, the reconstruction error will be within a certain range. However, when the sequence containing the abnormality is used as the input, the reconstruction error will be amplified, thereby achieving the effect of abnormal detection.

3.3 Abnormal score definition

The cosine similarity [20] is often used when the frequency of similar occurrence is high in a time series. It is used to represent the reconstruction error between the output vector $\hat{S}$ and the input vector S, which is defined as follows:

$$\cos\left(S_i, \hat{S}_i\right) = \frac{\sum_{j=1}^{n}\left(s_i^j \times \hat{s}_i^j\right)}{\sqrt{\sum_{j=1}^{n} s_i^{j2}} \times \sqrt{\sum_{j=1}^{n} \hat{s}_i^{j2}}}, \text{ with } n, \text{ the feature dimension.}$$

On this basis, the reconstruction anomaly score of the time series with recurrent behavior [7] can be defined as follows:

$$\text{abnormal score} = \sum_{i=k}^{k+C} 1 - \cos\left(S_i, \hat{S}_i\right), \text{ with } C, \text{ the length of the input window.}$$

4 Anomaly detection method

Following state of the art, we use the "autoencoder model" to reconstruct the radar time series and then use the reconstruction error to detect the anomaly. The anomaly detection process is sliced into the following steps:

1. To decode radar packet from raw radar data;
2. To extract radar series feature;
3. To reconstruct radar series by a deep neural network based on the autoencoder model;
4. To calculate the reconstruction error;
5. To highlight abnormal data.

The autoencoder belongs to the unsupervised learning methods class. This model is of interest when the type of abnormal data cannot be obtained in advance. Thus, the correct data can be used as a training sample so that the model can learn from the relevant characteristics of the correct data. After the model training is completed, if the input data's characteristics do not conform to the model's laws, the output will also be different, thus achieving the effect of anomaly detection.

4.1 Autoencoder in a nutshell

An autoencoder aims at reproducing an input n-dimensional vector S by an output n-dimensional vector $\hat{S}$. It comprises two components: an **encoder** and a **decoder**. The encoder maps an input vector S into an m-dimensional intermediate vector F; then, the decoder maps F to an output vector $\hat{S}$ that is expected to approximate the input vector S.

The encoder and decoder can be formally defined as the following two functions:

Encoder : $\varphi : R^n \rightarrow R^m$

Decoder : $\phi : R^m \rightarrow R^n$

Then, the objective function of the autoencoder can be defined as:

$$argmin_{\varphi,\phi} \|S - \phi(\varphi(S))\|_2^2$$

Our goal is to determine the appropriate function to minimize the error between the input vector S and the output vector $\hat{S} = \phi(\varphi(S))$. In the encoding phase, the mapping

[5] Scapy is a packet manipulation tool for computer networks

[6] In this paper the term packet and frame gave the same meaning.

[7] A detection category for SSR mode S.

Table 1 Radar message attributes

Attribute	Description	Unit	Range
Track plot number (TPN)	An unique reference to a plot record within a particular plot file	–	[0, 65535]
TS	Epoch standard time in UTC	s	Varies
THETA	Measured position of an aircraft in local polar coordinates	deg	[0, 360]
RHO	Measured position of an aircraft in local polar coordinates	nautical mile	[0, 250]
Calculated Ground Speed (CGS)	Calculated track velocity in polar coordinates	Knt	[0, 500]
Calculated Heading (CHDG)	Calculated track velocity in polar coordinates	deg	[0, 360]
Flight Level (FL)	Flight level information	hft	[0, 400]

relationship is defined as follows:

$$F = \sigma_1(W_1 S + B_1)$$

where $W_1 \in R^{m \times n}$ is a weight matrix, $B_1 \in R^m$ is an offset vector, and σ_1 is the activation function, i.e., sigmod function. In the decoding phase, the mapping relationship is defined as follows:

$$\hat{S} = \sigma_2(W_2 F + B_2)$$

where $W_2 \in R^{n \times m}\bullet$ is a weight matrix, $B_2 \in R^n$ is an offset vector, and σ_2 is the activation function.

4.2 LSTM

In this paper, the encoder and decoder of the autoencoder model consists in LSTM units. The LSTM architecture consists of memory cells used to learn long-term modes, each cell containing its current state and three non-linear gates: the forget gate, the input gate, and the output gate. The forget gate is responsible for determining how much memory information to forget. This innovative gate is linked to our data processing. This allows us to not only focus on past data and have properties without memory. It is determined by a non-linear function that outputs a number between 0 and 1, where 0 means forgetting all information in memory, and 1 means retaining all information in memory. The input gate is responsible for deciding how to update the old cell state, i.e., the new information is selectively recorded in the cell state. The output gate is responsible for deciding how much memorable information to pass to the next cell.

4.3 Data preprocessing

At first, it was necessary to preprocess the data so that it could be used to input the model. For that, we first sorted the data by different aircraft according to the TPN and the temporal continuity; then, we used the min-max scaling transform to normalize the data.

We then converted them into a three-dimensional shape with the sample number, the length of the window, and the feature number. Here, the data entered the model is in the form of a sliding window. Specifically, the data index of the first group entering the model is [0, 9], the second is [1, 10], the third is [2, 11], and so on. Table 2 shows some raw sample data before preprocessing.

We select about 800,000 pieces of data (time range is 4 hours) as training samples and then select 100 separate flights as test samples. Actually, the training data we use is not labeled, but we know that the data located at the flight

Table 2 Raw data sample

INDEX	TPN	TS	THETA	RHO	CGS	CHDG	FL
0	3	1555729447	209.0533447	22.203125	453.2	218.704834	330.5
1	3	1555729451	209.2730713	22.69140625	457.6	218.7322998	330.5
2	3	1555729455	209.465332	23.171875	452.54	218.5015869	330.5
3	3	1555729459	209.6685791	23.66015625	454.08	218.5235596	330.5
4	3	1555729463	209.866333	24.140625	452.98	218.7432861	330.5
5	3	1555729467	210.0201416	24.6328125	456.06	218.1445313	330.5
6	3	1555729471	210.1904297	25.125	457.16	218.0291748	330.5
7	3	1555729475	210.3717041	25.61328125	455.4	218.4960938	330.5
8	3	1555729479	210.50354	26.10546875	454.74	218.0731201	330.5
9	3	1555729483	210.6793213	26.59765625	456.06	218.5180664	330.5
10	3	1555729487	210.8166504	27.08984375	456.06	218.4356689	330.5

transition boundary will lead to higher anomaly scores. So this kind of data can be regarded as abnormal data. We can distinguish different aircraft by TPN and TS, and the location of the abnormal data is clearly known, which is equivalent to "labeled". Therefore, this method belongs to the semi-supervised learning method.

5 Analyze suspicious data in the dataset

Firstly, we focus on the possible anomalies in the dataset itself, which can help us set an accurate anomaly threshold. Specifically, we will test a series of individual flights without injection. It turns out four cases may cause a higher abnormal score. To sum up them:

- *normal change of angle*: when an aircraft crosses the radar area vertically, its angle ranges from 0 degree to 360 degrees or vice versa. These degrees are numerically different, but both degrees mean the same angle. This numerically leads to a corner jump, which is a normal behavior but results in an increase of the abnormal score. One solution to resolve it could be to use the sine of the angle instead of the angle;
- *change point*: there are some change points in the speed parameters of the aircraft, which also causes an increase of the abnormal score;
- *data continuity interruption*: a dataset might contain lost time sequence in a dataset. These losses can be either due to data transport or normal loss (the so-called radar cone of silence) or anomaly. In any case, these losses must be differentiated as they all increase of the abnormal score;
- *time series of violent fluctuations*: we may also have the case where the speed of the aircraft keeps fluctuating and some values may belong to outliers. Our method can identify them.

We give examples of two different aircraft that contain exactly these four types of anomalies. We have plotted the anomaly scores of these two aircrafts in Fig. 1. We can easily see four peaks that correspond to our four cases of anomalies previously identified.

To confirm whether these cases are true positives, we lay on the judgment of air traffic controllers that check these data. This is the only way to prevent detection the normal cycle change of the aircraft angle as abnormal.

Threshold of suspicious data Different thresholds can be specified according to different preferences. If we need more detection of various types of anomaly data, including a large amount of suspicious data, set a lower threshold, but at the same time it will also lead to higher false alarms.[8] If we choose to ignore some suspicious data, we can set a higher threshold, but at the same time it narrows the scope of anomaly detection. We calculate the reconstruction abnormal score of training data (after depth feature extraction), and define an abnormality threshold that 94% of the abnormal score is less than the value. This value, defined after a study of the ROC curve, allows us in our case to have the best ratio between high detection of true positives, but avoiding false positives as much as possible. This choice was made with the requirements provided by the industrial partner of this work, meaning it can be changed following the requirements needed. Data are considered suspicious if the abnormal score is greater than the threshold. Basically, we set this value by estimating the amount of such data based on the total number of flights and the length of the sliding window.

6 Spoofing attack detection

In this section, we focus on spoofing attack, a type of Man In The Middle attack. The replication and the retransmission of radar transmit signals, are designed to provide false information to a radar to corrupt received data.

This kind of attack has the potential to cause the radar to report false information and greatly increases the risk of a collision. We concentrate the study with the modification of data of the two aircrafts which we could observe previously.

We took the initial data and injected them with abnormal data. In particular, we have increased all the data (except the time parameter) with a rate of 10%. We did this on the serial numbers 100 to 110 and 140 to 150 in order to observe two attacks.

Figure 2 represents the abnormal score of these two aircrafts after injection of the attack. Modified sequences are marked in red.

At first, we observe that the model developed is not sensitive to the data that we have modified. We only observe slight fluctuations in the modified sequences that merge into the already existing fluctuations. Thus, the change in the abnormal score does not allow the abnormalities to be clearly identified. We, therefore, need to magnify them to detect them.

6.1 Feature enrichment

The vectors in a time series are often inter-dependent rather than being independent. Following the study of

[8] Note the situation that causes false alarm is mainly the change of the angle parameter near the period value. One improvement is to convert the polar coordinates into cartesian coordinates. In addition, the suspicious data mentioned above will also cause false alarms to some extent

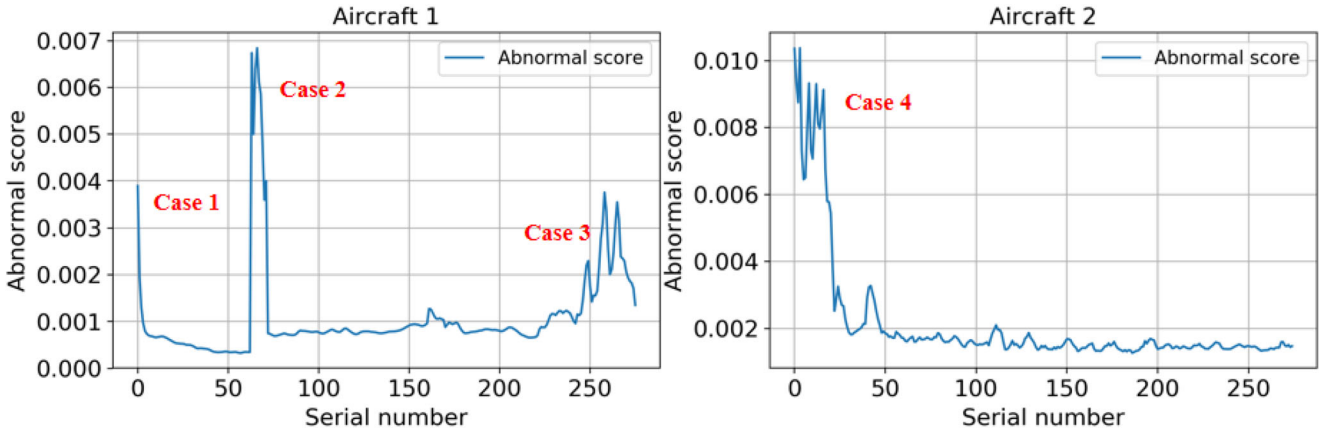


Fig. 1 The reconstructed anomaly scores of the two aircrafts illustrating the four cases of anomaly

Kieu et al. [14], we employ sliding windows to account for dependencies and compute statistical features in each window to obtain the deeper feature information. In the further processed time series, the feature space is much larger than in the raw time series, which helps the autoencoder to identify the most representative features for a small space. There is a three-step procedure to further process a raw time series.

For each window, we compute d derived features for each feature in the raw time series. Specifically, we consider $d = 2$ derived features—norm (NOR) and difference of norm (DON). We compute $e = 8$ statistical features for each derived feature : mean (MEA), minimum (MIN), maximum (MAX), 25%-quartiles (25Q), 50%-quartile (50Q), 75%-quartile (75Q), standard deviation (STD), and peak to peak (P2P). Then, $ed = 16$.

The window T_i, a vector of radar sequence denoted S_i, is defined as $T_i = \langle S_i, S_{i+1}, \ldots, S_{i+c-1}\rangle \in \mathbb{R}^{n\times c}$. For each window T_i, we define a subwindow $T_{i,\ell}$ $\left(\ell = \{1, 2 \ldots, C'\}, C' = \frac{2(C-b)}{[b]} + 1\right)$ with a step size b ($b > 1$, an even number) where the two consecutive windows have a overlap step size of $[b/2] = [b]/2$, i.e.,

$$T_{i,\ell} = \left\langle S_{i+[b/2](\ell-1)}, S_{i+[b/2](\ell-1)+1}, S_{i+[b/2](\ell-1)+2}, \cdots, S_{i+[b/2](\ell-1)+(b-1)}\right\rangle.$$

For each subwindow $T_{i,\ell}$, we define Norm $\mathrm{NOR}(T_{i,\ell}) \in \mathbb{R}^n$,the Euclidian length of vector $T_{i,\ell}$ and difference of norm $\mathrm{DON}(T_{i,\ell}) \in \mathbb{R}^n$, the difference between the norm of window $T_{i,\ell}$ and the norm of its previous window $T_{i-1,\ell}$ where n is the number of features (such as TPN, TS,...). The j-the element ($j \in \{1, \ldots, n\}$) of these two values $\mathrm{NOR}^j(T_{i,\ell})$ and $\mathrm{DON}^j(T_{i,\ell})$ are defined as follows:

$$\mathrm{NOR}^j(T_{i,\ell}) = \sqrt{\left(s^j_{i+[b/2](\ell-1)}\right)^2 + \left(s^j_{i+[b/2](\ell-1)+1}\right)^2 + \cdots + \left(s^j_{i+[b/2](\ell-1)+(b-1)}\right)^2}$$
$$\mathrm{DON}^j(T_{i,\ell}) = \mathrm{NOR}^j(T_{i,\ell}) - \mathrm{NOR}^j(T_{i,\ell-1})$$

Similarly and as shown in Fig. 3, a new time series T'_i is defined with another step window with a step size f ($f > 1$, an even number) and whose overlap length between two consecutive windows is $[f/2] = [f]/2$. Then, we define another window with the time series $T''_i =$

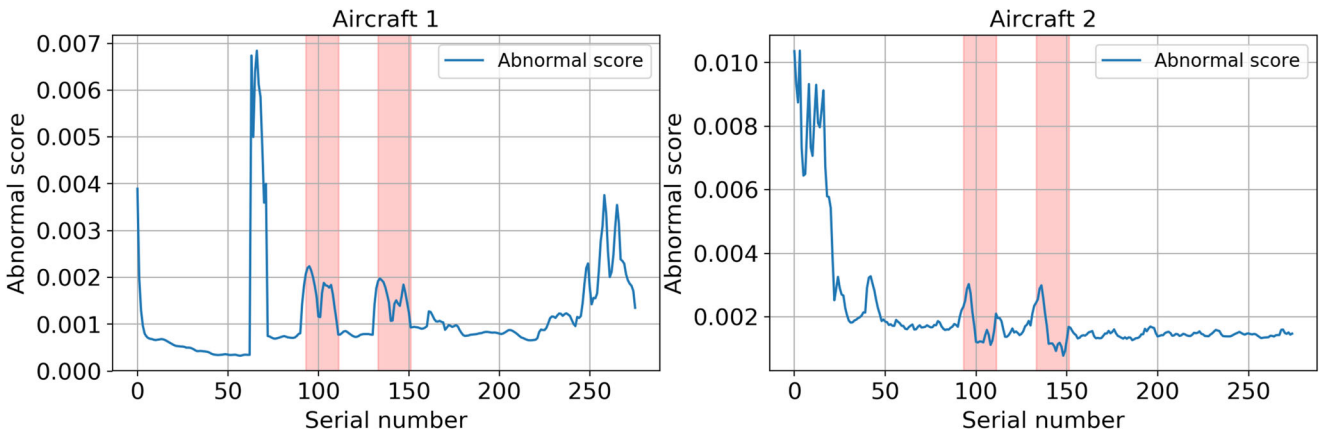


Fig. 2 Abnormal score of two aircrafts after injecting a spoofing attack

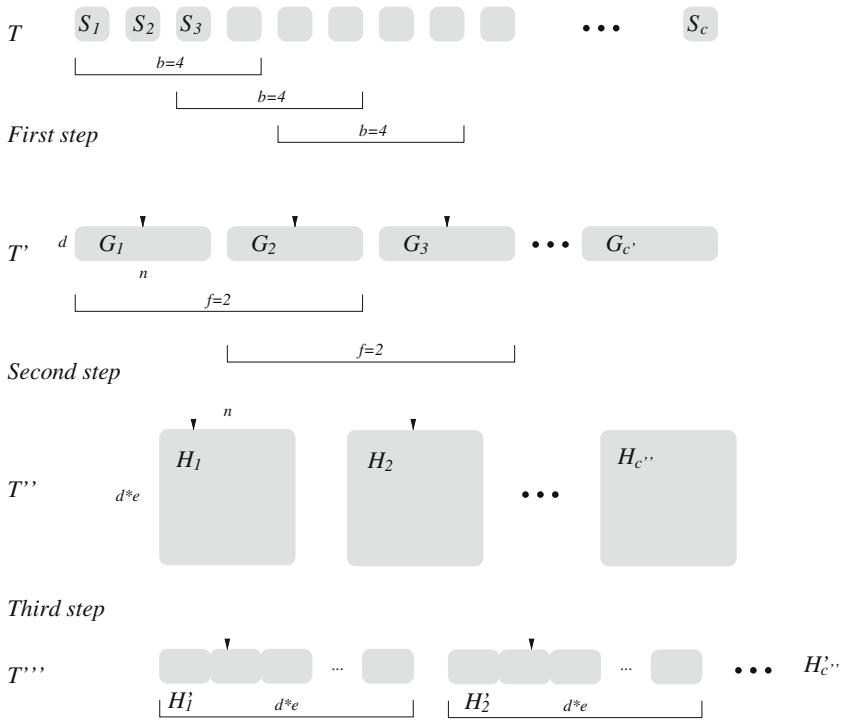


Fig. 3 Three steps of feature expansion

$\left\langle H_{i,1}, H_{i,2}, \ldots, H_{i,C''}\right\rangle$ such that (r, j) element of $H_{i,p} \in \mathbb{R}^{ed\times n}$ (denoted by $H_{i,p}^{r,j}$) with u the window size-index into f, is defined as follows:

$$
\begin{aligned}
H_{i,p}^{1,j} &= \frac{1}{f}\sum_{u=1}^{f} \mathrm{NOR}^j\left(T_{i,[f/2](p-1)+u}\right)\\
H_{i,p}^{2,j} &= \min_{1\le u\le f}\left\{\mathrm{NOR}^j\left(T_{i,[f/2](p-1)+u}\right)\right\}\\
H_{i,p}^{3,j} &= \max_{1\le u\le f}\left\{\mathrm{NOR}^j\left(T_{i,[f/2](p-1)+u}\right)\right\}\\
&\ldots\\
H_{i,p}^{9,j} &= \frac{1}{f}\sum_{u=1}^{f} \mathrm{DON}^j\left(T_{i,[f/2](p-1)+u}\right)\\
&\ldots\\
H_{i,p}^{16,j} &= \max_{1\le u\le f}\left\{\mathrm{DON}^j\left(T_{i,[f/2](p-1)+u}\right)\right\}\\
&\quad - \min_{1\le u\le f}\left\{\mathrm{DON}^j\left(T_{i,[f/2](p-1)+u}\right)\right\}
\end{aligned}
$$

Finally, we define $H'_{i,p} \in \mathbb{R}^{edn}$ by expanding and vectorizing $H_{i,p}$, that is,

$$H'_{i,p} = \left(H_{i,p}^{1,1}, H_{i,p}^{1,2}, \cdots, H_{i,p}^{1,n}, H_{i,p}^{2,1}, H_{i,p}^{2,2}, \ldots, H_{i,p}^{16,1}, \cdots, H_{i,p}^{16,n}\right)^T$$

The final window with the time series T_i''' is defined as $T_i''' = \left\langle H'_{i,1}, H'_{i,2}, \ldots, H'_{i,C''}\right\rangle$.

The different steps of this expansion are illustrated in Fig. 3.

For the two aircrafts above, we inject the same form of anomaly data in the same location. It can be seen in Fig. 4 that after deep extraction of features, the anomaly is significantly magnified. It is worth noting that after the deep feature extraction of the time series, the total length of the sequence becomes about half of the original, and the corresponding abnormal serial number also becomes half (e.g., from [100, 110] to [48, 53]). At the same time, we also added a detection threshold. The setting method is similar to the previous one, and the value is 0.194989. Then, we simulate more kinds of spoofing attacks.

6.2 Evaluation approach

To evaluate the performance of the learned model, we injected various types of anomalies. Their detailed description is listed below. As a matter of fact, when we alter a given value, we do it consistently by considering possible dependency with another one. The objective is not to mimic an attack that would lead to an obvious detection.

- *Theta deviation (THETA)*: anomalies are generated by modifying THETA parameter. We modified the original values of the THETA parameter by 45 degrees.
- *RHO deviation (RHO)*: anomalies are generated by modifying RHO parameter. We modified the original values of the RHO parameter by 25 nautical miles.
- *All parameters deviation (ALL)*: anomalies are generated by modifying all parameters (except the time

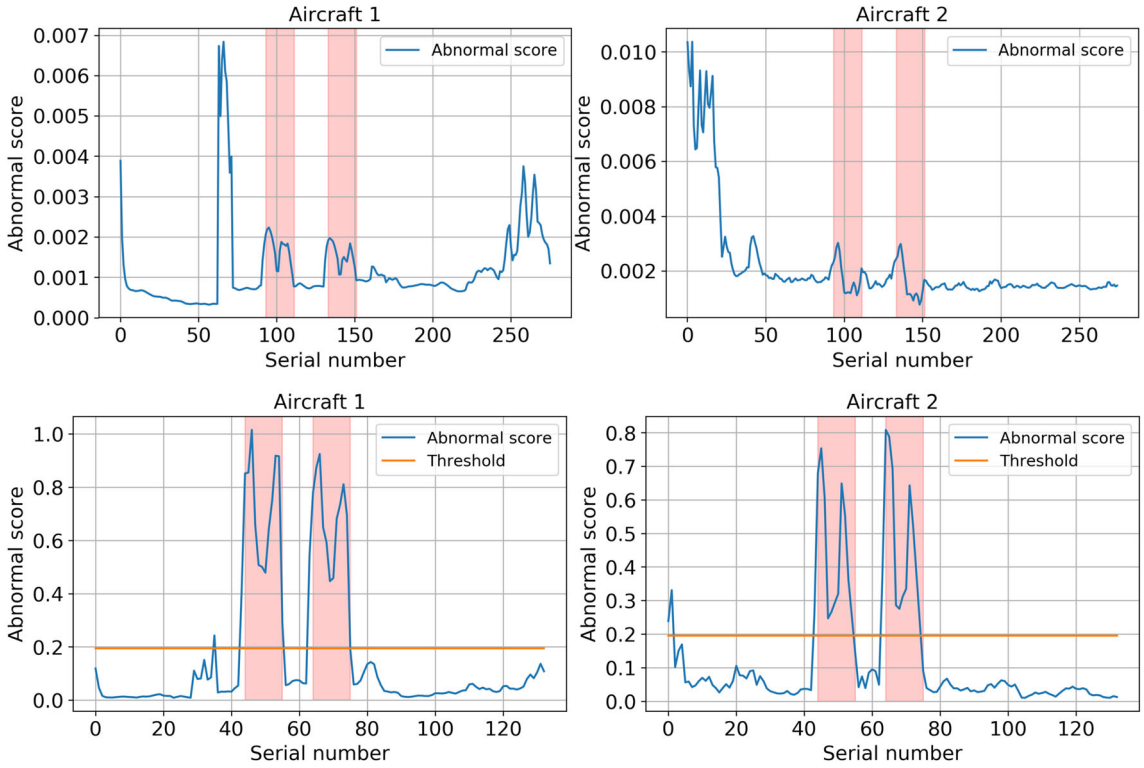


Fig. 4 Abnormal score of two aircrafts after injecting a spoofing attack (the second one after depth feature extraction)

parameter). We modified the original values of all parameters by 10 percent.

- *Random noise (RND)*: anomalies are generated by introducing random noise. We multiplied the original values of the message attributes of the Radar messages with a randomly generated floating number between 0 and 2.
- *Different route (ROUTE)*: anomalies are generated by replacing a segment of the Radar messages of the tested flight with a segment of messages from a different (legitimate) route. In the experiment, we replaced the test flight with 15 other flight messages.
- *Gradual drift (DRIFT_FL)*: anomalies are generated as a gradual drift in the flight level feature. This is done by modifying the flight level of a segment of messages by continuously increasing/lowering the flight level by an increasing multiplier of 400 feet (i.e., for the first message in the anomalous segment, the flight level is increased/decreased by 400 feet, the second message is increased/decreased by 800 feet, and so on). We generated two types of gradual drifts by raising the altitude value and lowering the altitude value.
- *Gradual drift (DRIFT_CGS)*: the same way as modifying flight levels. Specifically, for the first message in the anomalous segment the speed is increased/decreased by 10 knt, the second message is increased/decreased by 20 knt, and so on). We also generated two types of gradual drifts by raising the altitude value and lowering the altitude value.

6.3 Metrics

We evaluate the accuracy of anomaly detection methods by precision, recall and F1-Score, which are defined as follows:

- *Precision*: precision is the ratio of correctly predicted positive observations to the total predicted positive observations;
- *Recall (Sensitivity)*: recall is the ratio of correctly predicted positive observations to all observations in the actual class;
- *F1 score*: F1 Score is the weighted average of Precision and Recall.

TP, FP, and FN are referred to as true positive, false positive, and false negative, respectively. We might fail to detect potential anomalies if we only pay attention to the precision. However, some false positives might be received when we only focus on the recall. F1-Score builds up the accuracy and recall, is therefore used as the main evaluation metric in our experiments.

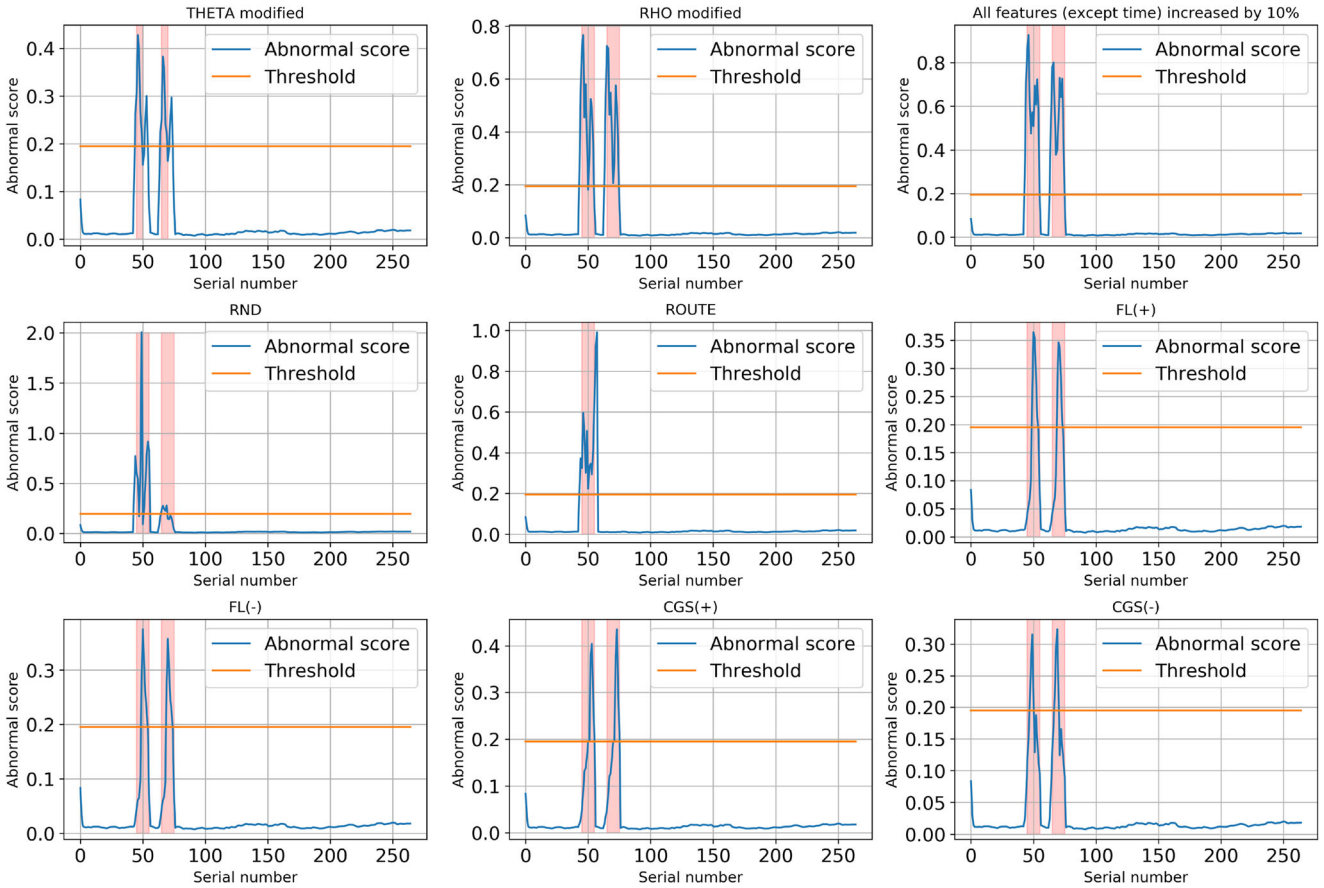


Fig. 5 Abnormal scores of modified characteristics from an attacked aircraft

Table 3 Overall experimental results focuses on the timestamp

Method	Evaluation	THETA	RHO	ALL	RND	ROUTE	FL(+)	FL(-)	CGS(+)	CGS(-)	MEAN
OC-SVM	Precision	0.6657	0.6627	0.6810	0.6900	0.6634	0.6503	0.6200	0.6655	0.6637	0.6625
	Recall	0.4962	0.4876	0.5496	0.5890	0.4875	0.4654	0.3879	0.4968	0.4879	0.4942
	F1 score	0.5319	0.5258	0.5715	0.6017	0.5276	0.5024	0.4259	0.5303	0.5211	0.5265
IF	Precision	0.1572	0.1274	0.7567	0.7584	0.7582	0.1347	0.1137	0.4419	0.1449	0.3770
	Recall	0.0680	0.0598	0.4752	0.4927	0.4712	0.0534	0.0537	0.1345	0.0801	0.2098
	F1 score	0.0737	0.0625	0.5573	0.5700	0.5567	0.0567	0.0575	0.1629	0.0821	0.2422
LOF	Precision	0.4125	0.5064	0.7337	0.7454	0.7884	0.4544	0.4579	0.4558	0.4540	0.5565
	Recall	0.2859	0.3663	0.5103	0.5138	0.5446	0.3288	0.3317	0.3276	0.3279	0.3930
	F1 score	0.8944	0.9991	0.9014	0.8585	0.8141	0.9118	0.9983	0.8169	0.8293	0.8915
LSTM	Precision	0.9074	0.9983	0.8884	0.8970	0.9220	0.9339	0.9972	0.7741	0.7782	0.8996
	Recall	0.2872	0.3333	0.3044	0.4635	0.2458	0.1501	0.1667	0.1680	0.1756	0.2550
	F1 score	0.4336	0.4998	0.4417	0.5728	0.3774	0.2581	0.2856	0.2585	0.2648	0.3769
Autoencoder	Precision	0.8789	0.8893	0.8885	0.8852	0.8887	0.8082	0.8299	0.8151	0.8248	0.8565
	Recall	0.7514	0.8833	0.8690	0.8179	0.8606	0.4123	0.4326	0.4313	0.4558	0.6571
	F1 score	0.7914	0.8762	0.8669	0.8307	0.8619	0.5243	0.5447	0.5416	0.5663	0.7116

Table 4 Experimental results focuses on the number of attacks

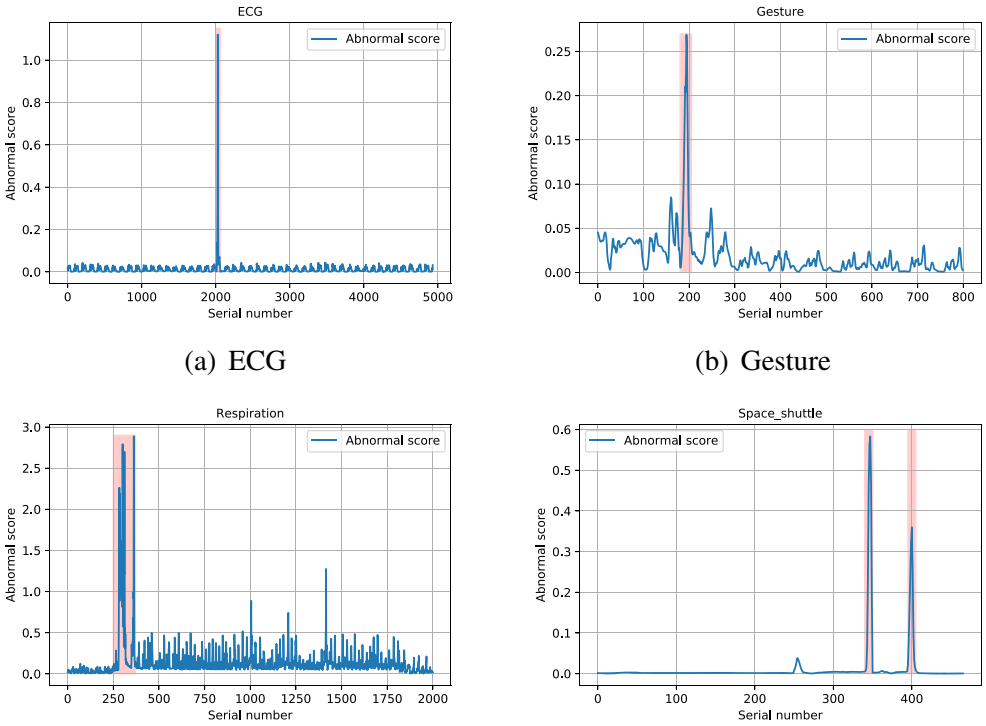
Method	Evaluation	THETA	RHO	ALL	RND	ROUTE	FL(+)	FL(-)	CGS(+)	CGS(-)	MEAN
LSTM	Precision	0.9074	0.9983	0.8884	0.8915	0.9220	0.9339	0.9972	0.7741	0.7782	0.8990
	Recall	0.8884	1.0000	0.9587	0.8760	0.7833	0.9008	1.0000	0.9669	0.9959	0.9300
	F1 score	0.8944	0.9991	0.9014	0.8585	0.8141	0.9118	0.9983	0.8169	0.8293	0.8915
Autoencoder	Precision	0.8789	0.8893	0.8885	0.8870	0.8887	0.8116	0.8337	0.8199	0.8290	0.8585
	Recall	0.9835	0.9959	0.9959	0.9669	0.9958	0.9669	0.9835	0.9793	0.9876	0.9839
	F1 score	0.9101	0.9288	0.9286	0.9108	0.9289	0.8574	0.8792	0.8730	0.8788	0.8995

6.4 Experimental result

Figure 5 visually represents an example of an aircraft injecting anomalies during the cruise phase. After tests on different stages of more than 100 individual aircraft, the results are presented in Table 3.

We consider three non-deep learning-based baselines—One-Class Support Vector Machines (OC-SVM, a linear transformation-based method), Local Outlier Factor (LOF, a distance-based method)and Isolation Forest (IF, an integration-based method). These methods are state-of-the-art unsupervised anomaly detection algorithms that can be used for time series data or continuous data. They are implemented through the Scikit-learn library. However, these methods only use the feature information of the current point, and do not consider the target motion information of the current point in the adjacent time, so the detection effect is not as good as autoencoder. In our previous research [8], we have used the LSTM neural network (a non-linear transformation-based method) to predict the time series and compare the predicted value with the received value to detect abnormal data, so we also performed the same experiment. The overall results show that the autoencoder model based on deep learning can achieve the best detection results in most of cases.

Further, we get higher metrics values if we calculate the recall by the number of attacks (our detection target is to detect two attacks located at two specific sequence segments). We can do this as the data are in the form of a window when entering the LSTM unit. If the abnormal score of a point exceeds the threshold, the entire window sequence where this point is located can lead to an



(a) ECG (b) Gesture

(c) Respiration (d) Space

Fig. 6 Anomaly detection applied to other datasets

abnormality. Table 4 shows the latest statistical results for LSTM and autoencoder. It shows that both methods have satisfactory detection results.

We also considered the effect of the length of the sliding window, for the results (average of various attack types) of using different sliding windows during feature enrichment. In general, the length of the sliding window has little effect on the results, on average we have a precision of 0.84682, a recall of 0.91568 and a F1 score of 0.8575. As the length of the window increases, the detection effect gradually deteriorates. The reason is that large windows mask the temporal changes during a short time, which adversely affects accuracy.

7 Opening

We conducted a preliminary test on several other ICS datasets (publicly available predictable time series data [16]) such as ECG, Gesture, Respiration, and Space Shuttle, to verify that the proposed method can be applied in a different context and to test the performance of our method. Among them, the ECG and Gesture data used in this paper are two-dimensional data. After the features enrichment process, the dimensions become 32 dimensions, and the total length becomes half of the original ($b = 4$, $f = 2$). Respiration and Space Shuttle are one-dimensional data, which becomes 16-dimensional after feature enrichment, and the total length also becomes half of the original. When testing the ECG dataset, we found that anomalies can be detected without enriching the features. However, to verify that feature enrichment does not impair on the detection effect, we have tested these datasets still under the condition of feature enrichment. We illustrate the results obtained with these datasets in Fig. 6. The red area is a potential abnormal area and it can be seen that the method used in this paper has a good detection effect. These results are promising and show that the method proposed can be applied to a different context. An extended publication will be dedicated to present these results.

8 Conclusion and future work

Based on the autoencoder model embedded in LSTM units, this paper performs anomaly detection on the parameters commonly used in radar data by comparing the reconstruction error of radar series. We first detect the anomalous data (suspicious data) that may exist in the radar dataset itself. Then, we enrich the radar time series to further detect various types of spoofing attacks and the results are satisfied.

Points for potential improvements of this work are:

- The modified granularity needs to be further refined;
- The time range of the training and the test dataset need to be further expanded, from one radar to different radars;
- Suspicious data in the dataset has not been removed, and the dataset needs further purification.

In future work, we expect to collect suspicious data from the radar dataset and work with air traffic controllers to determine the properties of such data; further purify the dataset to achieve more accurate results, and deal with more complex attacks.

References

1. Aggarwal CC (2017) An introduction to outlier analysis. In: Outlier analysis. Springer, pp 1–34
2. Bigham J, Gamez D, Lu N (2003) Safeguarding SCADA systems with anomaly detection. In: International workshop on mathematical methods, models, and architectures for computer network security. Springer, pp 171–182
3. Casanovas EE, Buchaillot TE, Baigorria F (2015) Vulnerability of radar protocol and proposed mitigation. In: ITU Kaleidoscope: trust in the information society (K-2015), 2015. IEEE, pp 1–6
4. Chan-Tin E, Heorhiadi V, Hopper N, Kim Y (2011) The frog-boiling attack: limitations of secure network coordinate systems. ACM Trans Inform Syst Secur (TISSEC) 14(3):1–23. Publisher: ACM New York, NY, USA
5. Cheung S, Dutertre B, Fong M, Lindqvist U, Skinner K, Valdes A (2007) Using model-based intrusion detection for SCADA networks. In: Proceedings of the SCADA security scientific symposium, vol 46. Citeseer, pp 1–12
6. Collins S, McCombie S (2012) Stuxnet: the emergence of a new cyber weapon and its implications. J Polic Intell Counter Terror 7(1):80–91
7. Compagnon P, Lefebvre G, Duffner S, Garcia C (2019) Routine modeling with time series metric learning. In: International conference on artificial neural networks. Springer, pp 579–592
8. de Riberolles T, Song J, Zou Y, Silvestre G, Larrieu N (2020) Characterizing radar network traffic: a first step towards spoofing attack detection. In: 2020 IEEE Aerospace conference. IEEE, pp 1–8
9. Dzung D, Naedele M, Von Hoff TP, Crevatin M (2005) Security for industrial communication systems. Proc IEEE 93(6):1152–1177
10. Farrell P, Schuurman M (2012) Using ASTERIX in accident investigation
11. Feng C, Li T, Chana D (2017) Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In: 2017 47th Annual IEEE/IFIP international conference on dependable systems and networks (DSN). IEEE, pp 261–272
12. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neur Comput 9(8):1735–1780. Publisher: MIT Press
13. Javaid A, Niyaz Q, Sun W, Alam M (2016) A deep learning approach for network intrusion detection system. In: Proceedings

of the 9th EAI International conference on bio-inspired information and communications technologies (formerly BIONETICS), pp 21–26
14. Kieu T, Yang B, Jensen CS (2018) Outlier detection for multidimensional time series using deep neural networks. In: 2018 19th IEEE International conference on mobile data management (MDM). IEEE, pp 125–134
15. Ma J, Perkins S (2003) Online novelty detection on temporal sequences. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 613–618
16. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series
17. Nader P, Honeine P, Beauseroy P (2014) lp-norms in one-class classification for intrusion detection in SCADA systems. IEEE Trans Indus Inform 10(4):2308–2317
18. Nanduri SKA (2016) Anomaly detection in aircraft performance data
19. Obregon L (2015) Secure architecture for industrial control systems. SANS Institute InfoSec Reading Room
20. Ristanti PY, Wibawa AP, Pujianto U (2019) Cosine similarity for title and abstract of economic journal classification. In: 2019 5th International conference on science in information technology (ICSITech). IEEE, pp 123–127
21. Skopik F, Friedberg I, Fiedler R (2014) Dealing with advanced persistent threats in smart grid ICT networks. In: ISGT. IEEE, pp 1–5
22. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112
23. Whitehead DE, Owens K, Gammel D, Smith J (2017) Ukraine cyber-induced power outage: analysis and practical mitigation strategies. In: 2017 70th Annual conference for protective relay engineers (CPRE), pp 1–8
24. Yang D, Usynin A, Hines JW (2006) Anomaly-based intrusion detection for SCADA systems. In: 5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05), pp 12–16
25. Zhu B, Sastry S (2010) SCADA-specific intrusion detection/prevention systems: a survey and taxonomy. In: Proceedings of the 1st workshop on secure control systems (SCS), vol 11, p 7