Wireless Networks (2023) 29:3051–3061
https://doi.org/10.1007/s11276-023-03353-1

Flight data outlier detection by constrained LSTM-autoencoder

Long Gao[1] · Congan Xu[1,2] · Fengqin Wang[3] · Junfeng Wu[1] · Hang Su[1]

Accepted: 4 April 2023 / Published online: 10 May 2023

Abstract

Detecting outliers of flight data is an important research field for flight safety. Deep learning methods have achieved remarkable performance in the outlier detection tasks for time series data. The majority of previous deep-learning-based outlier detection methods for flight data focus on either learning descriptive features by matching the distribution of inliers with autoencoder-based models, or learning semantic features by mapping inliers into a hyper-sphere with kernel functions, while the information of the given class samples is insufficiently utilized. To address this issue, in this paper, we propose a novel multi-task-based model that can jointly learn descriptive and semantic features. The proposed model is based on an LSTM autoencoder to reconstruct the inputs, and we design a constraining layer to pull the learned semantic features together. By jointly training two branches of the model, the proposed method can learn to fit the distribution of inputs as well as map inliers into a tight hyper-sphere, thus making outliers and inliers more distinguishable. Experimental results on the real flight dataset demonstrate the effectiveness of the proposed method compared to previous algorithms.

Keywords Outlier detection · Flight data · LSTM · Multi-task learning

1 Introduction

Flight data analysis is an important research field for flight status monitoring. Outlier detection of flight data has attracted increasing attention because it is of great importance for flight security. With the development of neural networks, deep learning-based methods have been widely applied in the time series data outlier detection task and achieved optimal performance on flight data processing compared to traditional methods.

To address the outlier detection task in the flight data, an intuitive method is distance-based algorithms. These algorithms usually calculate the distance between samples, then take the points that have a high distance from other points as the outliers [1–3]. These methods have been applied in some areas, such as monitoring the status of temperature sensors [2], engine [4], and operation errors [3]. Another way is using one-class support vector machine (OC-SVM) [5] or support vector data description (SVDD) [6] to map inlier samples into a hyper-space by kernel functions or deep learning networks [7–10]. These kernel-based methods perform well on small-scale datasets, however, they usually suffer from dimension explosion when dealing with high dimensional data. Also, the feature learning procedure is usually supervised only by the loss from the hyper-sphere, making them tend to learn a trivial solution. Besides, they usually neglect the detail features that also distinguishable for outliers.

Many studies also introduce deep learning models into time series outlier detection tasks because of their powerful feature learning ability and superior performance in classification [11–14] tasks. In this area, LSTMs become the

✉ Congan Xu
xcatougao@163.com

Long Gao
1017730430@qq.com

Fengqin Wang
nudtwangfengqin@163.com

Junfeng Wu
patrickwu0609@163.com

Hang Su
shpersonal_email@163.com

1 College of Air Common Service, Naval Aviation University, Yantai 264001, Shandong, China

2 Advanced Technology Research Institute, Beijing Institute of Technology, Jinan 250300, Shandong, China

3 College of Basic Sciences for Aviation, Naval Aviation University, Yantai 264001, Shandong, China

basic models because they can well depict the spatio-temporal correlation in sequence data. For example, some methods propose to detect outliers by predicting next time point data using LSTM [15–17]. The work [18] proposes a spatio-temporal correlation-based LSTM to realize the mapping among different parameters. The author also generate a dataset by injecting anomaly samples on real UAV flight data and verified the proposed method. Some methods propose to detect outliers by reconstructing inputs [19, 20]. Adversarial learning is also introduced into this task to promote the model fitting the distribution of inliers [21, 22]. For example, LSTM–GAN [22] is proposed to enhance the distribution learning ability of LSTM by designing a cycle structure. These deep learning-based methods usually focus on learning descriptive features by fitting the distribution of inlier samples. Thanks to the powerful feature learning abilities of CNNs, deep learning-based outlier detection methods have obtained optimal performance compared to traditional methods. However, these methods usually neglect to explicitly learn semantic features. Besides, they usually suffer from the overfitting problem in which outlier samples also can be reconstructed well, thus making outliers and inliers indistinguishable.

To address these problems in the outlier detection of flight data, in this paper, we propose a novel multi-task-based outlier detection method that can explicitly learn descriptive and semantic features simultaneously. The proposed model includes an LSTM-based autoencoder and a constraining layer. The LSTM-based autoencoder can match the distribution of inliers by reconstructing the inputs, while the constraining layer added in the bottleneck feature layer can construct a tight hyper-sphere by pulling all features of inliers together. By jointly optimizing two branches, the model can reach a balance between semantic and descriptive features, and achieve higher performance on the outlier detection tasks. Experimental results on the public flight dataset demonstrate the effectiveness of the proposed method.

The contribution of this work can be summarized as followings:

- We propose a novel outlier detection framework for flight data by adding constraints on the semantic features learned by the LSTM model.
- The proposed multi-task-based method can jointly learn descriptive and semantic features by optimizing two branches of the model.
- Experimental results on the public UAV flight data demonstrate the effectiveness of the proposed method compared to previous models that only focus on learning descriptive or semantic features.

2 Related works

2.1 Outlier detection schemes

Previous outlier detection works for time series data can be mainly divided into two schemes, learning semantic features and descriptive features. In this part, we summarize the related works of these two schemes as followings:

Some studies focus on learning semantic features by mapping the normal series samples into a compact feature space [23]. Two classical methods are OCSVM [5] and SVDD [6], which map the inlier samples into a hyperplane or hypersphere by kernel functions [8, 24]. For instance, [24] employs OCSVM to detect the outliers in the data collected from engines of aircrifts. [25] proposes to use OCSVM to detect outliers for predicting the risk of fatal accidents. [26] decreases the kernel matrix size in SVDD to accelerate the learning speed for large-scale datasets. Some methods also combine deep learning models with these methods by replacing the kernel functions with neural networks [27, 28]. These methods generally suppose that the classification boundary would be distinguishable by mapping features of inliers into a hyperspace that represents the normal class. However, the feature learning methods in this scheme are usually supervised only by the hyperspace, while they usually neglect to preserve information of the inputs, making them suffer from trivial solutions (e.g., mapping all features into 0) sometimes.

Some studies focus on learning descriptive features by fitting the distribution of inliers. The basic model of many methods in this scheme is LSTM-AutoEncoder (LSTM-AE), which compresses inputs using an encoder and reconstructs the inputs using a decoder [20, 29, 30]. Some components are employed to enhance the performance, such as adversarial training [22]. These methods generally suppose that the classification boundary would be more distinguishable by fitting the distribution of inliers better. Thus the autoencoder structure is used as the basic model to fit the distribution of given inliers samples [31]. The Variational AutoEncoder is also applied to detect outliers by reconstructing inputs. For example, the work [32] develops a Convolutional Variational AutoEncoder-based method (CVAE). Similar to the AutoEncoder-based method, CVAE employs the encoder to compress the input samples, and the decoder to reconstruct input samples. The reconstruction loss is calculated to identify anomalies. These deep learning-based methods obtain optimal performance for the flight data outlier detection task because of their powerful ability in learning representations. However, compared to traditional methods such as OCSVM, they neglect to explicitly learn the semantic information, which limits the performance of these methods. Also, these models sometimes can reconstruct outlier samples well, making outliers indistinguishable.

Multi-task structure networks have been widely applied in a variety of research fields such as multi-modal learning [33, 34] and classification [35, 36] tasks. Some works also propose to obtain sufficient representations by jointly learning different aspects features [37, 38]. Adding the constraint is a way to compress the information and promote the model to learn features [39, 40]. Previous two schemes works usually focus on either learning descriptive or semantic features, resulting in insufficient utilize of information. Inspired by these works, the proposed multi-task method can be regarded as a combination of the above-mentioned two schemes. The proposed method includes an LSTM-based autoencoder and a constraining layer. The LSTM-based autoencoder focuses on learning the distribution by reconstructing inputs, while the constraining layer focuses on learning the hyper-sphere by pulling all features together. By designing two branches to map features into a hypersphere as well as reconstructing the inputs, the model can explicitly learn and optimize both semantic and descriptive features for inliers. In the testing stage, the model can reconstruct inlier samples with small reconstruction loss, while collapse when dealing with outliers, thus making them more distinguishable.

2.2 LSTM

LSTM is an extension of recurrent neural networks (RNNs), it shows remarkable performance on the time series data processing [41, 42]. It includes three logistic sigmoid gates and one tanh layer. In LSTM, the hidden layer is a gated unit or gated cell. Suppose in time step t, the input vector of a LSTM network is x_t, the output is h_t, and the memory unit is c_t, the forget gate can be formulated as:

$$f_i = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{1}$$

The input gate can be formulated as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{2}$$

$$\widetilde{c_t} = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \widetilde{c_t} \tag{4}$$

The output gate can be formulated as:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \otimes tanh(c_t) \tag{6}$$

where $\otimes$, W, σ and b represent the vector element multiplication, weight, sigmoid function and bias, respectively.

The different gates inside LSTM boost its capability for capturing non-linear relationships for time-series datasets. In LSTM, the input value of a time point will be forwarded and merged with the next point, thus the hidden layer can learn information about all previous input points. By training the model, the hidden layer can preserve information about time points related to the task.

3 The proposed methods

3.1 Problem definition

The outlier detection task in the flight time-series data focuses on detecting sample points that have a different distribution from the inlier points. Formally, this task focuses on learning a model M from a given sample x belonging to the inlier class c. In the testing stage, the model can obtain a score s for a given sample x, where a lower score represents sample x has a higher probability belonging to class c.

3.2 Pipeline

The structure of the proposed method is a multi-task-based method that can both learn semantic and descriptive features by jointly optimizing two branches. It is composed of an encoder (yellow), a decoder (blue), and a constraining layer (black), as shown in Fig. 1.

Specifically, similar to the Deep-SVDD [9], the encoder encodes input samples into the semantic feature z, where z constructs a hypersphere for inlier samples. The decoder is trained to reconstruct the input samples by taking feature z as the input, thus preventing the model from mapping all samples into a trivial solution. The proposed constraining layer is added to the feature z to pull all features toward the mean value of z (the red dot denoted by $\overline{z}$).

Based on the analysis, the loss functions of the two branches can be formalized as followings. For the LSTM that includes an encoder and decoder, suppose m is the length of the input sequence, the loss of learning *s*emantic features by pulling all feature points together can be defined as:

$$L_s = \|z_i - \overline{z}\|^2 = \|E(x_i) - \overline{z}\|^2 \tag{7}$$

where L_s is the semantic feature learning loss. $\vec{x}$ and E represent the input and the encoder network, respectively.

For the constraining layer, the loss of learning *d*escriptive features by reconstructing inputs can be defined as:

$$L_d = \sum_{i=1}^{i=m} \|D(E(x_i)) - x_i\|_2^2 \tag{8}$$

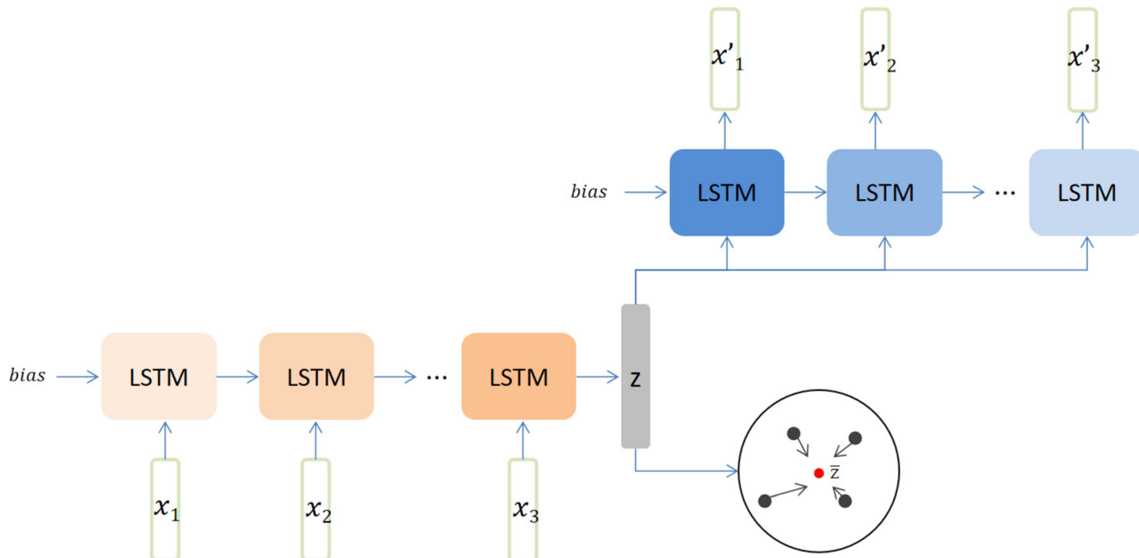

Fig. 1 The structure of the proposed method. It includes an Encoder (yellow), a Decoder (blue) and a constraining layer (black). The whole model focuses on reconstructing the input x (Color figure online)

where L_d is the descriptive feature learning loss. D represents the decoder network, respectively.

The whole loss L with two above-mentioned losses is

$$L = L_d + \gamma L_s \tag{9}$$

where γ is a coefficient to balance the influence of L_d and L_s.

The benefits of the proposed multi-task-based structure can be summarized as two aspects:

Compared to descriptive feature learning-based methods such as LSTM-AE, the proposed method can explicitly learn a tight hyper-sphere by training the encoder with the loss value generated from the constraining layer, thus describe the semantic information of inliers better.

Compared to semantic feature learning-based methods such as SVDD, the proposed method can prevent the model from falling into trivial solutions by reconstructing inputs with meaningful features.

In our method, the joint training procedure can reach a balance for a given task to achieve optimal performance, thus satisfying different requirements of different tasks.

3.3 Weights optimization

The weights of the encoder and decoder can be jointly optimized by reconstructing loss and constraining loss. Specifically, the weight of the decoder W_d can be optimized by the following formula:

$$W_d = W_d - \frac{\lambda}{b}\sum_{i=1}^{b}\frac{\partial L_{AE}}{\partial W_d} \tag{10}$$

where λ, b are the learning rate and batch size, respectively. The weights of the encoder W_e can be optimized by the following formula:

$$W_e = W_e - \frac{\lambda}{b}\sum_{i=1}^{b}\left(\frac{\partial L_{AE}}{\partial W_d} + \gamma\frac{\partial L_{con}}{\partial z_i}\right) \tag{11}$$

3.4 Outlier score calculation

The score is used to measure the probability of a given sample belonging to the outliers. Considering that the decoder can amplify the difference between inliers and outliers, we employ the reconstruction error as the outlier score. Specifically, the outlier score s of a given sample x can be calculated by the following formula:

$$s = \sum_{i=1}^{i=m}\|D(E(x_i)) - x_i\|^2 \tag{12}$$

3.5 Quantitatively analysis

In this section, we discuss a preliminary quantitative analysis to show that the proposed method has an optimal constraint $\hat{c}$ that can make outliers and inliers most distinguishable.

Researches have proposed some theories to analyze the principle of self-encoder [43]. The most famous and widely accepted method is the information bottleneck theory.

Suppose that D is the information that a given network preserves, c is the information coding rate of the given sample, for a given inlier sample p and outlier sample n, suppose s_p, s_n represent the reconstruction losses of samples p and n, respectively, we have two extreme conditions:

If c=100%, the network can preserve all information, thus D=100%. In this condition, $s_p = 0$ and $s_n = 0$, which means p and n are unable to be distinguished by the reconstruction loss.

If c=0%, the network can not learn any information, thus D=0%. In this condition, the difference of the reconstruction losses is the sum of pixels difference of samples p and n.

In the outlier detection task, CAE which has a bottleneck layer is a method to compress the information. Suppose D_p and D_n are the information learned by a given CAE model for sample p and n. We have $D_p\, gt\, D_n$ because the model is trained by inlier samples.

Adding the constraint is a way to compress the information. Suppose Δp and Δn represent the change of the information for inlier and outlier samples when the coding rate changes Δc. Under a weak constraint, $\Delta p\ \Delta n$, which means adding the constraint makes the information of outliers reduce faster. But considering that in the extreme condition when c=0%, we have D=0%, so under a strict constraint, $\Delta p > \Delta n$. Thus formally, there exists a coding rate $\hat{c}$:

$$if \quad c > \hat{c}, \quad then \quad \Delta p < \Delta n; \tag{13}$$

$$f \quad c < \hat{c}, \quad then \quad \Delta p > \Delta n; \tag{14}$$

where $\hat{c}$ represents the coding rate when the information change of inliers and outlier samples are the same.

A model can obtain the optimal performance when the information of the inliers and outliers have the maximum difference. Considering that the difference of reconstruction losses has the direct ratio to the preserved information, we have:

$$\begin{aligned} \Delta s &= s_p - s_n \\ &\propto D_c^p - D_c^n \\ &= \int_c^{100} \Delta p \; dc - \int_c^{100} \Delta n \; dc \\ &= \int_c^{100} (\Delta p - \Delta n) dc \end{aligned} \tag{15}$$

Considering Eqs. 13 and 14, Eq. 15 can obtain the optimal value when $c = \hat{c}$, where the inliers and outliers have the largest information difference, thus making them most distinguishable.

4 Experiments

4.1 Datasets

To evaluate the performance of the proposed method, we use the public data provided by the University of Minnesota UAV Laboratory.[1] The original dataset includes a variety of parameters recorded in the flight procedure. Considering that the flight height is a significant importance characteristic for flight safety, this paper focuses on detecting outliers for the height. This dataset includes six flights, where all points of the height values in the first three flights are normal samples, while the other three flights includes some outlier points. The details of the dataset are shown in Table 1.

The time-height curves of six flights are shown in Fig. 2. We can see that curves of flights #97, #98, and #104 are stable, while there exists three outlier series in flight #111 (the red curve), and one outlier series in flights #112 and #121. These outliers might be caused by the mistakes of GPS, which would result in error flight operations and thus influence flight safety. In this work, we focus on detecting the above-mentioned outlier series.

4.2 Experimental settings

In the experiment, Keras [44] and TensorFlow [45] are used to implement the algorithm code. Adam [46] is employed as the optimizer, with the learning rate setting to 0.001. The latent feature dimension is set to 20. The model is trained for 2000 epochs. Considering that updating the encoder by the loss from the constraining layer would influence the learning procedure of the autoencoder, to stable the training procedure, the balance coefficient of two losses γ is set to 0.1. The window sliding [47] is used for the data, with the window size setting to 20.

The training procedure of the proposed model is shown in Algorithm.

Algorithm 1 Training procedure of the proposed algorithm.

Initialize the model M, number of epochs(nb_epoch), batch_size;
for epoch=1:nb_epoch
  for i=1:nb_batch
    Update W_e and W_d by training the Encoder and Decoder of M;
    Calculate the center point of features $\overline{z}$ extracted from inliers;
    Update W_e by training the Encoder with Constraining layer;
  end
end
Obtain model M;
Input testing time series sample x_t, calculate the reconstruction loss and obtain the score s_t.

[1] Retrieved from University of Minnesota Digital Conservancy, 2022. Available at https://conservancy.umn.edu/handle/11299/163580.

Table 1 Dataset statistics

ID	#	Points num	Outlierpoints	Outliers number
1	97	14794	–	0
2	98	21015	–	0
3	104	25836	–	0
4	111	13327	740–1842 3340–3387 4967–5035	1220
5	112	10646	1316–1740	425
6	121	27097	3558–3881	324

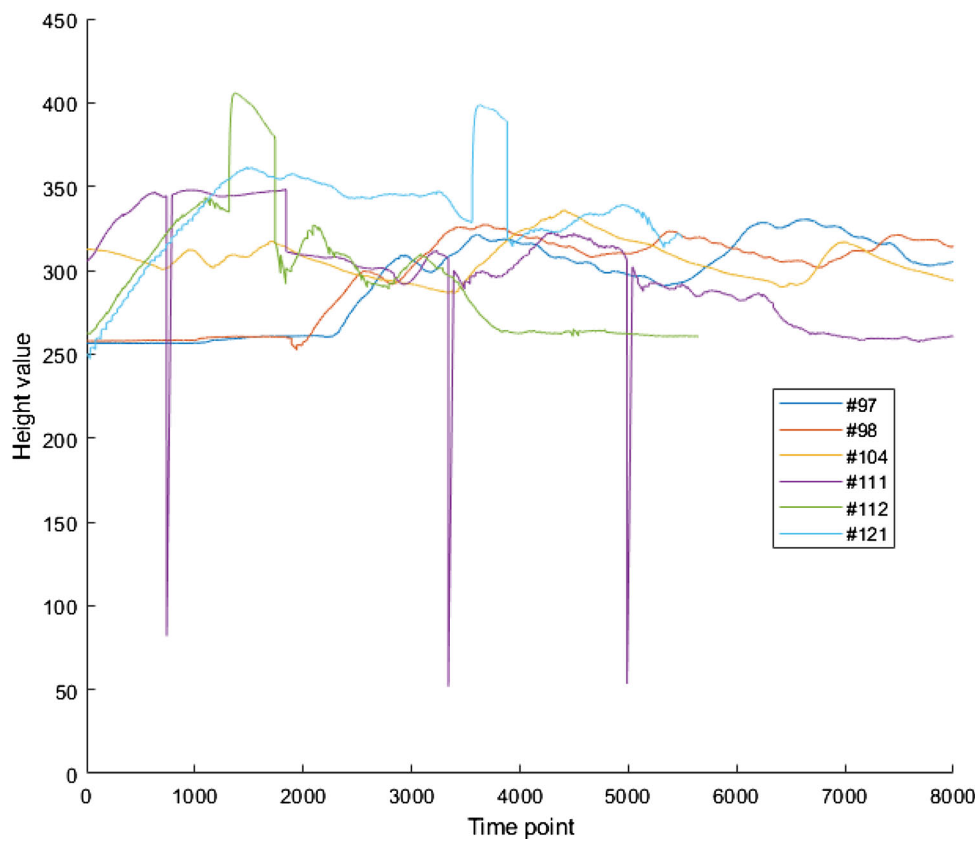

Fig. 2 The curve of time-height. There exists outlier areas in flights #111, #112 and #121

4.3 Compared methods

The proposed method is compared with the following commonly used methods:

(1) *K*-Means: *K*-Means is a commonly used method to detect outliers by measuring the distance of a given sample to the clustering center. Considering that there is only one class in the training set, we first use *K*-Means to cluster inlier samples and obtain the clustering center point. Then the distance between samples and the center point is taken as the abnormality score of a given sample.

(2) OCSVM: OCSVM focuses on mapping the inputs into a hyper-plane by kernel functions. Here the RBF kernel is used for all experiments in this paper. RBF kernel includes two hyper-parameters, penalty coefficient c_{RBF} and fitting coefficient γ_{RBF}. In this experiment, the default values of c_{RBF} and γ_{RBF} are used for all experiments.

(3) LSTM-AE: This is the basic model of our proposed method, which uses the LSTM model to train an autoencoder network. To compare with the proposed method, in this experiment, the network structure of the LSTM is the same as the proposed method.

(4) AutoEncoder: AutoEncoder (AE) is a multi-layer perceptron network that focuses on reconstructing the inputs by an encoder-decoder structure. In this experiment, we take the input as a feature vector and the model is trained to reconstruct the input. To compare the performance better, we use the same network nodes number and layers number as the

Table 2 Comparison of our method to other previous algorithms

Data	Metrics	*K*-Means	OCSVM	LSTM-AE	AE	LSTM-GAN	OURS
#111	ACC	70.44	22.18	76.93	77.00	78.34	79.16
	Pre	77.89	–	78.79	78.81	79.94	80.12
	Rec	92.55	–	93.2	93.3	94.62	93.41
	F1	84.59	–	85.39	85.45	86.66	86.26
#112	ACC	90.27	7.73	90.72	92.01	92.56	92.46
	Pre	92.27	7.73	92.15	92.9	93.82	94.21
	Rec	95.44	7.73	94.33	94.69	98.42	98.23
	F1	93.83	7.73	93.23	93.79	96.06	96.18
#121	ACC	90.57	5.89	90.91	88.83	95.60	95.47
	Pre	93.89	5.89	92.17	91.99	96.46	96.87
	Rec	94.23	5.89	94.52	95.37	98.95	97.46
	F1	94.06	5.89	93.33	93.65	97.69	97.16

proposed method to ensure the two models have a similar structure.

(5) LSTM-GAN: Integrating adversarial training is a commonly-used method to enhance the learning abilities of autoencoder models. LSTM-GAN uses the LSTM-based autoencoder as the basic model, and introduces adversarial training to enhance the performance. This method also designs a Cycle-GAN structure, in which the Gaussian noise is also input into the generator to enhance the model's learning ability. By training the model in a cycle framework, this method can match the distribution of inliers better, thus making outliers and inliers more distinguishable. In this method, to compare the performance better, the generator part has the same structure as our proposed method, while the discriminator has the same structure as the encoder of the generator, except that the last layer is replaced by a full connection layer to realize binary-class classification.

4.4 Evaluation metrics

For all experiments, the Accuracy (*Acc*), Precision (*Pre*), Recall (*Rec*), and $F1$ score ($F1$) are used to evaluate the performance of different algorithms. The definition of these evaluation metrics are shown as followings:

$$Acc = \frac{TN + TP}{TN + TP + FN + FP} \tag{16}$$

$$Pre = \frac{TP}{TP + FP} \tag{17}$$

$$Rec = \frac{TP}{TP + FN} \tag{18}$$

$$F1 = \frac{2 * Pre * Rec}{Pre + Rec} \tag{19}$$

where *TP* represents an inlier sample is classified into the inlier class, *FP* represents an outlier sample is classified into the inlier class, *TN* represents an outlier sample is classified into the outlier class, and *FN* represents an inlier sample is classified into the outlier class.

4.5 Result of different algorithms

The results of all methods are shown in Table 2. From Table 2 we can see that generally the performance of all methods on #111 is lower than #112 and #113, indicating that the first dataset is a relatively difficult task. The recall values are higher than the precision values among all experiments, this might be because outliers occupy a relatively low proportion in the whole dataset. Deep-learning-based methods perform better than *K*-Means and OCSVM, validating the effectiveness of deep-learning-based methods. OCSVM method collapse in these three tasks, which might because OCSVM maps all inputs into a trivial hyper-space. LSTM-GAN achieves higher performance compared to LSTM-AE, showing that adversarial training can improve LSTM to fit the distribution, thus making samples more distinguishable. The proposed method obtains the optimal performance on most of the evaluation metrics, validating the effectiveness of the proposed method. Especially, our method obtains better performance compared to LSTM-AE, demonstrating that adding constraint can promote the LSTM-AE to learn better representations.

4.6 Hyper-parameter analysis

The proposed method introduces hyper-parameters learning rate λ and γ, where λ is the learning rate of the autoencoder, γ is the coefficient to balance reconstruction loss and constraining loss. These hyper-parameters would influence the performance of the proposed method. To explore the influence of hyper-parameters, we also test the performance when setting $\lambda \in \{10^{-3}, 10^{-4}\}$ and $\gamma \in \{1, 0.1, 0.01\}$. Considering that from Table 2 we can observe that flight #111 is a relatively hard task, here we take flight #111 as an example to conduct all experiments.

The results are shown in Fig. 3. From the figure we can see that generally the performance when setting $\lambda = 10^{-3}$ is better than $\lambda = 10^{-4}$. The proposed method obtains high performance compared to the LSTM-AE in most hyper-parameter settings, validating the constraining layer can learn better classification margin, and our method is stable with different settings.

5 Conclusion

To detect outliers for flight data, this paper proposes a novel multi-task structure to simultaneously learn descriptive features and semantic features. By jointly optimizing the two branches of the proposed method, the proposed method can learn a classification boundary to make outliers and inliers more distinguishable. Experimental on the real flight data validates the effectiveness of the performance. Hyper-parameter analysis shows the proposed method is robust to different settings. As a network layer, the proposed method can be integrated into other LSTM-based outlier detection models. In the future, we will explore more experiments and analyze the

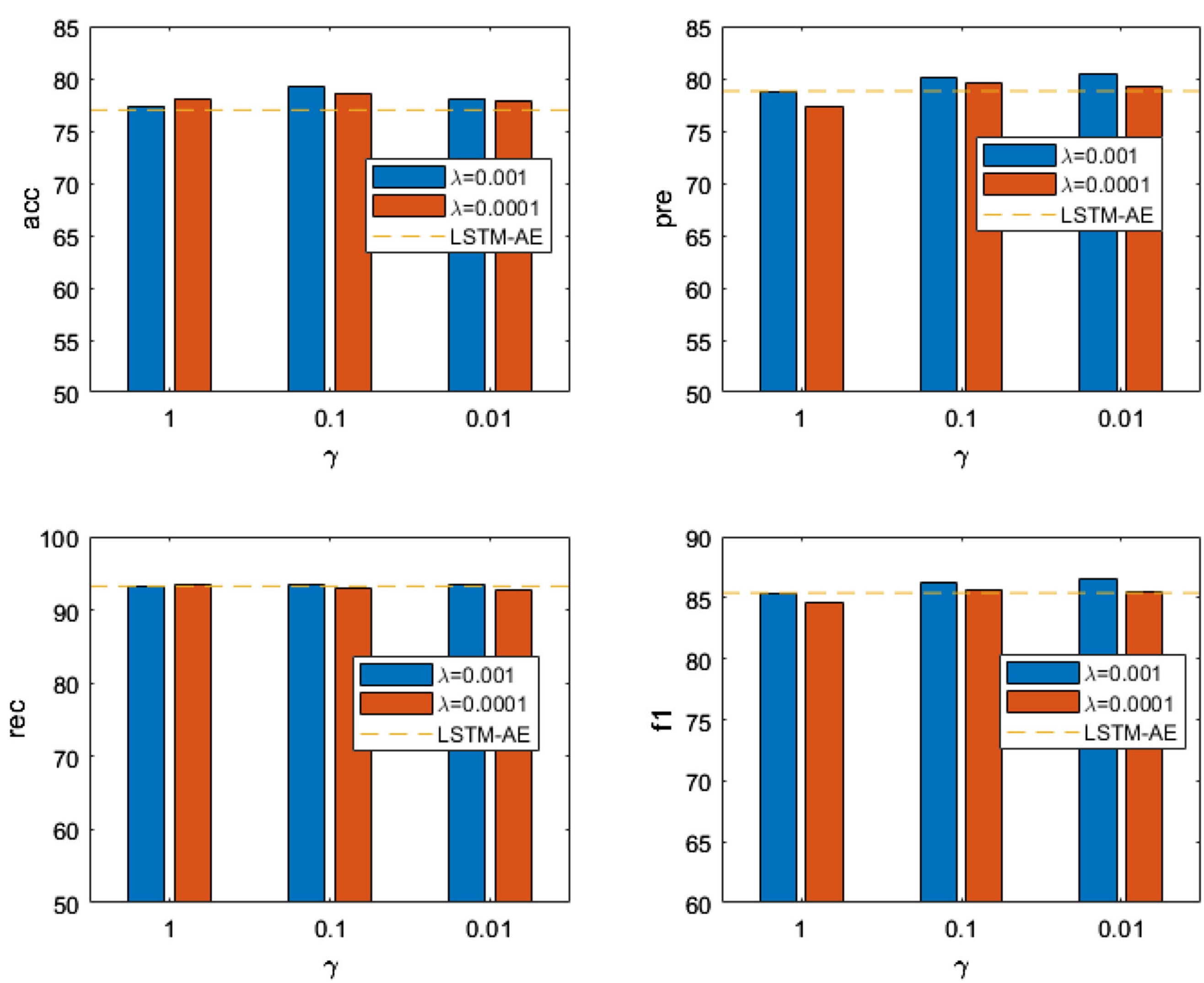

Fig. 3 The results with different hyper-parameters. the yellow dotted line is the LSTM-AE as the baseline

influence of hyper-parameters to validate the effectiveness of our method.

Acknowledgements This work is supported by The National Natural Science Foundation of China (No.62271499, No.61971432, No.62022092), The Young Elite Scientists Sponsorship Program by CAST (2020-JCJQ-QT-011).

Author contributions Long Gao proposed the idea and conducted the experiments. Congan Xu optimized the method and checked the manuscript. Fengqin Wang conducted the experimental comparison and checked the manuscript. Junfeng Wu and Hang Su discussed the idea and wrote the initial version of the paper.

Data availability The dataset used in the manuscript is a public dataset that can be found at: https://conservancy.umn.edu/handle / 11299/163580.

Declarations

Conflict of interest The authors declare no competing financial interests.

References

1. Budalakoti, S., Srivastava, A. N., & Otey, M. E. (2009). Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems Man and Cybernetics Part C, 39*(1), 101–113.
2. Matthews, B., Srivsatava, A.N., Schade, J., Schleicher, D.R., & Kiniry, M. (2013). Discovery of abnormal flight patterns in flight track data. In: 2013 Aviation Technology, Integration, and Operations Conference.
3. Li, L., Das, S., Hansman, R. J., Palacios, R., & Srivastava, A. N. (2015). Analysis of flight data using clustering techniques for detecting abnormal operations. *Journal of Aerospace Computing, Information, and Communication, 12*(9), 587–598.
4. Bay, S.D., & Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of the Ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2003)
5. Olkopf, B.S., Williamson, R., Smola, A., Shawe-Taylor, J., & Platt, J. (2000). Support vector method for novelty detection. In: Advances in Neural Information Processing Systems.
6. Tax, D., & Duin, R. (1999). Support vector domain description. Pattern recognition letters.
7. Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition, 62*, 121–134.
8. Camerini, V., Coppotelli, G., & Bendisch, S. (2018). Fault detection in operating helicopter drivetrain components based on support vector data description. *Aerospace Science and Technology, 73*, 48–60.
9. Ruff, L., Vandermeulen, R. A., Görnitz, N., Deecke, L., & Kloft, M. (2018). Deep one-class classification. *International Conference on Machine Learning, 24*, 11053.
10. Rong-Xiao, G. (2021). Anomaly detection method for UAV sensor data based on LSTM–OCSVM. *Journal of Chinese Computer Systems, 85*, 700–705.
11. Tu, Y., Lin, Y., Zha, H., Zhang, J., Wang, Y., Gui, G., & Mao, S. (2022). Large-scale real-world radio signal recognition with deep learning. *Chinese Journal of Aeronautics, 35*(9), 35–48.
12. Bao, Z., Lin, Y., Zhang, S., Li, Z., & Mao, S. (2022). Threat of adversarial attacks on dl-based IoT device identification. *IEEE Internet of Things Journal, 9*(11), 9012–9024. https://doi.org/10.1109/JIOT.2021.3120197
13. Liu, S., Li, Y., & Fu, W. (2022). Human-centered attention-aware networks for action recognition. *International Journal of Intelligent Systems, 37*, 10968–10987.
14. Wu, Q., Li, Y., Lin, Y., & Zhou, R. (2018). Weighted sparse image classification based on low rank representation. *CMC Computer Materials Continua, 7*, 15.
15. Luo, W., Wen, L., & Gao, S. (2017). Remembering history with convolutional LSTM for anomaly detection. In: 2017 IEEE International conference on multimedia and expo (ICME)
16. Shabtai, A., & Habler, I. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages
17. Jianli, D., Yunkai, Z., Jing, W., & Huaichao, W. (2019). Ads-b anomaly data detection model based on deep learning. *Acta Aeronautica et Astronautica Sinica, 40*(11), 452.
18. Zhong, J., Zhang, Y., Wang, J., Luo, C., & Miao, Q. (2022). Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatio-temporal correlation. *IEEE Transactions on Reliability, 1*, 71.
19. Reddy, K.K., Sarkar, S., Venugopalan, V., & Giering, M. (2016). Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach. In: Annual Conference of the PHM Society.
20. Wang, X., Du, Y., Lin, S., Cui, P., Shen, Y., Yang, Y. (2019). Self-adversarial variational autoencoder with gaussian anomaly prior distribution for anomaly detection.
21. Donahue, J., Krhenbühl, P., & Darrell, T. (2016). Adversarial feature learning.
22. Wang Fengqin, W. L., & Long, G. (2022). UAV flight data anomaly detection algorithm based on LSTM–GAN. *Journal of Chinese Inertial Technology, 45*, 264–271.
23. Villa-Peacuterez, M.E.A.-C. (2021). Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *Knowledge-Based Systems, 785*, 106878.
24. Zhu, Y., Du, C., Liu, Z., Chen, Y.-B., & Zhao, Y. (2022). A turboshaft aeroengine fault detection method based on one-class support vector machine and transfer learning. *Journal of Aerospace Engineering, 35*, 04022085.
25. Jiang, Y., Liu, R., Le, N., & Zheng, Y. (2019). A method for the outlier flights detection of the final approach based on foqa data. In: 2019 IEEE 1st international conference on civil aviation safety and information technology (ICCASIT).
26. Zhanping, S. W. J. T. F. (2014). Flight data novelty detection method based on improved SVDD. *Chinese Journal of Scientific Instrument, 35*, 932–939.
27. Kou, L., Chen, J., & Qin, Y. (2022). The robust multi-scale deep-SVDD model for anomaly online detection of rolling bearings. *Sensors, 22*(15), 5681.
28. Zhou, Y., Liang, X., Zhang, W., Zhang, L., & Song, X. (2021). VAE-based deep SVDD for anomaly detection. *Neurocomputing, 453*, 131.
29. Ji, Y., Wang, L., Wu, W., Shao, H., & Feng, Y. (2020). A method for LSTM-based trajectory modelling and abnormal trajectory detection. *IEEE Access, 99*, 1–1.
30. Fried, A., & Last, M. (2021). Facing airborne attacks on ADS-b data with autoencoders. *Computers and Security, 109*(2), 102405.
31. Zhang, W., Hu, M., & Du, J. (2022). An end-to-end framework for flight trajectory data analysis based on deep autoencoder network. *Aerospace Science and Technology, 127*, 107726.

32. Memarzadeh, M., Matthews, B., & Avrekh, I. (2020). Unsupervised anomaly detection in flight data using convolutional variational auto-encoder.
33. Liu, S., Gao, P., Li, Y., Fu, W., & Ding, W. (2022). Multi-modal fusion network with complementarity and importance for emotion recognition. *Information Sciences, 619*, 679–694.
34. Novitasari, S., Do, Q.T., Sakti, S., Lestari, D., & Nakamura, S. (2018). Multi-modal multi-task deep learning for speaker and emotion recognition of tv-series data. In: 2018 Oriental COCOSDA—international conference on speech database and assessments.
35. Xing, H., Xiao, Z., Qu, R., Zhu, Z., & Zhao, B. (2022). An efficient federated distillation learning system for multi-task time series classification. *IEEE Transactions on Instrumentation and Measurement, 1*, 452.
36. Ling, C., Donghui, C., Fan, Y., & Jianling, S. (2021). A deep multi-task representation learning method for time series classification and retrieval. *Information Sciences, 555*, 17–32.
37. Fu, X., Peng, Y., Liu, Y., Lin, Y., Gui, G., Gacanin, H., & Adachi, F. (2023). Semi-supervised specific emitter identification method using metric-adversarial training. *IEEE Internet of Things Journal, 12*, 1–1.
38. Zhu, Q., Chen, J., Shi, D., Zhu, L., Bai, X., Duan, X., & Liu, Y. (2020). Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction. *IEEE Transactions on Sustainable Energy, 11*, 509–523.
39. Wu, Q., Li, Y., & Lin, Y. (2019). Medical image restoration method via multiple nonlocal prior constraints. *Journal of Intelligent and Fuzzy Systems, 38*(4), 1–15.
40. Zheng, L., Hongzhi, W., Xiaoou, D., & Tianyu, M. (2021). Industrial time series determinative anomaly detection based on constraint hypergraph. *Knowledge-Based Systems, 233*, 107548.
41. Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry, 131*(3), 103498.
42. Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics, 228*(10), 4568.
43. Geiger, B. C., & Kubin, G. (2020). Information bottleneck: Theory and applications in deep learning. *Entropy (Basel, Switzerland), 22*, 1408.
44. Chollet, F. (2015). Keras. https://keras.io.
45. Abadi, M., Barham, P., Chen, J., Chen, Z., & Zhang, X. (2016). *Tensorflow: A system for large-scale machine learning*. USENIX Association.
46. Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. CoRR **abs/1412.6980**.
47. Hou, C., Liu, G., Tian, Q., Zhou, Z., Hua, L., & Lin, Y. (2022). Multisignal modulation classification using sliding window detection and complex convolutional network in frequency domain. *IEEE Internet of Things Journal, 9*(19), 19438–19449.

Long Gao received the Ph.D. degree in College of Computer from National University of Defense Technology, Changsha, China, in 2020. He is currently a lecture in the College of Air Common Service, Naval Aviation University, Yantai, China. His current research interests include machine learning, target detection, and ship recognition.

Congan Xu received the M.S. and Ph.D. degrees from Naval Aviation University, Yantai, China, in 2013 and 2016, respectively. He is currently an Associate Professor of Naval Aviation University. He has undertaken more than 10 projects, including National Natural Science Foundation of China. His research interests include intelligent perception and fusion, and deep learning and its application.

Fengqin Wang received M.S. degree in Computer science and technology from National University of Defense Technology and Ph.D. degree in Science of Military Equipment from Naval Aviation University.She is an associate professor, mainly engaged in the analysis and Application Research of flight big data. She is the author of three books, more than 50 articles, and more than 4 inventions.

Junfeng Wu received the Ph.D. degree in College of Beihang University, Beijing, China, in 2016. He is currently an associate professor in the College of Air Common Service, Naval Aviation University, Yantai, China. His current research interests include UAV cluster situation awareness, target detection and recognition.

Hang Su received the M.S degree in the College of Air Common Service from Naval Aviation University, Yantai, China, in 2022. His current research interests include deep-learning, target detection, and ship recognition.