Telemetry data integrity monitoring system

V V Berkholts¹, A M Vulfin¹, A I Frid¹

¹ Ufa State Aviation Technical University, Ufa, Karla Marxa str., 12, 450000, Russia
E-mail torina4@yandex.ru

Abstract. The article presents a way to improve the system for monitoring the integrity of telemetric information received from a mobile object. The method is based on comparing telemetric information received from a mobile object and information generated by a model. The comparison is carried out on the basis of proximity metrics of technological time series using deep learning neural networks

Keywords: Data integrity; Information Security; time series, neural networks

1. Introduction

Safety requirements for mobile objects oblige to register and store data about their state using on-board control devices and recorders. Similar requirements apply to both ground transport [1] and air [2]. Ensuring the reliability of data transmission systems about an object - telemetric information (TMI), is one of the important tasks of ensuring the effective functioning of a mobile object. The possibility of transmitting TMI about the actual state of individual modules during operation and the entire complex of equipment to the manufacturer makes it possible to increase the efficiency of operating a mobile object in a normal state, in the event of failures or attacks by intruders, during incident investigation.

The need to transfer recorded data on the state of a mobile object (MO) to a manufacturer for the purpose of subsequent analysis is one of the possible ways to improve promising on-board systems [3, 4, 5] and allows the MO to gain a competitive advantage. The problem of building secure systems for collecting, storing and processing data on the state of a mobile object in such systems is urgent.

Monitoring the integrity of data received from a mobile object to a manufacturer was previously considered by the authors in [6]. The method presented there was based on a comparison of technological time series (TTS) [7], characterizing the behavior of the parameters of a mobile object and the model of the same object installed at the manufacturer. Comparison of two TBRs was carried out using a set of fuzzy logic rules proposed by the authors.

The disadvantage of this approach is the low probability of detecting an intruder, due to the fact that the proposed rules for making decisions for a large number of MO modes of operation are not a sufficiently flexible mechanism for assessing the proximity of TBP, and therefore for establishing the fact of intruder.

An additional factor that reduces the likelihood of detecting intruders is that the correlation and determination coefficients used in [6] illustrate the proportion of variance, and the consistency of TBR is actually determined by only two parameters: the proportion of variance and the average percentage of deviation, which leads to a decrease in the accuracy of the estimation of consistency. TVR.
The purpose of this article is to increase the likelihood of making the right decision about the fact of an attack on data integrity.

To achieve this goal, the following tasks were solved:
1. a classifier of operating modes of a mobile object has been developed;
2. mathematical models of possible attacks of an intruder have been determined;
3. A classifier of attacks by an intruder has been developed;
4. a block for making a decision on the presence or absence of an attacker in the current time window has been developed.

2. Classifier of operating modes of a mobile object

In the previous work of the authors [6], the main factor that reduces the likelihood of detecting an attacker was the partitioning of data into multiple operating modes. This partitioning is redundant and complicates the set of rules for deciding whether an attack has occurred. Thus, it is necessary to determine the minimum required number of operating modes of a mobile object.

As a mobile object, we will consider a gas turbine engine (GTE) with an automatic control system (ACS) installed on an aircraft (AC).

To solve the problem of clustering the GTE parameters by operating modes, the data recorded on board the aircraft were taken. A fragment of the flight data recording 60 minutes long was taken. These data are recorded with reference to standard atmospheric conditions. The sampling period is 0.25 seconds.

For the task of selecting the number of modes, 2 parameters were identified that are most sensitive to changes in the ACS GTE operating modes: specific fuel consumption G and high-pressure rotor speed n2, and it is more informative to analyze the increment of these parameters.

To solve the problem of classifying the operating modes of the ACS GTE, the clustering process is carried out in a sliding time window. For a qualitative classification, the width of the time window must be at least five samples in order to recognize the classes of states of the ACS GTE.

Clustering was carried out on a sliding time window of 15 counts. Table 1 provides an example of such time windows.

| № window | ΔG1  | … | ΔG14 | ЄN1  | … | ЄN14 |
|----------|------|---|------|------|---|------|
| 1        | IG_2-G_1 | | IG_15-G_14 | ЄN_2- ЄN_1 | | ЄN_15- ЄN_14 |
| 2        | IG_3-G_2 | | IG_16-G_15 | ЄN_3- ЄN_2 | | ЄN_16- ЄN_15 |
| 3        | IG_4-G_3 | | IG_17-G_16 | ЄN_4- ЄN_3 | | ЄN_17- ЄN_16 |
| …        |       |   |       |       |   |       |
| 140000   | IG_14002-G_14001 | | IG_14015-G_14014 | ЄN_14002- ЄN_14001 | | ЄN_14015- ЄN_14014 |

The selection of the number of operating modes (clusters) was carried out using the "Elbow method" [8]. The essence of the method is that it is necessary to select such a number of clusters so that adding another one does not improve the quality of clustering.

Figure 1 illustrates the application of the Elbow method to the selected parameters. The criterion by which the quality of clustering was assessed with a different number of clusters, inertia (the sum of squared errors in the context of clustering). In Figure 1, inertia is located on the ordinate, and the abscissa is the number of clusters k. The “fracture” characteristic of this method occurs on 2 clusters. Thus, 2 classes of states of the ACS GTE are distinguished. In a similar way, the number of state classes for other mobile objects can be selected.
To make sure the data breaks down really well into the two state classes of a mobile object, visualization of this breakdown is needed. Visualization was carried out by two methods: principal component analysis (PCA) [9] and t-distributed Stochastic Neighbor Embedding (t-SNE) [10].

PCA visualization is shown in Figure 2a, t-SNE visualization is shown in Figure 2b.

In Figure 2a clearly shows a large cluster of points in the vicinity of the coordinate (0,0). Since not the values of the parameters themselves at the moment of time \( t \) were clustered, but only their increments, it is possible to determine the physical meaning of such a distribution of points. The increments of parameters close to zero indicate the static operating mode of the ACS GTE, the increments located far from the center of coordinates indicate the dynamic operating mode of the object.

After determining the operating modes of a mobile object, it is necessary to select a method for classifying TMI, which will be used in the future to determine the fact of an attacker. The k-nearest neighbors (KNN) algorithm, one of the most popular machine learning algorithms, was chosen as the classification algorithm [11]. After testing the classification algorithm on a test sample, the classification accuracy was 0.991.

3. Mathematical models of attacks
In order to recognize attacks based on the attacker model described in [12], mathematical models of four types of attacks on TMI were developed.
1. Attack. Overlay multiplicative noise

1A. Attack in the entire time window. The monitored parameter after manipulations by the attacker takes the following value:

\[ x_i^z = a_i \cdot x_i^r, \]  

\( i, i \in [1, n] \) - iteration number in a time window of length \( n \), \( x_i^r \) - the parameter value in the \( i \)-th iteration, \( x_i^z \) - the value of the parameter, after manipulation by the attacker, \( a_i \), - coefficient, a random number, and \( a_i \in (-\infty; -1,1) \cup (1,1; +\infty) \)

1B. Attack in part of the time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = \begin{cases}  
  k_i \cdot x_i^r, & i \leq \frac{n}{2}, k_i \in (-1.1; 1.1) \\
  a_i \cdot x_i^r, & i > \frac{n}{2}, a_i \in (-\infty; -1,1) \cup (1,1; +\infty) 
\end{cases} \]  

\( k_i \) - natural noise ratio arising from data transmission, which is a random number in a small range

1C. Attack in a small part of the time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = \begin{cases}  
  k_i \cdot x_i^r, & i \leq \frac{2n}{3}, k_i \in (-1.1; 1.1) \\
  a_i \cdot x_i^r, & i > \frac{2n}{3}, a_i \in (-\infty; -1,1) \cup (1,1; +\infty) 
\end{cases} \]  

2. Substitution of data similar to real ones.

2A. Attack in the entire time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = a_i \cdot x_i^r \pm b_i, \]  

\( a_i \), - coefficient, a random number, and \( a_i \in (0.5; 0.9) \cup (1.1; 1.5) \)

\( b_i \), - coefficient, a random number, and \( |b_i| \geq 0.2 \cdot x_i^r \).

We also prohibit the mutually compensating values \( a_i \) and \( b_i \). For example, the following event cannot be perceived as an attacked signal:

\( a_i = 0.8, x_i^r = 5, b_i = 1 \),

so \( x_i^z = 5 \), which is equal to the original \( x_i^r \).

2B. Attack in part of the time window. The parameter after the manipulation of the attacker takes the following value:

\[ x_i^z = \begin{cases}  
  k_i \cdot x_i^r, & i \leq \frac{n}{2}, k_i \in (-1.1; 1.1) \\
  a_i \cdot x_i^r \pm b_i, & i > \frac{n}{2}, a_i \in (0.5; 0.9) \cup (1.1; 1.5), |b_i| \geq 0.2 \cdot x_i^r 
\end{cases} \]  

2C. Attack in a small part of the time window. The parameter after manipulation by the attacker takes the following value:
\[ x_i^z = \begin{cases} 
    k_i \cdot x_i^r, & i \leq \frac{2n}{3}, k_i \in (-1.1; 1.1) \\
    a_i \cdot x_i^r \pm b_i, & i > \frac{2n}{3}, a_i \in (0.5; 0.9) \cup (1.1; 1.5), |b_i| \geq 0.2 \cdot x_i^r, 
\end{cases} \]  

(6)

3. Increase or decrease the signal while maintaining the behavior of the signal (Additive Noise Overlay)

3A. Attack in the entire time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = x_i^r \pm b_i, \]  

(7)

\( b_i \) - coefficient, a random number, and \( |b_i| \geq 0.2 \cdot x_i^r \).

3B. Attack in the entire time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = \begin{cases} 
    k_i \cdot x_i^r, & i \leq \frac{n}{2}, k_i \in (-1.1; 1.1) \\
    x_i^r \pm b_i, & i > \frac{n}{2}, |b_i| \geq 0.2 \cdot x_i^r.
\end{cases} \]  

(8)

3C. Attack in a small part of the time window. The parameter after manipulation by the attacker takes the following value:

\[ x_i^z = \begin{cases} 
    k_i \cdot x_i^r, & i \leq \frac{2n}{3}, k_i \in (-1.1; 1.1) \\
    a_i \cdot x_i^r \pm b_i, & i > \frac{2n}{3}, a_i \in (0.5; 0.9) \cup (1.1; 1.5), |b_i| \geq 0.2 \cdot x_i^r.
\end{cases} \]  

(9)

4. Forgery of controllers and external influences. When falsifying control and external influences, the model will generate signals from the automatic control system of the gas turbine engine that do not correspond to the actual values of the parameters. Thus, the data generated by the model can be considered attacked.

4A. Impact across the entire time window. To simulate this kind of impact, you can substitute a real signal into any function

\[ y_i^z = f(x_i^r). \]  

(10)

\( y_i^z \) - parameter value generated by the model for forged control actions, \( f(x_i^r) \) - any function.

4B. Attack in part of the time window. The parameter after manipulation by the attacker takes the following value:

\[ y_i^z = \begin{cases} 
    k_i \cdot x_i^r, & i \leq \frac{n}{2}, k_i \in (-1.1; 1.1) \\
    f(x_i^r), & i > \frac{n}{2}.
\end{cases} \]  

(11)
4C. Attack in a small part of the time window. The parameter after manipulation by the attacker takes the following value:

\[ y_i^z = \begin{cases} k_i \cdot x_r^i, & i \leq \frac{2n}{3}, k_i \in (-1.1; 1.1) \\ f(x_r^i), & i > \frac{2n}{3}, \end{cases} \]  

(12)

4. Construction of classifiers for the presence and type of attack on TMI

To build a classifier of the type of attack on TMI, it is necessary to form a training and test sample. To create such samples, 14000 time windows were formed, 15 samples each. For the experiments, the specific fuel consumption G was taken as the parameter modified by the attacker.

An example of some time windows for building an attack type classifier is presented in Table 2.

| Time windows | Model data | Attacked/not attacked data | State class | Attack type |
|--------------|------------|----------------------------|-------------|-------------|
| G0           | G14        | G20                        | S/D         | 0           | 1           | 2           | 3           | 4           |
| 2.8          | 4.0        | 2.8                        | 3.8         | D           | 0           | 1           | 0           | 0           |
| 1.3          | 1.3        | 2.2                        | 44.         | S           | 0           | 0           | 0           | 0.4         | 0.6         |
| 1.3          | 1,         | 1.4                        | 1.4         | S           | 1           | 0           | 0           | 0           |

If data with different types of modifications fell into the same time window, the shares of the attacks were entered in the "Attack type" columns, as shown in line 2.

Attack type classifier is a multi-input and multi-output classifier, so conventional classification algorithms (for example, the usual kNN and Random Forest) are not suitable for this task. An algorithm is needed to build multiple regression. To solve this problem, Multioutput random forest regressor is used.

The quality of classification of attack types by this method was 0.65.

The low quality of classification by multiple regression methods forces us to switch to more complex classification algorithms. It is necessary to apply algorithms built on neural networks to this problem.

Neural networks are a universal tool for approximating functions, which allows them to be used in solving classification problems [13]. A typical task when using neural networks is choosing the neural network architecture (number of layers, activation functions, etc.) for each task. Table 3 shows the iterations done to find the attack type classifier.

Activation functions listed in the table:

1. Tanh-hyperbolic tangent

\[ \sigma(x) = \tanh(x) \]  

(13)

2. ReLu is a rectified linear unit. Less computationally intensive than hyperbolic tangent or sigmoid because it performs simpler mathematical operations. Therefore, it makes sense to use ReLu when creating deep neural networks.

\[ \sigma(x) = \max(0, x) \]  

(14)

3. Sigmoid-sigmoid function

\[ \sigma(x) = \frac{1}{1+e^{-x}} \]  

(15)
As can be seen from Table 3, a further increase in the number of neurons and layers does not give a gain in the quality of attack classification. Thus, neural network number 4 gives the best quality of attack type classification with an accuracy of 0.90.

The type of attack is not the most important metric in assessing the integrity of data, it is much more important to understand whether the data is modified or not. To assess the presence of an attack, the following set of time series proximity metrics was used: determination coefficient (R2), mean absolute percent error (MAPE), Euclidean distance, and dynamic time transformation metric (DTW) [14]. Thus, the input data packet for the classifiers of the presence of an attack is as follows (table 4):

| №  | Description of the neural network | Accuracy |
|----|----------------------------------|----------|
|    | Layer | Activation function | Number of neurons | of |
| 1  | 1     | Tanh          | 100          | 0.83 |
| 2  | 1     | RELU          | 25           |     |
|    | 3     | Sigmoid       | 5            |     |
| 2  | 1     | RELU          | 100          | 0.89 |
| 2  | 2     | RELU          | 25           |     |
|    | 3     | Sigmoid       | 5            |     |
| 3  | 1     | RELU          | 100          | 0.90 |
| 2  | 2     | RELU          | 50           |     |
|    | 3     | RELU          | 50           |     |
|    | 4     | Sigmoid       | 10           |     |
| 4  | 1     | RELU          | 100          | 0.90 |
| 2  | 2     | RELU          | 50           |     |
|    | 3     | RELU          | 50           |     |
|    | 4     | Sigmoid       | 10           |     |
| 5  | 1     | RELU          | 100          | 0.79 |
| 2  | 2     | RELU          | 50           |     |
|    | 3     | RELU          | 50           |     |
|    | 4     | RELU          | 10           |     |
| 6  | 1     | RELU          | 150          | 0.90 |
| 2  | 2     | RELU          | 100          |     |
|    | 3     | RELU          | 100          |     |
|    | 4     | Sigmoid       | 20           |     |

The classification quality of the classifier constructed using the Random Forest method was 0.89. It is possible to improve the quality of the classification using algorithms built on neural networks..

| R2          | MAPE      | Euclid distance | DTW | State class | Attack |
|-------------|-----------|-----------------|-----|-------------|--------|
| 0.96        | 6.83      | 3.8             | 7   | dynamic     | 0      |
| 0.38        | 10.24     | 5               | 14  | static      | 1      |
| 0.38        | 11.4      | 10              | 17  | static      | 1      |

The classification quality of the classifier constructed using the Random Forest method was 0.89. It is possible to improve the quality of the classification using algorithms built on neural networks.

Table 5. Finding the best classifier for the presence of an attack
| Layer | Activation function | Layer |
|-------|---------------------|-------|
| 1     | ReLu                | 50    | 0.85 |
| 1     | ReLu                | 50    |
| 3     | Out                 | 1     |
| 2     | ReLu                | 100   | 0.90 |
| 2     | ReLu                | 50    |
| 3     | Out                 | 1     |
| 3     | ReLu                | 100   | 0.90 |
| 2     | ReLu                | 100   |
| 3     | Out                 | 1     |
| 4     | ReLu                | 50    | 0.98 |
| 2     | ReLu                | 50    |
| 3     | ReLu                | 50    |
| 4     | Out                 | 1     |
| 5     | ReLu                | 100   | 0.98 |
| 2     | ReLu                | 50    |
| 3     | ReLu                | 50    |
| 4     | Sigmoid             | 10    |

As can be seen from the table, further increase in the number of neurons and layers does not give a gain in the quality of establishing the fact of an attack. Thus, neural network number 4 gives the best classification quality when establishing the fact of an attack.

5. Conclusion
The article presents a way to improve the method of monitoring the integrity of data transmitted from the board of a mobile object, based on a comparison of two time series - obtained from the model of the mobile object and from the object itself. The improvement of the method consists in the justification of the number of state classes for each type of mobile objects. On the example of the operation of the ACS GTE, two classes of object states were distinguished: static and dynamic. This separation allows the classification process to explain a greater difference between two TTSs in a dynamic state class than between two TTSs in a static state class.

To further classify the type of possible attacks, mathematical models have been developed of the attacker's possible impact on the TVR. Four main types of actions have been formed, and on the basis of these data, a classifier of the type of attack of an attacker, built on a deep neural network, has been developed. The accuracy of determining the type of attack was 0.98.

To establish the fact of the presence or absence of an attack, a binary classifier was constructed, the input vector of which is the metrics of the proximity of TTS (determination coefficient, MAPE, Euclid distance and DTW), as well as the class of the state of the mobile object (static or dynamic). The attack presence classifier is also based on a deep learning neural network. The accuracy of determining the presence or absence of an attack was 0.98.

Acknowledgment
The work is partially supported by the RFBR grant # 20-08-00668.

References
[1]. Decision support in the task of ensuring information security of aviation telemetry systems. Order of the Ministry of Transport of Russia dated February 13, 2013 No. 36 "On approval of requirements for tachographs installed on vehicles, categories and types of vehicles equipped with tachographs, rules of use, maintenance and control over the operation of tachographs installed on vehicles."
[2]. Order of the Ministry of Transport of the Russian Federation of July 31, 2009 N 128 "On Approval of the Federal Aviation Rules" Preparation and Performance of Flights in Civil Aviation of the Russian Federation "(with amendments and additions)

[3]. Gorbatov V S, Zhukov I Yu, Murashov O N 2014 Cryptographic protocol for authentication and generation of a common key for vehicle control devices Saf.of Info. Tech 5 24 27-

[4]. Chuyanov G A , Kosyanchuk V V, Selrseyuk N I, Kravchenko S V 2014 Directions for improving on-board equipment to improve the safety of aircraft flights Izv. SFedU. Tech Sci.. 6 155

[5]. Bronnikov A M 2017 The efficiency of the technical operation of the onboard system of the aircraft, unattended during the service period Sci. Bull. of MSTU GA. 6 12

[6]. Guzairov M B Frid A I, Vulfin A M, Berholts V V 2019 The concept of integrity of telemetric information about the state of an aircraft power plant monitoring Proc.of 2019 ICOECS (Ufa:Ufa aviation technical state university) 10.1109/ICOECS46375.2019.8950020

[7]. Vulfin A M, Frid A I 2011 A neural network model for analyzing technological time series within the framework of the Data Mining methodology Inf. and Con.Sys. 5 31-38

[8]. Robert L Thorndike 1953 Who Belongs in the Family Psychometrika. 18 4: 267-76.

[9]. Jolliffe I T 2002 Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY XXIX, 487 p. 28

[10]. Gashi I, Stankovic V, Leita C, Thonnard O 2009 An Experimental Study of Diversity with Off-the-shelf AntiVirus Engines Proc of the IEEE Inter. Symp. on Net. Com and App. (Cambridge, MA) 4-11.10.1109/NCA.2009.14

[11]. Hastie T, Tibshirani R, Friedman J 2001 The Elements of Statistical Learning Springer, ISBN 0-387-95284-5

[12]. Guzairov M B, Frid A I, Vulfin A M, Berholts V V 2020 Decision support in the task of ensuring information security of aviation telemetry systems Wor. of Intern. Sym. Rel and Qual (accepted for publication)

[13]. Kim, Y 2014 Convolutional neural networks for sentence classification IEMNLP 1746 -51

[14]. Ghazi Al-Naymat, Sanjay Chawla, Javid Taheri 2009 Sparse DTW: A novel approach to speed up Dynamic Time Warping The 2009 Aus. D Min. 101 117-127 arXiv:1201.2969