GAN-FDSR: GAN-Based Fault Detection and System Reconfiguration Method

Zihan Shen, Xiubin Zhao, Chunlei Pang * and Liang Zhang

Information and Navigation School, Air Force Engineering University, Xi’an 710077, China; zh_shen520@163.com (Z.S.); zhaoxiubin926@163.com (X.Z.); zhangliang_nudt@nudt.edu.cn (L.Z.)
* Correspondence: master_s2020@163.com

Abstract: Fault detection and exclusion are essential to ensure the integrity and reliability of the tightly coupled global navigation satellite system (GNSS)/inertial navigation system (INS) integrated navigation system. A fault detection and system reconfiguration scheme based on generative adversarial networks (GAN-FDSR) for tightly coupled systems is proposed in this paper. The chaotic characteristics of pseudo-range data are analyzed, and the raw data are reconstructed in phase space to improve the learning ability of the models for non-linearity. The trained model is used to calculate generation and discrimination scores to construct fault detection functions and detection thresholds while retaining the generated data for subsequent system reconfiguration. The influence of satellites on positioning accuracy of the system under different environments is discussed, and the system reconfiguration scheme is dynamically selected by calculating the relative differential precision of positioning (RDPOP) of the faulty satellites. Simulation experiments are conducted using the field test data to assess fault detection performance and positioning accuracy. The results show that the proposed method greatly improves the detection sensitivity of the system for small-amplitude faults and gradual faults, and effectively reduces the positioning error during faults.

Keywords: GNSS/INS integrated system; generative adversarial networks; chaos; fault detection and identification; integrity monitoring

1. Introduction

The global navigation satellite system (GNSS)/inertial navigation system (INS) integrated system compensates well for the deficiencies of a single navigation system by fusing the information of GNSS and INS [1], so as to obtain more accurate and reliable navigation results, and is widely used in many fields such as agricultural automation, robot control, traffic safety, and unmanned driving [2–4].

Data fusion in the GNSS/INS integrated system is generally implemented using Kalman filtering algorithms [5], such as the cubature Kalman filter [6] and the unscented Kalman filter [7]. The INS is a self-contained dead-reckoning system that can accomplish navigation tasks independently of external information, which allows it to be free from environmental interference, but its errors will accumulate over time, leading to a rapid drop in positioning accuracy [8]. GNSS can provide the absolute position in real time, but its signal is fragile and susceptible to interference from many environmental factors, such as multipath effects and electromagnetic interference [9,10]. The integrated system exploits the complementary characteristics of both navigation systems, which allows it to obtain superior performance compared to a stand-alone INS or GNSS [11]. In practical navigation applications, complex observation environments often interfere with GNSS measurements. If the fault is not detected or eliminated in time, the performance advantages of integrated system disappears, and the accuracy and reliability of the system is reduced [12]. Therefore, a fault detection and system reconfiguration method that is adaptable to the environment and reliable is essential for the application of combined navigation [13].
Classical fault detection methods include the residual chi-square tests and the state chi-square tests, both of which are based on the Kalman filter [14]. They are simple to implement and detect faults quickly by judging the relationship between the distribution of the detection function and the fault threshold. Autonomous integrity monitored extrapolation (AIME) adds extrapolation cycles to the chi-square tests, and the addition of historical information enhances the detection sensitivity of the system [15]. The literature [16] combines support vector machine (SVM) with AIME to improve the ability to detect gradual faults. These methods focus on fault detection performance, and generally reconfigure the system through fault isolation in subsequent filtering. Furthermore, some methods have been proposed to reconfigure the disturbed system during fault detection. Teunissen proposed the detection, identification, and adaptation (DIA) method to detect faults and reconfigure the system [17,18], which uses the optimal model error estimation to correct the state estimation after identifying the fault to reduce the impact of the fault on the system. Modification of the classical extended Kalman filter (EKF) using robust M-estimation can effectively enhance the immunity of the system to interference [19]. The method based on the variance shift outlier model (VSOM) and EKF is able to handle single or multiple faults well in GNSS measurements by down-weighting only the identified faulty measurements while detecting faults in the raw pseudo-range data [20]. Setting two separate filters for GNSS faults and filter faults not only identifies and removes faulty measurements, but also eliminates the effects of filter faults [21]. However, the uncertainty and ambiguity of faults in complex environments will degrade the detection performance of these methods. In addition, reconfiguring the system by reducing the weight of failures alone poses a continuity risk in a low redundancy system.

With the substantial progress in artificial intelligence algorithms in recent years, neural networks are quite valuable in the field of fault detection [22]. With features such as adaptability, nonlinearity and robustness, neural networks are able to compensate for detection failures caused by traditional methods due to uncertainty and ambiguity of faults. The literature [23] combines Gaussian process regression with Kalman filtering to incorporate predicted innovation into the calculation of the fault detection function (FDF), which leads to better detection performance than with the residual chi-squared tests. The data-driven adaptive neuron fuzzy inference system (ANFIS)-based method [24] was applied to fault detection of on-board navigation sensors in unmanned aerial vehicles (UAVs) and achieved good results in terms of detection accuracy and misdetection rate. In the literature, deep neural networks have been added to the support vector machine fault detection algorithm [25]. The data fusion using the data predicted by the neural network to replace the faulty measurement when a fault occurs can effectively improve the stability of the system. Deep learning-based fault detection methods are mostly implemented based on time series prediction, and the accuracy of the prediction greatly affects the detection performance. In addition, using the difference between the current state and the predicted value to detect anomalies may be insufficient in complex and dynamic observation environments.

In recent years, generative adversarial networks (GAN) have achieved great success not only in image processing but also in time-domain sequence generation [26–29] by building deep learning models through adversarial training. Additionally, the generators and discriminators trained by adversarial training as well as the unsupervised learning technique hint at the feasibility of GAN applications in the field of fault detection [30,31]. In this paper, GAN is applied to the fault detection algorithm of a tightly-coupled GNSS/INS integration system for the first time, and a novel GAN-based fault detection and system reconfiguration method is designed which offers superior fault detection performance and more adaptable troubleshooting strategies. Unlike the method in [23–25] that uses the predicted values of neural networks to participate in fault detection, GAN-FDSR fully exploits the characteristics of adversarial training in the fault detection phase and designs a fault detection function based on the generation score and the discriminant score, which is more sensitive to small-amplitude faults and gradual faults. In the troubleshooting phase, a
dynamic selection strategy of fault isolation and system reconfiguration is designed by analyzing the relative differential precision of positioning (RDPOP) of satellites under different observation environments, which compensates for the accuracy degradation and integrity risk brought by traditional troubleshooting methods in poorer observation environments. Furthermore, considering that the integrated navigation data are nonlinear, we analyze the chaotic characteristics of the navigation data in the data preprocessing phase, and map the one-dimensional time series to the high-dimensional space through phase space reconstruction to improve the learning efficiency of the model for nonlinear characteristics.

This paper is organized as follows. In Section 2, the data preprocessing method is presented, including the mathematical model of GNSS/INS tight integration and the reconstruction method for chaotic sequences. Section 3 describes each part of the proposed method in detail. In Section 4, simulation experiments are carried out to analyze the performance of the proposed method under different environments. Finally, the summary for this paper is presented in Section 5.

2. Data Preprocessing

2.1. Tightly Coupled GNSS/INS Integration Model

In the tightly coupled GNSS/INS integration model, the INS module solves the navigation parameters separately, and then performs integrated filtering with the GNSS measurements, and the obtained estimated error is used to correct the INS output. The specific flow chart is shown in Figure 1.

![Figure 1. GNSS/INS tight integration architecture.](image)

Tightly coupled GNSS/INS integration contains two parts: the state model and the measurement model. The state model consists of the error state equations of INS and GNSS, which is usually expressed as:

\[ \dot{X} = FX + GW \]  \hspace{1cm} (1)

where \( F \) is the system dynamic matrix, \( G \) is the noise coefficient matrix, and \( W \) is the system noise vector.

The state vector in the filter is given by:

\[ X = [(\epsilon)^T, (\delta V)^T, (\delta P)^T, (\delta w_{ib})^T, (\delta f)^T, \delta t_u, \delta t_{ru}]^T \]  \hspace{1cm} (2)

where \( \epsilon, \delta V, \) and \( \delta P \) denote the INS error states of attitude, velocity, and position, respectively; \( \delta w_{ib} \) and \( \delta f \) represent the gyro and accelerometer bias vectors; \( \delta t_u \) and \( \delta t_{ru} \) denote the range bias and range drift related to the receiver clock, respectively.

The measurement model can be written as:

\[ Z = HX + V \]  \hspace{1cm} (3)

where \( Z, H \) and \( V \) donate the measurement vector, the measurement matrix and the measurement noise vector, respectively.
Assuming a discrete-time process, the state equation and measurement equation can be rewritten as follows:

\[ X_k = \Phi_{k,k-1} X_{k-1} + \Gamma_{k,k-1} W_{k-1} \]  
\[ Z_k = H_k X_k + V_k \]

where \( X_k \), \( \Phi_{k,k-1} \), \( Z_k \) and \( H_k \) are the state vector, transition matrix, the measurement vector and the measurement model matrix, respectively. \( W_{k-1} \) and \( V_k \) are the process noise and measurement noise, and both are zero-mean Gaussian white noise with process covariance matrices \( Q_k \) and \( R_k \).

State update in extended Kalman filtering consists of state estimation and one-step prediction of the covariance matrix, which can be given as follows:

\[ \hat{X}_k = \Phi_{k,k-1} \hat{X}_{k-1} \]  
\[ P_{k,k-1} = \Phi_{k,k-1} P_{k-1} \Phi_{k,k-1}^T + \Gamma_{k-1} Q_{k-1} \Gamma_{k-1}^T \]

where \( \hat{X} \) is the estimation of the state vector and \( P_{k,k-1} \) is its covariance matrix.

The measurement update in the filter is given by:

\[ \hat{X}_k = \hat{X}_{k,k-1} + K_k v_k \]  
\[ P_k = (I - K_k H_k) P_{k,k-1} \]

where \( K_k \) is the Kalman gain matrix, \( v_k \) is the residual vector, \( I \) denotes the identity matrix. \( K_k \) and \( v_k \) can be rewritten as follows:

\[ K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} \]  
\[ v_k = Z_k - H_k \hat{X}_{k,k-1} \]

2.2. Chaotic Characteristic Analysis and Phase Space Reconstruction

There are many factors that affect time series analysis, especially in dynamical systems where the dynamical equations of these factors are often nonlinear and more likely to be chaotic. If a sequence shows chaotic properties, reconstructing it in phase space is an important method to study the state of a dynamical system. The essence of this approach is to transform the extension of the time series into the interpolation of phase space. Based on the Takens’ embedding theorem [32], the optimized delay time and minimum embedding dimension are determined according to the actual time series. The one-dimensional time series can then be reconstructed into a phase space with the same topological meaning as the motive power system. A reasonable selection of the delay time and embedding dimension then allows reconstruction of the phase space with satisfactory adaptability, thereby producing the internal nonlinear mapping. By exploiting the properties of the Lyapunov exponent, this approach can recognize and make short-term predictions of chaotic time series [33,34].

The time delay and the embedding dimension are the keys to phase space reconstruction. However, there are no uniform selection criteria [35]. In this section, we will analyze the chaotic characteristics of navigation data, and use the mutual information method [36] and CAO method [37] to determine these parameters. Concerning the latter method, CAO is the last name of Professor Cao Liangyue, who proposed the method that can solve the embedding dimension with a small amount of data.

Here, \( \rho_C^k(\varepsilon) \) and \( \rho_C^k(\varepsilon) \) are defined as the GNSS pseudo-range and the INS pseudo-range of the \( k \)th satellite. The difference in the pseudo-range increment for the \( k \)th satellite can be calculated as:

\[ \rho_D^k(\varepsilon) = [\rho_C^k(\varepsilon + 1) - \rho_C^k(\varepsilon)] - [\rho_I^k(\varepsilon + 1) - \rho_I^k(\varepsilon)] \]

where \( \varepsilon = 1, 2, 3, \ldots, N \). A time series \( R = \{r_1, r_2, r_3, \ldots, r_N\} \) is obtained, where \( r_N = \rho_D^k(N) \).

Next, we take the data of the 8th satellite as an example to analyze the chaotic characteristics of \( R \). The details of the experimental data are presented in Section 4.
2.2.1. Determining the Time Delay

The mutual information method is a well-established method for determining the delay time which can describe the nonlinearity of the sequence and maximize the amount of information contained in the time series.

We expand the one-dimensional time series $R$ to the two-dimensional phase space $(A, B)$, and equate the phase space into $\beta \times \beta$ cells, where $A = \{r_1, r_2, r_3, \ldots, r_{N-\tau}\}$, $B = \{r_{\tau+1}, r_2, r_3, \ldots, r_N\}$, and the value interval of $\beta$ is $[100, 200]$ generally. The following formula can be obtained:

$$E(\tau) = -\sum_{ab} p_{ab}(\tau) \ln \frac{p_{ab}(\tau)}{p_a p_b}, \quad a, b = 1, \ldots, \beta$$

(13)

where $E(\tau)$ is the average information entropy of the time series; $p_a = \frac{\text{Row}[a]}{N-\tau}$ denotes the probability that the phase point of $(A, B)$ falls in the $a$th column cell; $p_b = \frac{\text{Col}[b]}{N-\tau}$ denotes the probability that the phase point falls in the $b$th row cell; $\text{Row}[a]$ and $\text{Col}[b]$ denote the counts of phase points in row $a$ and column $b$, respectively; $p_{ab}(\tau)$ is the joint probability; and $\tau$ is the time delay. As $\tau$ increases, $E(\tau)$ of $r_t$ and $r_{t+\tau}$ gradually decrease until a critical point where they are no longer correlated. This point is usually taken as the first minimal point.

Figure 2 shows the average information entropy at different delay times. At 10 s, the average information entropy reaches a minimal value for the first time, which indicates that the correlation between sequence fragments reaches a minimum at this time, so $\tau$ is 10 s.

![Figure 2. Average information entropy under different delay times.](image)

2.2.2. Finding the Embedding Dimension

Using the CAO method to find the embedding dimension enables the reconstructed time series to be free of information folding in the phase space, which is widely used in chaotic sequence analysis. The algorithm steps are as follows.

Starting from a smaller embedding dimension $m$, the $m$-dimensional and $m+1$-dimensional phase spaces are reconstructed. The ratio of Euclidean distances between adjacent points in phase space $h(i, m)$ is calculated as follows:

$$h(i, m) = \frac{\|r_i^{m+1} - r_i^{m+1}\|_\infty}{\|r_i^m - r_i^m\|_\infty}$$

(14)

where $r_i^{m+1}$ and $r_i^m$ represent the $i$th and $\hat{i}$th state points in the $m+1$-dimensional phase space; $r_i^m$ is the vector with the smallest Euclidean distance from $r_i^m$ in the $m$-dimensional phase space, $\hat{i} = 1, 2, \ldots, M$, where $M = N - (m - 1)\tau$ and $\hat{i} \neq i$; $\|r_i^m - r_i^m\|_\infty$ represents the Euclidean distance between $r_i^m$ and $r_i^m$.

The average of all ratios in $m$-dimensions is calculated as:
We then define 
\[ E_m = \frac{E(m+1)}{E(m)} \] 
(15)
to describe the change in \( E(m) \) when the embedding dimension is changed, and repeat the above steps until \( E_m = \frac{E(m+1)}{E(m)} \) is stable.

When \( m \) increases to a certain value, \( E_m \) converges to 1 if the sequence has chaotic properties. At this point the attractor of the time series has been completely opened in the phase space, and the increase of the embedding dimension will no longer make the signature sequence contain more information. The variation of \( E_m \) in each dimension according to the CAO method is shown in Figure 3. When the dimension reaches 9, the variation is dynamically stable in a small range. At this point, it can be considered that the reconstruction of the time series in this dimension can reflect the change pattern of the original system approximately in the phase space, so the embedding dimension is 9.

**Figure 3.** Using CAO method to find the embedding dimension.

### 2.2.3. Calculating the Lyapunov Exponent

The maximum Lyapunov exponent is an important index to judge whether the signal has chaotic characteristics. If the Lyapunov exponent is greater than 0, it means that no matter how small the initial distance between the two trajectories is in the phase space of the system, the difference will increase exponentially with time and eventually chaos will emerge. The steps for calculating the Lyapunov exponent are as follows:

1. Perform a fast Fourier transform (FFT) on the time series to obtain the average period \( P \);
2. Reconstruct \( R \) in the phase space according to \( m \) and \( \tau \). The obtained multidimensional sequence can be expressed as:
\[
Y = \begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_M
\end{bmatrix}
= \begin{bmatrix}
r_1 & r_{1+\tau} & r_{1+2\tau} & \cdots & r_{1+(m-1)\tau} \\
r_2 & r_{2+\tau} & r_{2+2\tau} & \cdots & r_{2+(m-1)\tau} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
r_M & r_{M+\tau} & r_{M+2\tau} & \cdots & r_{M+(m-1)\tau}
\end{bmatrix}
\] (16)
where \( Y \) is an \( M \times m \) dimensional matrix, \( Y_1 \) denotes the phase points, and \( M \) is the number of phase points in the reconstructed phase space;
3. Find the closest point \( Y_{\hat{j}} \) for each \( Y_j \) in the phase space and calculate the initial distance as follows:
\[
d_{j}(0) = \min \| Y_j - Y_{\hat{j}}\|, \quad \| j - \hat{j} \| > P
\] (17)
where \( \hat{j} = 1, 2, \cdots, M \) and \( \hat{j} \neq j \);
4. Calculate the distance between each $Y_j$ and the corresponding $\hat{Y}_j$ in the phase space after $i$ steps.

$$d_j(i) = \|Y_{j+i} - \hat{Y}_{j+i}\|, \ t = 1, 2, \cdots, \min(m - j, M - j)$$  \hfill (18)

5. Calculate the average of all $\ln d_j(i)$ for each $i$ as follows:

$$D(t) = \frac{1}{q\Delta t} \sum_{j=1}^{q} \ln d_j(i)$$  \hfill (19)

where $q$ is the number of non-zero $d_j(i)$, and $\Delta t$ represents the sample period. The slope of the line is the largest Lyapunov exponent when linear regression is performed by the least square method.

Table 1 lists the chaos indices of the visible satellites. Clearly, the time series $R$ of each visible satellite has chaotic characteristics in the experiment, and the data can be reconstructed using chaos theory.

Table 1. Chaos indices of the visible satellites.

| Satellites | $\tau$ | $m$ | Largest Lyapunov Exponent |
|------------|--------|-----|---------------------------|
| G1 \textsuperscript{1} | 8      | 7   | 0.2105                    |
| G3         | 7      | 11  | 0.1081                    |
| G6         | 8      | 11  | 0.1436                    |
| G7         | 6      | 9   | 0.2074                    |
| G8         | 10     | 9   | 0.1545                    |
| G10        | 7      | 9   | 0.1694                    |
| G11        | 6      | 9   | 0.1176                    |
| G17        | 7      | 10  | 0.1898                    |
| G20        | 5      | 11  | 0.2333                    |

\textsuperscript{1} G1: The 1th satellite.

### 3. GAN-FDSR: GAN-Based Fault Detection and System Reconfiguration Method

#### 3.1. Model Training

The left part of Figure 4 shows the training structure of the model. In this work, the popular variants of the recurrent neural network (RNN) named gated recurrent unit (GRU) networks were employed to build base models of the generator and discriminator to capture temporal correlations of navigation data \cite{38}. Similar to the long short-term memory neural network (LSTM), GRU solves the problems of gradient disappearance, gradient explosion and a lack of long-term memory capacity of RNN, allowing RNN to utilize long-range temporal information effectively. \cite{39}. In addition, GRU models have fewer parameters and less complicated structures compared with LSTM models so it requires less time for model training \cite{40}. The generator takes a random sequence in the latent space as input to generate a pseudo-sequence that is similar to the original sequence. The discriminator takes the original sequence and the pseudo-sequence generated by the generator as input, and judges the authenticity of the data. During training, the generator aims to generate pseudo-sequences that can deceive the discriminator, while the discriminator aims to recognize the pseudo-sequences. The generator and discriminator compete with each other and co-evolve to eventually reach a dynamic equilibrium.
In adversarial training, we consider the sequences of the training set to be normal, while the pseudo-sequences produced by the generator are abnormal. Before the training, the dataset needs to be partitioned into multiple sub-sequences to achieve better training results. We partition \( Y \) into a set of subsequences \( Y_{test} = \{ y_s, s = 1, 2, \cdots, n \} \) using a sliding window of window size \( w \) and step size \( u \), where \( n = \frac{(M-w)}{u} \). In addition, a set of multi-dimensional sequences \( Z = \{ z_s, s = 1, 2, \cdots, n \} \) with the same dimension as \( Y_{test} \) is taken from the random space as input to the generator. The model training strategy is as follows.

- Feed random sequences generated from random space into the generator to generate pseudo-sequences.
  \[ Z = \{ z_s, s = 1, 2, \cdots, n \} \Rightarrow G(Z) \]
- Pseudo-sequences and original sequences are inputted to the discriminator for discriminating.
  \[ Y_{test} = \{ y_s, s = 1, 2, \cdots, n \} \Rightarrow D(Y_{test}) \Rightarrow G(Z) \Rightarrow D(G(Z)) \]
- Optimize the parameters of the discriminator.
  \[ \min \frac{1}{n} \sum_{s=1}^{n} \left[ - \log D(x_s) - \log(1 - D(G(z_s))) \right] \]
- Optimize the parameters of the generator.
  \[ \min \sum_{s=1}^{n} \log(-D(G(z_s))) \]
- Record the parameters of the discriminator and generator in the current iteration after completing the iterative training.

3.2. Detection Function and Detection Threshold

After sufficient iterations of training, the pseudo-sequence is extremely similar to the original sequence, and the discriminator is sufficiently sensitive to anomalous sequences, although it is difficult to distinguish the pseudo-sequence from the original sequence. Therefore, the fault detection function consists of two parts: the generation score and the discrimination score.

Supposing that the system obtains the pseudo-range at the moment \( N + 2 \), we can derive \( r_{N+1} \) according to Equation (12) and reconstruct it in phase space using Equation (16). Eventually a multi-dimensional sequence \( y_{new} \) of \( m \) dimensions and length \( w \) can be obtained, where \( r_N = y_{new}^{r_{N,m}} \).
The discriminator score can be obtained from \( D(y_{\text{new}}) \) directly, but computing the generator score requires the pseudo-sequence that is most similar to the input sequence. In other words, the random sequence that best matches the \( y_{\text{new}} \) needs to be found. First, \( z^1 \) is extracted from the random space and fed into the generator to obtain a pseudo-sequence \( G(z^1) \). Then we use the generation error obtained by calculating the similarity between \( G(z^1) \) and \( z^1 \) as the gradient to update \( z \):

\[
\min_{z^k} \Lambda(y_{\text{new}}, G(z^k)) = 1 - \text{Cov}(y_{\text{new}}, G(z^k)) \tag{20}
\]

After sufficient iterations, the \( z^k \) that makes \( G(z^k) \) most similar to \( y_{\text{new}} \) is the random sequence corresponding to \( y_{\text{new}} \). The generation scores is calculated as follows:

\[
G_s(y_{\text{new}}) = \left| y_{\text{new}} - G(z^k) \right| \tag{21}
\]

The fault detection function for \( y_{\text{new}} \) can be written as:

\[
\text{Score}(y_{\text{new}}) = \lambda G_s(y_{\text{new}}) + (1 - \lambda) D(y_{\text{new}}) \tag{22}
\]

where \( \lambda \) is the weight associated with the training effect of the generator and the discriminator. The performance of the generator and discriminator reaches a dynamic equilibrium after the training is completed. \( \lambda \) can be calculated according to the dispersion degree of the generation and discrimination errors. The specific calculation process is as follows.

1. Calculate the generation error using the pseudo-sequences and the original sequences:

\[
R^G = \{ |y_s - G(z_s)|, s = 1, 2, \ldots, n \} ;
\]

2. \( L^G = \{ l^G_s, s = 1, 2, \ldots, n \} \) is obtained after normalization, where \( l^G_s = \frac{|y_s - G(z_s)| - \min(R^G)}{\max(R^G) - \min(R^G)} \).

3. The dispersion degree of the generation error can be given as:

\[
S^G = \sqrt{\frac{\sum_{s=1}^{n} (l^G_s - \bar{l}^G)^2}{n - 1}} \tag{23}
\]

where \( \bar{l}^G \) is the mean value of the elements in \( L^G \).

The classification accuracy \( \{ c_u, u = 1, 2, \ldots, 2n \} \) is obtained by feeding \( Y_{\text{test}} \) and pseudo-sequences into the trained discriminator, and the discriminant error can be expressed as \( R^D = \{ |c_u - 0.5|, u = 1, 2, \ldots, 2n \} \). Similarly, the dispersion degree of the discriminant error \( S^D \) can be calculated by normalizing and calculating the standard deviation. The weight can be expressed as:

\[
\lambda = \frac{S^D}{S^G + S^D} \tag{24}
\]

If the navigation system is fault-free, then there is a null hypothesis \( H_0: \text{Score} \sim N(\mu_1, \sigma) \), otherwise there is an alternative hypothesis \( H_1: \text{Score} \sim N(\mu_2, \sigma) \). In other words, the mean value of Score is shifted when a fault occurs. We can construct the following detection criteria through the tolerable false alarm rate \( \alpha \):

\[
\begin{cases}
\text{Score} > T_2, \text{ fault occurs} \\
\text{Score} \leq T_2, \text{ no fault occurs}
\end{cases} \tag{25}
\]

where \( T_2 \) is the detection threshold of the false alarm rate \( \alpha \).
3.3. System Reconfiguration

When the observation redundancy is high, isolating fault measurements introduces small positioning errors because the system has high integrated filtering accuracy. However, when the satellite configuration is poor or the number of visible satellites is low, isolating fault observations reduces the available information, which leads to a rapid decrease in filtering accuracy with increasing fault duration, and eventually seriously affects the accuracy and reliability of the whole system.

In practical navigation applications such as UAV and vehicle navigation, the system needs to be able to adjust the reconfiguration scheme dynamically when the satellite observation environment becomes unstable due to factors such as building block and electromagnetic interference. The influence on the integrated filtering is investigated by calculating the RDPOP of the faulty satellite. The formula is as follows:

\[
\text{RDPOP} = \sqrt{\sum_{j=7}^{9} \tilde{P}_{kj}^{ij} - \sum_{j=7}^{9} P_{kj}^{ij}} - \frac{\sum_{j=7}^{9} P_{kj}^{ij}}{\sqrt{\sum_{j=7}^{9} P_{kj}^{ij}}}, \quad i = 1, \ldots, n
\]  

(26)

where \(i\) denotes the dimension of the isolated observation; \(\tilde{P}_{kj}^{ij}\) and \(P_{kj}^{ij}\) represent the \(j\)th element on the diagonal of the covariance matrix \(\tilde{P}_{kj}\) and \(P_{kj}\), respectively; \(\tilde{P}_{kj}\) is the covariance matrix of new state estimation after fault isolation consistent with Equation (A6) in Appendix B; and \(P_{kj}\) is consistent with Equation (9).

RDPOP reflects the decrease in positioning accuracy after isolating a satellite. Therefore, a system reconfiguration scheme can be constructed based on RDPOP as follows:

\[
\begin{cases}
\text{RDPOP} > \eta, \quad \text{measurement reconfiguration} \\
\text{RDPOP} \leq \eta, \quad \text{fault isolation}
\end{cases}
\]

(27)

where \(\eta\) is the detection threshold of RDPOP, which is determined according to the requirements of filtering accuracy generally. The fault measurement is isolated when the RDPOP of the fault satellite is less than \(\eta\). Otherwise, the pseudo-range calculated by \(G(z^k)\) and Equation (12) replace the faulty measurement to integrated filtering.

The specific process of the GAN-based fault detection and system reconfiguration method is as follows.

1. Preprocess raw pseudo-range data.
2. The pre-processed data is fed into a trained GAN model to calculate the detection score. The system is considered faulty and requires system reconfiguration if the score is greater than the detection threshold \(T_2\), otherwise the integrated filtering continues.
3. Calculate the RDPOP value of the faulty satellite. If RDPOP is greater than \(\eta\), use the generated pseudo-range to replace the faulty measurement for integrated filtering; otherwise, isolate the fault measurement.
4. Obtain the navigation solution of the tightly coupled system by correcting the INS output with the error estimation from the integrated filtering.

The architecture of GAN-FDSR is shown in Figure 5.
Figure 5. The architecture of GAN-FDSR.

4. Field Test Results and Analysis

The actual data is collected by a vehicular integrated navigation system which consists of a vehicle, two GNSS antennas, and a POS320 inertial navigation device. The inertial sensor specifications of the POS320 are listed in Table 2. The test equipment is shown in Figure 6.

Table 2. Inertial sensor technical specifications.

| Inertial Sensor   | Bias     | Bias Stability | Measuring Range |
|------------------|----------|----------------|-----------------|
| Gyro             | 0.5°/h   | 0.5°/h         | ±300°/s         |
| Accelerometer    | 25 mg    | 25 mg          | ±10 g           |

Figure 6. Test equipment.
4.1. Analysis of RDPOP

In order to analyze the RDPOP of satellites under different environments and the influence on positioning accuracy, four groups of satellite observation environments provided in Table 3 are set artificially based on the measured data, and the positioning deviation of the faulty satellite is 20 m.

Table 3. Satellite observation environments.

| Fault Section | Faulty Satellite | Number of Visible Satellites | Geometric Configuration |
|---------------|------------------|------------------------------|-------------------------|
| Env 1         | 199–219          | G6                           | 9                       |
| Env 2         | 199–219          | G6                           | 5                       |
| Env 3         | 256–276          | G11                          | 5                       |
| Env 4         | 256–276          | G11                          | 5                       |

The RDPOP of the faulty satellite in each environment is shown in Figure 7, with $\eta$ set to 0.1. The system reconfiguration is performed using the fault isolation (FI) and fault adaptation (FA) described in detail in Appendices B and C, respectively. The positioning errors in each environment during the fault are shown in Figure 8.

In Figure 8, the positioning errors of Env 1 and Env 3 are much smaller than those of Env 2 and Env 4. Although the faulty satellite of Env 1 and Env 2 are the same, the positioning error of Env 2 increases sharply because of the decrease in the number of visible satellites. Env 3 and Env 4 both have five visible satellites, but the positioning error varies greatly. This is because G11 has a greater effect on the positioning accuracy under configuration 2, which leads to a large increase in the positioning error.

Figures 7 and 8 reveal that the faulty satellites with large RDPOP pose enormous challenges for system reconfiguration. FI and FA introduce substantial loss of positioning accuracy in poor observation environments. Therefore, the system reconfiguration scheme should be able to adjust to the changes of the observation environment dynamically according to the weight of the faulty satellite in the filter.

![Figure 7. RDPOP of the faulty satellite in each environment.](image-url)
4.2. Analysis of Fault Detection Performance and Positioning Accuracy

In order to verify the advantages of the method proposed in this paper, a fault is added to the measurement of G7 during 300–360 s. The configuration of the satellites and the RDPOP of G7 are shown in Figure 9. Two additional methods are introduced for comparison. One is the fault detection and isolation method (FDI), and the other is the fault detection and adaptation method (FDA). The fault detection methods for FDI and FDA are given in Appendix A, and their system reconfiguration methods are FI and FA respectively.
The types of faults in the integrated navigation can be roughly divided into large-amplitude faults, small-amplitude faults and gradual faults. Large-amplitude faults are simple to detect due to rapid and large changes, whereas small-amplitude faults and gradual faults are a problem in fault detection due to their small initial amplitude. To test the GAN-FDSR fault detection performance, five fault types are set which contain two step faults and three gradual faults. The specific fault parameters are shown in Table 4. The false alarm rate is 0.001 in the experiment. The detection threshold for FDI and FDA is 2.88, denoted as $T_1$, according to Appendix A. The detection threshold for GAN-FDSR is 1.2, denoted as $T_2$.

**Table 4. Fault types.**

| Fault Type | Failure Mode | Characteristics of Fault |
|------------|--------------|-------------------------|
| Type 1     | Step fault   | 20 m                    |
| Type 2     | Step fault   | 10 m                    |
| Type 3     | Gradual fault| 1 m/s                   |
| Type 4     | Gradual fault| 0.5 m/s                 |
| Type 5     | Gradual fault| 0.3 m/s                 |

Figure 10 shows the fault detection results of the three methods in each environment. Large-amplitude faults lead to rapid changes in the detection function, so all the methods perform well under fault Type 1. It is worth noting that the detection function of FDI and DIA decreases gradually after the fault is identified, which is caused by the filtering error. The detection function changes rapidly due to the large-amplitude fault, so all methods perform well under Type 1. The step fault in Type 2 has a small amplitude which leads to the failure of FDI and FDA to identify the fault, but GAN-FDSR performs well in detecting this fault. In addition, the detection delay occurs in all methods when gradual faults occur, and GAN-FDSR has much smaller detection delays compared with FDI and FDA.

Table 5 lists the detection delay and missing alarm rate of three methods to demonstrate the detection performance. Both FDI and FDA use the residual chi-square test in the experiment, so their detection performance is similar. Benefiting from the sensitivity to small-amplitude faults, GAN-FDSR has lower detection delay and missing alarm rate compared to traditional methods, which brings huge performance advantages.
Figure 10. Fault detection functions for three methods: (a–e) show the fault detection functions under Type 1, Type 2, Type 3, Type 4 and Type 5 faults, respectively.
Table 5. Performance of fault detection.

|                  | Detection Delay | Missing Alarm Rate |
|------------------|-----------------|-------------------|
|                  | FDI  | FDA  | GAN-FDSR | FDI  | FDA  | GAN-FDSR |
| Type 1           | 0    | 0    | 0        | 0    | 0    | 0        |
| Type 2           | 60   | 60   | 0        | 100% | 100% | 0        |
| Type 3           | 18   | 18   | 3        | 30%  | 30%  | 5%       |
| Type 4           | 34   | 34   | 6        | 57%  | 57%  | 10%      |
| Type 5           | 54   | 54   | 9        | 90%  | 90%  | 15%      |

In addition to fault detection performance, positioning accuracy is another critical metric in navigation. In Figure 9, it can be seen that G7 has a large impact on the integrated filtering during a fault, which creates a large troubleshooting challenge. Therefore, improper troubleshooting may lead to a rapid accumulation of positioning errors which brings great risks to users.

Figure 11 gives the positioning errors of the three methods for different fault types, where FF represents fault-free. The positioning errors of FDI and FDA show a cumulative tendency over time except for the step-type positioning errors caused by undetected faults under Type 2. Although FDI and FDA reduce the localization error after the fault is identified, the absence of valid measurements increases the positioning error again over time. In contrast, GAN maintains a lower positioning error with a smaller error accumulation rate under all types of faults. Considering both Figures 9 and 11, we observe that GAN-FDSR adjusts the system reconfiguration strategy in time when the RDPOP exceeds the threshold.

Considering that there will be a certain amount of error even if the system is fault-free, Figure 12 shows the errors of the three methods under each fault type relative to fault-free system to analyze the positioning accuracy quantitatively. Although the fault in Type 1 was identified, the positioning errors of FDI and FDA were the largest among all the types, which demonstrates once more that caution is necessary in handling fault observations when the observation environment is poor. However, the errors of GAN-FDSR are quite close under each type and much lower than the other methods as the integrated system benefits from the timely reconfiguration of the measurements and the small error in the generated data.
Figure 11. Positioning errors for three methods: (a–e) show the positioning errors under Type 1, Type 2, Type 3, Type 4 and Type 5 faults, respectively.
Figure 12. Positioning errors relative to a fault-free system.

5. Conclusions

In this paper, a GAN-based fault detection and system reconfiguration method for a tightly coupled GNSS/INS integrated system is proposed. This method greatly improves the fault detection performance and positioning accuracy, and helps to improve the integrity and reliability of the system. A novel fault detection method is designed based on the generator and discriminator of GAN, which departs from the traditional method that using residuals for fault detection. We investigated the feasibility of chaos theory for the processing of pseudo-range data to improve the learning efficiency of the model for nonlinear characteristics through phase space reconstruction. We also analyzed the relationship between RDPOP and positioning accuracy under different observation environments to develop a dynamic system reconfiguration scheme. Finally, several experiments were conducted to assess fault detection performance and positioning accuracy using measured data. The results show that: (a) The method shows excellent detection performance and significantly reduces detection delays and missing alarm rates for difficult small-amplitude and gradual faults. (b) The positioning error of the system during faults is reduced effectively by dynamically selecting an appropriate system reconfiguration strategy according to different environments.

However, the study presented in this paper is based on the assumption that the INS always works properly, and the possibility of inertial sensor failures needs to be taken into account to improve the method in the future. In addition, the incipient fault diagnosis of the integrated navigation system also deserves further study [41].

Author Contributions: Conceptualization and methodology, Z.S. and X.Z.; software and validation, Z.S. and C.P.; writing—original draft preparation, Z.S.; writing—review and editing, Z.S., L.Z. and C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 41904014, and the Natural Science Basic Research Program of Shaanxi, grant number 2022Jq-645.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
Appendix A. Fault Detection and Identification

The covariance matrix of the $v_k$ can be expressed as:

$$ P_{vk} = H_k P_{k,k-1} H_k^T + R_k $$  \hspace{1cm} (A1) $$

The $v_k$ obeys a Gaussian distribution with zero mean if the signal is fault-free. However, if the measurement vector contains faults, the mean of the $v_k$ will shift. Based on this feature, the fault detection function is constructed as follows:

$$ \lambda_k^i = \frac{c_i^T P_{vk}^{-1} v_k}{\sqrt{c_i^T P_{vk}^{-1} c_i}}, i = 1, \ldots, n $$  \hspace{1cm} (A2) $$

where $c_i$ is the unit vector with the $i$th element equal to one, and $n$ is the dimension of the measurement vector. If the system is fault-free, $\lambda_k^i$ obeys a Gaussian distribution with zero mean, but if there is a fault, the mean is considered to be non-zero. The following detection criteria can be constructed from the tolerable false alarm rate $\lambda$:

$$ \left\{ \begin{array}{ll} \lambda_k^i > N_{a/2}(0,1), & \text{fault occurs} \\ \lambda_k^i \leq N_{a/2}(0,1), & \text{no fault occurs} \end{array} \right. $$  \hspace{1cm} (A3) $$

where $N_{a/2}(0,1)$ is the detection threshold of the false alarm rate $\alpha$, and indicated by $T1$ in the paper.

Appendix B. Fault Isolation

The method avoids disturbance to the system by isolating fault observations. Assuming that the $i$th dimension of the observation vector is faulty, the new measurement vector after fault isolation can be expressed as:

$$ \tilde{Z}_k = \begin{bmatrix} Z_{k}^1 \\ \vdots \\ Z_{k}^{i-1} \\ \tilde{Z}_{k}^{i+1} \\ \vdots \\ Z_{k}^{n} \end{bmatrix} = \begin{bmatrix} H_{k}^1 \\ \vdots \\ H_{k}^{i-1} \\ H_{k}^{i+1} \\ \vdots \\ H_{k}^{n} \end{bmatrix} X_k + \begin{bmatrix} V_{k}^1 \\ \vdots \\ V_{k}^{i-1} \\ V_{k}^{i+1} \\ \vdots \\ V_{k}^{n} \end{bmatrix} = \tilde{H}_k X_k + \tilde{V}_k $$  \hspace{1cm} (A4) $$

where $Z_{k}^{i}$, $H_{k}^{i}$ and $V_{k}^{i}$ are the $i$th row of $Z_k$, $H_k$ and $V_k$. The new state estimation and its covariance matrix can be expressed as [42]:

$$ \tilde{P}_{ki} = (I - \tilde{K}_k \tilde{H}_k) P_{k,k-1} $$

$$ = P_{k,k-1} - P_{k,k-1} \tilde{H}_k^T (\tilde{K}_k + \tilde{H}_k P_{k,k-1} \tilde{H}_k^T)^{-1} \tilde{H}_k P_{k,k-1} $$  \hspace{1cm} (A5) $$

$$ \tilde{X}_k = \tilde{X}_{k,k-1} + \tilde{K}_k (\tilde{Z}_k - \tilde{H}_k \tilde{X}_{k,k-1}) $$

$$ = P_{k,k-1} \tilde{H}_k^T (\tilde{K}_k + \tilde{H}_k P_{k,k-1} \tilde{H}_k^T)^{-1} \tilde{Z}_k + (I - P_{k,k-1} \tilde{H}_k^T (\tilde{K}_k + \tilde{H}_k P_{k,k-1} \tilde{H}_k^T)^{-1} \tilde{H}_k) \tilde{X}_{k,k-1} $$

$$ = (P_{k,k-1}^{-1} + \tilde{H}_k^T \tilde{K}_k^{-1} \tilde{H}_k)^{-1} (P_{k,k-1}^{-1} \tilde{X}_{k,k-1} + \tilde{H}_k^T \tilde{K}_k^{-1} \tilde{Z}_k) $$

$$ = \tilde{P}_{ki} (P_{k,k-1}^{-1} \tilde{X}_{k,k-1} + \tilde{H}_k^T \tilde{K}_k^{-1} \tilde{Z}_k) $$  \hspace{1cm} (A6) $$

where $\tilde{K}_k$ is the noise covariance matrix of the fault-free measurements.
Appendix C. Fault Adaptation

FA uses the optimal estimate of the error to correct the state estimation to reduce the effect of faults on the combined filtering. Assuming that the $i$th dimension of the observation vector is faulty, the measurement equation can be expressed as:

$$Z_k = H_k X_k + V_k + e_i \nabla$$

where $e_i$ is the unit vector with the $i$th element equal to one; and $\nabla$ is the error of the $i$th dimensional measurement.

The optimal estimation of the error is given as:

$$\nabla = (e_i^T P_{vk}^{-1} e_i)^{-1} e_i^T P_{vk}^{-1} v_k$$

The covariance matrix of $\nabla$ can be written as:

$$P_\nabla = (e_i^T P_{vk}^{-1} e_i)^{-1}$$

The new measurement equation after fault adaptation can be obtained from the equation:

$$\tilde{Z} = Z_k - e_i \nabla = H_k X_k + \tilde{\xi}_k$$

where $\tilde{\xi}_k$ is the new measurement noise with a covariance matrix $\tilde{R}_k$ which can be written approximately as $R_k + e_i P_\nabla e_i^T$ after adaptation. Then, the new state estimation and its covariance matrix can be expressed as:

$$\tilde{X}_k = P_k^{-1} (H_k)^T (H_k \tilde{R}_k^{-1} H_k + e_i P_\nabla e_i^T)^{-1} \tilde{Z}$$

$$\tilde{P}_k^{-1} = P_k^{-1} + H_k^T \tilde{R}_k^{-1} H_k$$

References

1. Wang, C.; Xu, A.; Sui, X.; Hao, Y.; Shi, Z.; Chen, Z. A seamless navigation system and applications for autonomous vehicles using a tightly coupled GNSS/UB/INS/MAP integration scheme. Remote Sens. 2022, 14, 27. [CrossRef]

2. Sun, R.; Zhang, W.; Zheng, J.; Ochieng, W.Y. GNSS/INS Integration with Integrity Monitoring for UAV No-fly Zone Management. Remote Sens. 2020, 12, 524. [CrossRef]

3. Gu, S.; Dai, C.; Fang, W.; Wang, Y.; Zhang, Q.; Lou, Y.; Niu, X. Multi-GNSS PPP/INS tightly coupled integration with atmospheric augmentation and its application in urban vehicle navigation. J. Geod. 2021, 95, 64. [CrossRef]

4. LaForest, L.; Hasheminasab, S.M.; Zhou, T.; Flatt, J.E.; Habib, A. New strategies for time delay estimation during system calibration for UAV-based GNSS/INS-assisted imaging systems. Remote Sens. 2019, 11, 1811. [CrossRef]

5. Stepanov, O.A. Optimal and Suboptimal Filtering in Integrated Navigation Systems. In Aerospace Navigation Systems; John Wiley Sons, Ltd.: Hoboken, NJ, USA, 2016; Chapter 8; pp. 244–298. [CrossRef]

6. Zhao, Y. Performance evaluation of Cubature Kalman filter in a GPS/IMU tightly-coupled navigation system. Signal Process. Off. Publ. Eur. Assoc. Signal Process. 2016, 119, 67–79. [CrossRef]

7. Hu, G.; Gao, S.; Zhong, Y. A derivative UKF for tightly coupled INS/GPS integrated navigation. ISA Trans. 2015, 56, 135–144. [CrossRef]

8. Gao, D.; Hu, B.; Qin, F.; Chang, L.; Xu, L. A real time gravity compensation method for INS based on BPNN. IEEE Sens. J. 2021, 21, 13584–13593. [CrossRef]

9. Shaheen, E.M. Mathematical Analysis for the GPS Carrier Tracking Loop Phase Jitter in Presence of Different Types of Interference Signals. Gyroscopy Navig. 2018, 9, 267–276. [CrossRef]

10. Liu, W.; Gu, M.; Mou, M.; Hu, Y.; Wang, S. A distributed GNSS/INS integrated navigation system in a weak signal environment. Meas. Sci. Technol. 2021, 32, 115108. [CrossRef]

11. Gautier, J.D.; Parkinson, B.W. Using the GPS/INS generalized evaluation tool (GIET) for the comparison of loosely coupled, tightly coupled and ultra-tightly coupled integrated navigation systems. In Proceedings of the 59th Annual Meeting of the Institute of Navigation and CIGTF 22nd Guidance Test Symposium (2003), Albuquerque, NM, USA, 23–25 June 2003; pp. 65–76.

12. Zhang, C.; Zhao, X.; Pang, C.; Zhang, L.; Feng, B. The influence of satellite configuration and fault duration time on the performance of fault detection in GNSS/INS integration. Sensors 2019, 19, 2147. [CrossRef]

13. Sukkarieh, S.; Nebot, E.M.; Durrant-Whyte, H.F. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Trans. Robot. Autom. 1999, 15, 572–578. [CrossRef]
14. Gao, G.; Gao, S.; Hong, G.; Peng, X.; Yu, T. A robust INS/SRS/CNS integrated navigation system with the chi-square test-based robust Kalman filter. Sensors 2020, 20, 5909. [CrossRef]
15. Liu, Y.; Li, S.; Fu, Q.; Liu, Z.; Zhou, Q. Analysis of Kalman filter innovation based GNSS spoofing detection method for INS/GNSS integrated navigation system. IEEE Sens. J. 2019, 19, 5167–5178. [CrossRef]
16. Zhong, L.; Liu, J.; Li, R.; Wang, R. Approach for detecting soft faults in GPS/INS integrated navigation based on LS-SVM and AIME. J. Navig. 2017, 70, 561–579. [CrossRef]
17. Yang, L.; Shen, Y.; Li, B.; Rizos, C. Simplified algebraic estimation for the quality control of DIA estimator. J. Geod. 2021, 95, 14. [CrossRef]
18. Zaminpardaz, S.; Teunissen, P.J.G. DIA-datsnooping and identifiability. J. Geod. 2019, 93, 85–101. [CrossRef]
19. Crespiillo, O.G.; Medina, D.; Skaloud, J.; Meurer, M. Tightly coupled GNSS/INS integration based on robust M-estimators. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS) (2018), Monterey, CA, USA, 23–26 April 2018; pp. 1554–1561. [CrossRef]
20. Zhang, F.; Wang, Y.; Gao, Y. A novel method of fault detection and identification in a tightly coupled, INS/GNSS-integrated system. Sensors 2021, 21, 2922. [CrossRef]
21. Wang, S.; Zhan, X.; Zhai, Y.; Liu, B. Fault detection and exclusion for tightly coupled GNSS/INS system considering fault in state prediction. Sensors 2020, 20, 590. [CrossRef]
22. Liu, F.; Sun, X.; Xiong, Y.; Huang, H.; Guo, X.; Zhang, Y.; Shen, C. Combination of iterated cubature Kalman filter and neural networks for GPS/INS during GPS outages. Rev. Sci. Instrum. 2019, 90, 125005. [CrossRef]
23. Zhu, Y.; Cheng, X.; Wang, L. A novel fault detection method for an integrated navigation system using Gaussian process regression. J. Navig. 2016, 69, 905–919. [CrossRef]
24. Sun, R.; Cheng, Q.; Wang, G.; Ochieng, W.Y. A novel online data-driven algorithm for detecting UAV navigation sensor faults. Sensors 2017, 17, 2243. [CrossRef]
25. Guo, C.; Li, F.; Tian, Z.; Guo, W.; Tan, S. Intelligent active fault-tolerant system for multi-source integrated navigation system based on deep neural network. Neural Comput. Appl. 2020, 32, 16857–16874. [CrossRef]
26. Peng, D.; Yang, W.; Liu, C.; Lü, S. SAM-GAN: Self-Attention supporting Multi-stage Generative Adversarial Networks for text-to-image synthesis. Neural Netw. 2021, 138, 57–67. [CrossRef]
27. Mosser, L.; Dubrule, O.; Blunt, M.J. Stochastic reconstruction of an ololitic limestone by Generative Adversarial Networks. Transp. Porous Med. 2018, 125, 81–103. [CrossRef]
28. Li, R.; Wu, C.; Liu, S.; Wang, J.; Wang, G.; Liu, G.; Zeng, B. SDP-GAN: Salient detail preservation Generative Adversarial Networks for high perceptual quality style transfer. IEEE Trans. Image Process. 2021, 30, 374–385. [CrossRef]
29. Liang, J.; Tang, W. Sequence Generative Adversarial Networks for wind power scenario generation. IEEE J. Sel. Areas Commun. 2020, 38, 110–118. [CrossRef]
30. Li, D.; Chen, D.; Shi, L.; Jin, B.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In Proceedings of the International Conference on Artificial Neural Networks (2019), Munich, Germany, 17–19 September 2019. [CrossRef]
31. Thomas, S.; Philipp, S.; Sebastian, M.W.; Georg, L.; Ursula, S.E. f-AnoGAN: Fast unsupervised anomaly detection with Generative Adversarial Networks. Med. Image Anal. 2019, 56, 30–44. [CrossRef]
32. Takens, F. Detecting strange attractors in turbulence. In Dynamical Systems and Turbulence, Warwick; Rand, D., Young, L.S., Eds.; Springer: Berlin/Heidelberg, Germany, 1980; Volume 898. [CrossRef]
33. Xu, Y.; Huang, Y.; Xu, X.; Xiao, F. Improved hybrid model for predicting concrete crack openings based on chaos theory. Math. Probl. Eng. 2022, 2022, 5174774. [CrossRef]
34. Wei, N.; Yin, L.; Li, C.; Liu, J.; Li, C.; Huang, Y.; Zeng, F. Data complexity of daily natural gas consumption: Measurement and impact on forecasting performance. Energy 2022, 238, 122090. [CrossRef]
35. Wang, X.; Li, W.; Yin, C.; Zeng, S.; Liu, P. A multiple-parameter approach for short-term traffic flow prediction. Mod. Phys. Lett. B 2021, 35, 2150245. [CrossRef]
36. Fraser, A.M.; Swinney, H.L. Independent coordinates for strange attractors from mutual information. Phys. Rev. A Gen Phys. 1986, 33, 1134–1140. [CrossRef][PubMed]
37. Cao, L. Practical method for determining the minimum embedding dimension of a scalar time series. Phys. D 1997, 110, 43–50. [CrossRef]
38. Huang, Y.; Han, X.; Zhao, L. Recurrent neural networks for complicated seismic dynamic response prediction of a slope system. Eng. Geol. 2021, 289, 106198. [CrossRef]
39. Jalayer, A.; Kahanis, M.; Pourmasoumi, A.; Beheshti, A. HAM-Net: Predictive business process monitoring with a hierarchical attention mechanism. Knowl.-Based Syst. 2022, 236, 107722. [CrossRef]
40. Gao, S.; Huang, Y.; Zhang, S.; Han, J. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. J. Hydrool. 2020, 589, 125188. [CrossRef]
41. Wu, Y.; Jiang, B.; Wang, Y. Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains. ISA Trans. 2020, 99, 488–495. [CrossRef]
42. Zhang, C.; Zhao, X.; Pang, C.; Li, T.; Zhang, L. Adaptive fault isolation and system reconfiguration method for GNSS/INS integration. IEEE Access 2020, 8, 17121–17133. [CrossRef]