

RESEARCH

Open Access



NIDD: an intelligent network intrusion detection model for nursing homes

Feng Zhou¹, Xin Du¹, Wenli Li^{2*}, Zhihui Lu^{1*} and Jie Wu¹

Abstract

In nursing homes using technologies such as IoT, big data, cloud computing, and machine learning, there is a constant risk of attacks such as Brute Force FTP, Brute Force SSH, Web Attack, Infiltration, and Botnet during data communication between individual terminals and the cloud server. Therefore, effectively identifying network communication data is essential to protect data communication security between individual terminals and the cloud server. Aiming at the data mentioned above regarding communication security issues, we propose an intelligent intrusion detection model NIDD (Network Intelligent Data Detection) model that combines deep convolution generation adversarial network (DCGAN) with Light Gradient Boosting Machine (LightGBM) and Shapley Additive exPlanations (SHAP). The NIDD model first generates new attack samples by learning the feature distribution of the existing attack sample data and effectively expands the rare attack samples. Secondly, we use the Light Gradient Boosting Machine (LightGBM) algorithm as the base classifier to train the dataset and start to build the intrusion detection model. Then use Shapley Additive exPlanations (SHAP) to analyze the contribution of the classification results, and adjust the model parameters according to the analysis results. Finally, we obtain the optimal model for the intelligent detection model of network intrusion. This paper conducts experimental tests on the NSL-KDD dataset. The experimental results show that the NIDD model built based on Light Gradient Boosting Machine can detect Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, PROBE, R2L, and U2R attacks with an accuracy of 99.76%. Finally, we re-verified the NIDD model on the CIC-IDC-2018 dataset. The results once again proved that the NIDD model could solve the data communication security between the nursing robot and the cloud server and the data before the IoT terminal and the cloud server. Communication security provides a sufficient guarantee.

Keywords: Generative adversarial networks, Intrusion detection, Artificial Intelligence, Internet of things, Cloud Computing, Big Data

Introduction

The aging population makes nursing homes that already lack nursing staff even worse. To solve the problem of a lack of professionals, nursing homes will carry out informatization and intelligent construction. Nursing robots will be involved in the nursing service part of the renovated nursing home. The five dimensions of electronic health record construction, health dynamic monitoring, health analysis and evaluation, active health intervention,

and continuous health tracking constitute a comprehensive and effective health management service.

In the process of these services, the nursing robot needs to communicate with the cloud server before providing related nursing services for the elderly. The sphygmomanometer needs to transmit data to the nursing terminal through the Internet of Things, and the nursing terminal transmits the data to the cloud server. Health analysis and evaluation require accurate analysis and evaluation of the current user's health status only after computing the data uploaded by devices such as sleep pads, oximeters, blood lipid meters, uric acid meters, and blood glucose meters through the network. Nursing homes increasingly

*Correspondence: lwlszpt@163.com; lzhu@fudan.edu.cn

¹ Fudan University, Shanghai, China

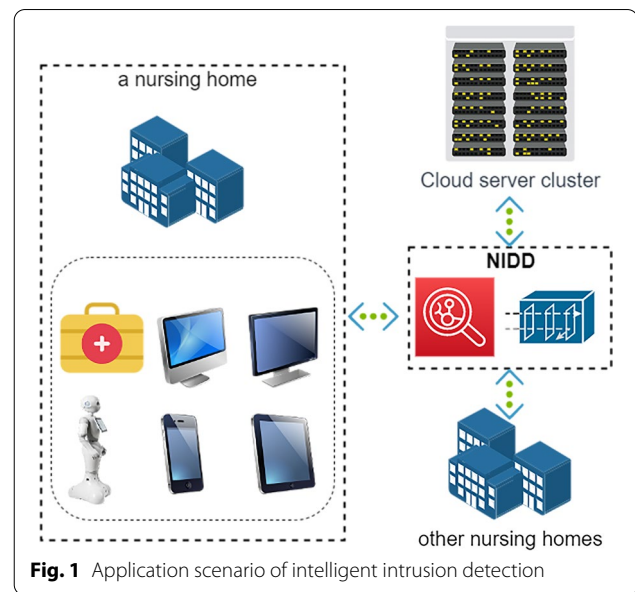
² Shenzhen Polytechnic, Shenzhen, China

depend on various information systems, intelligent systems, and IoT terminal devices. Therefore, it is crucial to ensure the security of data communication between the nursing robot and the cloud server and the security of data communication between the IoT terminal and the cloud server. Intrusion detection ensures the security of network communication between these systems, equipment, and equipment, and systems and equipment have become essential to protecting individual information security in nursing homes.

Network intrusion detection is to identify abnormal attack information in regular network traffic. In order to reduce the leakage of related data caused by network intrusion, network intrusion detection has become a standard active defense method in current network security technology. With the rapid development of machine learning technology in recent years, domestic and international scholars have conducted much research on network intrusion detection and identification based on machine learning. However, in the actual research process of regular network traffic anomaly detection, the data for normal traffic is much larger than the data for abnormal traffic, and the proportion of various types of abnormal traffic after classification is not uniform—essential reasons for poor accuracy and performance.

According to our research, the primary data for the security protection of individual information in nursing homes are data about electronic health records, daily physiological detection, daily sleep, daily service monitoring, IoT device, and daily operation management.

Figure 1 is an application scenario of intelligent network intrusion detection applied to nursing homes. We take a nursing home that has undergone informatization and intelligent transformation to illustrate the application scenario of intelligent intrusion detection. For the elderly living in the nursing home, the nursing staff will use the tablet computer to create an electronic health record for the elderly. The nursing-end application will transmit the data to the cloud server through the network during this process. Nursing staff use sphygmomanometers, oximeters, blood lipid meters, uric acid meters, blood glucose meters, and other equipment to conduct health checks for the elderly. During this process, these IoT terminals transmit data to the nursing-end application through Bluetooth and then transmit the data to the nursing-end application through the network. When the nursing staff provides nursing services for the elderly, the nursing robot will cooperate with the nursing staff to provide relevant services for the elderly according to the task instructions sent by the received cloud server. The family-end application obtains the health status and nursing



service data of the elderly in real-time through data communication with the cloud server.

Since the individual information of the nursing home is stored in the cloud server, to realize the security protection of the individual information of the nursing home, it is necessary to detect and identify the network communication traffic data between each terminal and the cloud server, as well as other network communication traffic data requesting the cloud server. To realize the need to protect individual information security in nursing homes, we used the NSL_KDD dataset and the CIC-IDC-2018 dataset as training samples. Finally, a network intrusion detection model NIDD (Network Intelligent Data Detection) is designed based on a deep convolution generation adversarial network (DCGAN) and based on Light Gradient Boosting Machine (LightGBM) and Shapley Additive exPlanations (SHAP). Generative adversarial networks (GAN) are composed of two neural networks: a generative network and a discriminant network. The generative network repeatedly learns the distribution of actual samples and finally achieves the purpose of generating fake samples with high authenticity, thereby enhancing and expanding the dataset. DCGAN combines convolutional neural networks (CNN) and GAN to ensure the quality of the generated sample data and the diversity of samples. The relevant features are trained based on the Light Gradient Boosting Machine (LightGBM). Finally, we analyze the contribution of each feature to the classification result using Shapley Additive exPlanations (SHAP), and parameters are adjusted to obtain the best model. Experiments show that the model not only effectively

solves the problems of low intrusion detection recognition accuracy, high false alarm rate, and limited recognition types caused by sparse sample data and unbalanced sample data types. Moreover, the model has specific improvements in sample training efficiency, method execution efficiency, method interpretability, and method robustness.

The goal of this paper is that in the application scenario of information security in nursing homes, the network intrusion intelligent detection model can effectively identify attacks in network communication. The training samples used in this paper are the NSL-KDD dataset and the CIC-IDC-2018 dataset. Firstly, the problem of data sample imbalance is solved based on a deep convolution generation adversarial network (DCGAN). Then based on Light Gradient Boosting Machine (LightGBM), an intrusion detection model of network communication traffic data is constructed. Secondly, the contribution of each feature to the classification results is analyzed using Shapley Additive exPlanations (SHAP). Thirdly, parameter optimization is performed. Finally, the protection of personal information security in nursing homes is realized.

This paper has five chapters in total. The first chapter is the introduction, which mainly describes the current research background and status of network intrusion detection and summarizes the main content of this paper as a whole. The second chapter is about model design, which mainly describes the process of designing a network intrusion intelligent detection model applied to nursing homes. The third chapter is the realization of the model, which mainly describes the realization process of a network intrusion intelligent detection model applied to nursing homes. The fourth chapter analyzes the results, which mainly expresses the comparative analysis of the experimental results and the analysis of the contribution value of the characteristic variables that affect the experimental results. The fifth chapter is the conclusion, which mainly expresses the research results of this paper and the prospect for the future.

Related work

Much research has been devoted to solving network intrusion detection in recent years. Since the data for regular traffic is much larger than for abnormal traffic, the proportion of various types of abnormal traffic after classification is uneven. As a result, the network intrusion detection and identification model based on machine learning has low efficiency, performance, and accuracy. To solve these problems, researchers have mainly conducted research based on machine learning, deep neural networks, and generative adversarial networks.

Biswapriyo Chakrabarty [1] et al. proposed a comprehensive intrusion detection algorithm based on K-Centroid clustering and genetics. The model trained by the algorithm using Kd99Cup and NSLKDD datasets has a reasonable detection rate. Saras Saraswathi [2] et al. proposed a global iterative optimization algorithm that combines integer-coded genetic algorithm (ICGA) and particle swarm optimization (PSO) with neural network-based extreme learning machines (ELM) for classification tasks. The algorithm has the characteristics of easy implementation and fast operation speed. It has been widely used in parameter optimization and other work and has shown promising results. Prabhat Kumar [3] et al. proposed a novel intrusion detection system that combines K-Nearest Neighbors, XGBoost, Gaussian Naive Bayes, and Random Forest. Maryam Samadi Bona [4] et al. proposed a new hybrid method based on Fruit Fly Algorithm (FFA) and Ant Lion Optimization (ALO). The simulation results of this method on the datasets KDD Cup99, NSLKDD, and UNSW-NB15 have a sure accuracy. Based on the UNSW-NB15 dataset, Vikash Kumar [5] et al. proposed a novel intrusion detection system approach. The method achieved [6] an MFM of 84.5%, ADR of 90.32, and FAR of 2.01%. Arushi Agarwal [7] et al. built a model based on the UNASW-NB15 dataset using Naive Bayes (NB), Support Vector Machine (SVM), and K-nearest neighbor (KNN).

S. H. Khan [8] et al. proposed a cost-based deep neural network to automatically learn robust feature representations for both majority and minority samples. The computational overhead of the training process of this method is low. However, when the sample information of a small number of types is oversampled due to the lack of diversity of sample data, it is prone to overfitting during training [9]. When under-sampling is performed, the property information of the sample data will be lost, thereby reducing the ability to solve the imbalance of sample data types. Guang-Bin Huang [10] et al. proposed Extreme Learning Machine (ELM). Since the algorithm is superior to traditional machine learning algorithms, support vector machines (SVM), and neural networks regarding learning speed and generalization performance, it has been widely used in network intrusion detection in recent years. Steven Z Lin [11] et al. proposed a new character-level intrusion detection model based on convolutional neural networks. The model performs better in terms of high accuracy, high detection rate, and low false positive rate. Junjiao Liu [12] et al. proposed a novel two-level anomaly detection framework based on a convolutional neural network (CNN). The framework identifies normal and abnormal from existing mixed data and learns normal behavior from average data. Mohammad Shurman [13] et al. proposed

two methods to detect Distributed Reflection Denial of Service (DDoS) attacks in IoT. Nevertheless, it can only detect Distributed Reflection Denial of Service (DDoS) attacks in IoT. Sydney Mambwe Kasongo [14] et al. proposed a Feed-Forward Deep Neural Network (FFDNN) method for wireless intrusion detection systems based on the Wrapper Based Feature Extraction Unit (WFEU). For AWID, the method achieves an overall accuracy of 99.66% and 99.77% for binary and multi-class classification configurations, respectively. Erxue Min [15] et al. proposed a novel intrusion detection system named TR-IDS based on word embedding and a text convolutional neural network (Text-CNN). The system employs manually designed features and extracts salient features from the payload to improve intrusion detection performance.

Xiao Wang [16] et al. proposed a new attack framework model called AT-GAN. AT-GAN can efficiently generate diverse and realistic unconstrained adversarial examples. AT-GAN can learn a distribution of adversarial examples very close to the accurate data distribution. Weiwei Hu [17] et al. proposed a generative adversarial network (GAN)-based algorithm named MalGAN to generate adversarial malware examples. This example can bypass black-box machine learning-based detection models. Muhammad Usama [18] et al. proposed an adversarial attack method based on a black-box machine learning/deep learning intrusion detection system and a generative adversarial network (GAN). The method successfully avoids intrusion detection while retaining network traffic's typical functional behavior characteristics. Milad Salem [19] et al. proposed a new intrusion detection method based on Cycle-GAN. The experimental results show that the AUC increases from 0.55 to 0.71, and the percentage of detected unseen anomalies increases from 17.07–80.49%. The AI-enabled IoT-CPS algorithm proposed by Lakshmana Kumar Ramasamy [20] et al. has achieved excellent classification and patient disease prediction results. It is

some do not effectively detect the packet header information of network traffic data in the process of intrusion detection, and some are in the process of implementation. The model does not use the network traffic data of malicious attacks, and some models identify fewer types of attacks. Some models are inherently flawed, such as extreme learning machines (ELMs) that only support single-hidden layer feedforward neural networks. Convolutional Neural Network (CNN) models are less interpretable in the feature extraction process. K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) algorithms are computationally inefficient in datasets with large sample sizes. The feedforward deep neural network (FFDNN) model is complex, the process is not interpretable, and its performance is low under small data samples. In short, the existing research cannot meet the needs of the current intelligent network intrusion detection in nursing homes. The NIDD model we propose can be well applied to the current application scenarios of intelligent network intrusion detection in nursing homes.

Model design

This section describes the base model used in the model design process, the specific process used to construct the model, and the evaluation metrics used.

Generative adversarial nets

Generative Adversarial Nets (GAN) is a new framework for estimating generative models through an adversarial process proposed by Ian J. Goodfellow [22] et al. Generative Adversarial Nets (GANs) are described as a minimax game. The discriminative model D tries to minimize its reward $V(D, G)$, the generative model G tries to maximize the loss of the discriminative model D , and the mathematical formula for minimizing the maximum is described in the following formula (1).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

believed that the algorithm can be more widely used if it is supplemented with features for data protection. Using the KDDcup99 dataset, Edeh Michael Onyema [21] et al. proposed a Cyborg Intelligence intrusion detection framework based on AdaBoost ensemble learning, which achieved good accuracy, detection rate, and low false positive rate.

In the existing research, some models do not use any samples with label information in the training process,

In formula (1), G represents the generative model, D represents the discriminant model, $P_{data}(x)$ represents the real sample distribution, P_z represents the generated sample distribution, x represents the sample of $P_{data}(x)$, z represents a random vector, $D(x)$ represents x is the probability of the real sample, $G(z)$ represents the sample generated by the generative model G based on the input z , and the cost function $V(D, G)$ represents the alternate minimization and maximization of

$D(x)$ and $G(z)$. Maximizing the probability of assigning to training samples and generating correct labels for new samples is achieved by training the discriminative model D . The $\log(1 - D(G(z)))$ minimization is achieved by training the generative model G . By making $D(x)$ infinitely close to 1 and $D(G(z))$ infinitely close to 0, the optimal solution $\min G$ of the generative model G is obtained.

The main steps in the construction of a deep convolution generation adversarial network (DCGAN) [23] are as follows: We replace all pooling layers with stridden convolutions in the discriminative model and all pooling layers with fractionally stridden convolutions in generative models. Batch normalization (BN) is introduced in both generative and discriminative models. Then, we remove the fully connected layer. The output layer of the generative model uses the Tanh activation function, and all other layers use the ReLU activation function. All layers in the discriminative model use the LeakyReLU activation function.

The optimization of deep convolution generation adversarial network (DCGAN) on Generative Adversarial Nets (GAN) mainly includes: Using stridden convolutions instead of sampling layers and convolutions instead of fully connected layers. Each layer in the generative model G and the discriminative model D introduces batch normalization (BN). All layers in discriminative model D use the LeakyReLU activation function. ReLU is still used in the generative model G , but the Tanh activation function is used in the output layer of the generative model G . We train the model with Adam optimizer.

Classification base model

The XGBoost algorithm is an integrated machine learning algorithm with solid fault tolerance, controllable complexity, and fast parallel speed proposed by Chen [24] et al. Due to its high prediction accuracy and outstanding execution efficiency, this algorithm has recently been widely used in Kaggle machine-learning competitions. The XGBoost algorithm is an improved version of the gradient-boosting decision tree algorithm. Gradient Boosting Decision Tree is a widely used decision tree-based ensemble machine learning algorithm that performs very well in data analysis, multi-classification, and prediction. The two algorithms are similar in that they are both composed of decision trees for classification and regression; the difference is that XGBoost reduces the complexity and overfitting of the model by using a standard penalty term in the objective function. In addition, the loss function of the [25] GBDT algorithm uses a first-order Taylor expansion, and XGBoost uses a second-order Taylor expansion.

The implementation process of the XGBoost algorithm is as follows: after combining multiple decision tree models into an integrated learner, the second-order Taylor expansion is used, and the regular term is added to the loss function to control the model complexity and overfitting. Its predicted output value is the sum of the predicted output values of each tree pair sample in the K trees, as shown in the formula (2) function:

$$\hat{y}_i = \Phi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (2)$$

In formula (2), \hat{y}_i represents the predicted output value of the model, x_i represents the i th sample, and \mathcal{F} represents the space of the Classification and Regression Tree (CART), and its function representation is shown in formula (3):

$$\mathcal{F} = \{f(x) = \omega_{q(x)}\} \left(q : \mathbb{R}^m \rightarrow T, \omega \in \mathbb{R}^T \right) \quad (3)$$

In formula (3), the mapping model of the tree is represented by q (that is, the weight of each training sample to the corresponding leaf node), the weight set of all leaf nodes is represented by $\omega_{q(x)}$, and the number of leaf nodes is represented by T . The approximate objective function using the second-order Taylor expansion is shown in formula (4).

$$\mathcal{L}^t \simeq \sum_{i=1}^n \left[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (4)$$

In formula (4), g_i represents the loss gradient of the i th sample output in the first order, and h_i represents the loss gradient of the i th sample output in the second order. Define $I_j = \{i | q(x_i) = j\}$ as the set of samples that fall into the leaf node j . Combining the above, the final objective function can be obtained as shown in formula (5).

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Upsilon T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \Upsilon T \end{aligned} \quad (5)$$

The base model of the XGBoost [25] algorithm is a binary tree. Adding leaf nodes is achieved by splitting. For each split, the split points on all features are first traversed. Second, calculate the objective function gain of all samples split according to the split point one by

one. Finally, the split point with the most significant gain is selected from the split points, and the split point with the most significant gain is split to generate a leaf node. Such an algorithm first needs to save the eigenvalues of the training samples and the result of feature sorting. Secondly, the objective function gain calculation is required when traversing the split points. Then, since the access of the features to the gradient is random, the access order of different features is also different. Therefore, this algorithm consumes a lot of space and time, and the Cache cannot be optimized.

The LightGBM [26] algorithm is based on the XGBoost algorithm and uses the Histogram algorithm to reduce the number of split points. Use the GOSS algorithm to reduce the number of samples. Reduce the number of features using the EFB algorithm. Suppose there are n sample training sets $\{x_1, \dots, x_n\}$, each x_i is an s -dimensional vector in the X^s space, and the loss function gradient is marked as $\{g_1, \dots, g_n\}$. In the GOSS algorithm, the samples are firstly arranged in descending order according to the absolute value of the gradient, $a \times 100\%$ large gradient instances are reserved as subset A , and the remaining $(1 - a) \times 100\%$ small gradient instances are subset A^C . A subset B of size $b \times |A^C|$ is used, and finally the samples are divided according to the variance gain $\tilde{V}_j(d)$ calculated on $A \cup B$, as shown in Eq. (6).

$$\tilde{V}_j(d) = \frac{1}{n} \left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{n} \sum_{x_i \in B_l} g_i \right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{n} \sum_{x_i \in B_r} g_i \right)^2}{n_r^j(d)} \right) \quad (6)$$

In formula (6), $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$. The LightGBM algorithm dramatically reduces the complexity of the newly added leaf nodes by using the three algorithms of Histogram, GOSS, and EFB. In addition, the Histogram algorithm converts the features from floating point numbers to integers of 0~255 bits for storage. Therefore, LightGBM algorithm training consumes less time and less memory space than XGBoost algorithm training.

Shapley additive explanations

In order to obtain the contribution of the characteristic variables in the sample to the prediction results, we use the machine learning interpretation framework SHAP developed by Lundberg [27] et al. to calculate the contribution value of the feature variable, that is, the SHAP value. The calculation formula is shown in formula (7).

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (7)$$

In formula (7), ϕ_i represents the contribution of the i th feature, s represents the feature subset, $F \setminus \{i\}$ represents the feature set, F represents the total number of input features, $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ represents the predicted output value of the model when the sample has only the eigenvalues in $x_{S \cup \{i\}}$, and $f_S(x_S)$ represents the predicted output value of the model when the sample has only the eigenvalues in x_S . The linear function $f(x)$ used by the additive model when describing regression is shown in Eq. (8).

$$f(x) = g(x) = \phi_0 + \sum_{i=1}^M \phi_i' x_i \quad (8)$$

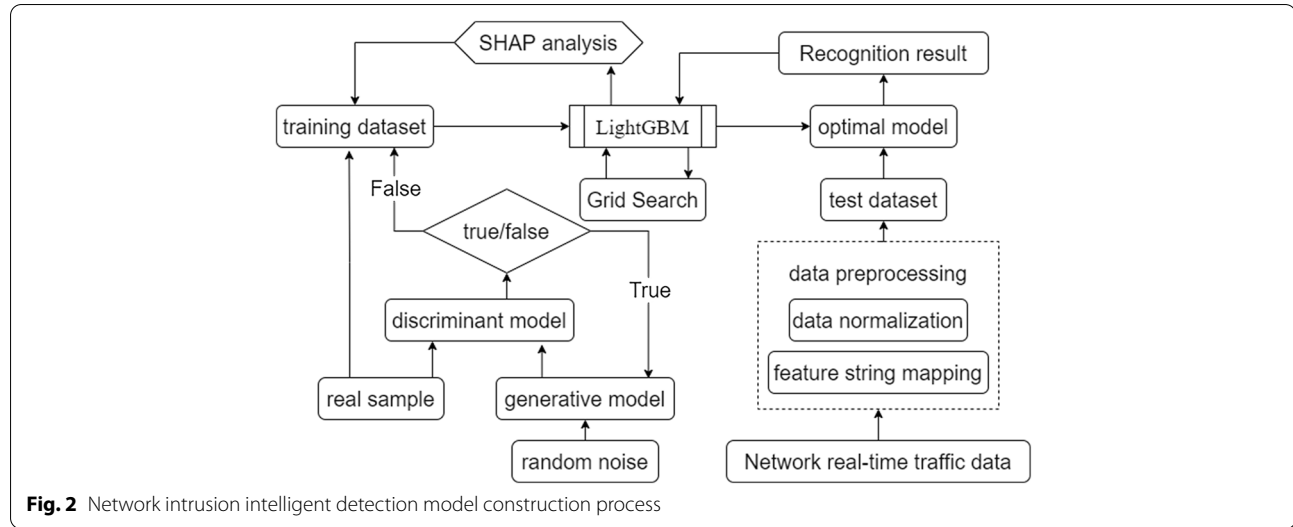
In formula (8), $f(x)$ represents the interpretation of the predicted value of sample x in the model. ϕ_0 represents the mean of the model's predicted output values, and x_i represents the i th feature sample.

NIDD model building

The network intrusion intelligent detection model proposed in this paper's NIDD (Network Intelligent Data Detection) model construction process is shown in Fig. 2.

The detailed process of the network intrusion intelligence detection model-building process is as follows:

- The generative and discriminative models of the Deep Convolutional Generative Adversarial Network (DCGAN) are trained alternatively using the NSL-KDD dataset.
- A mixture of authentic samples and samples generated based on the DCGAN algorithm was used as the training dataset to train the models constructed based on LightGBM.
- The resultant contribution values were analyzed using Shapley Additive exPlanations (SHAP), the training samples were optimized, and the training dataset was updated based on the analysis results.
- The updated training dataset is used to set the range of parameters of the model, load Grid Search, train the constructed NIDD model, and update the model



parameters based on the scores of each model parameter value.

- Real-time traffic data is listened to and captured, and the captured data is feature mapped, digitized, normalized, and manually annotated to form a test dataset.
- The optimal model is tested using the test dataset, and based on the test results, the model is again optimized.
- Iterations were repeated until the model accuracy reached a pre-set threshold of 99.5%, Precision reached a pre-set point of 92%, Recall reached a pre-set threshold of 92%, False Positive Rate (FPR) achieved a pre-set threshold of 100%, and F1-Measure reached a predetermined point of 92%.

The core algorithmic process of the NIDD model consists of two main parts.

```

Input:
P(z): random noise
{x1, x2, ..., xn}: the real sample
Output:
D: training dataset
1: for i = 0 → epochs do:
2:   Sampling batch samples at random from random noise P(z) to get {z1, z2, ..., zn}
3:   The generator generates the sample data P(g(z))
4:   for j = 0 → k do:
5:     randomly sample batch samples {x1, x2, ..., xn} = {g(z1), g(z2), ..., g(zn)} from the
6:     generated sample P(g(z))
7:     A random sample of batch samples from the real sample gives
8:     {x1, x2, ..., xn}
9:     X += {x1, x2, ..., xn} + {x1, x2, ..., xn}
10:   End for
11: End for
12: Pre-processing the dataset X
13: divide the dataset X into 5 equal parts, 1 for testing and 4 for training, to obtain the
14: training dataset D
15: Output the training dataset D

```

Algorithm 1: dcgan model training

Algorithm 1 is DCGAN MODEL TRAINING, based on DCGAN, to solve the data sample balancing problem.

In Algorithm 1, is the ratio of the discriminator to generator training. Here the is 1. the is the number of times the DCGAN algorithm works, the is the number of samples, and the is the dataset.

```

Input:
D: training dataset
T: test dataset
Output:
NIDD model:
1: for k in 1: epochs_N do:
2:   Train the constructed NIDD model and load the SHAP explanatory model
3:   Adjust the feature variables according to the contribution values of the SHAP analysis
4:   Update the training dataset D
5:   Initially, set the model for each parameter range
6:   Load the base model, load Grid Search
7:   Train the constructed NIDD model
8:   Update the model parameters based on the scores of each model parameter value
9:   Perform accuracy using test dataset T
10:  if Accuracy >= Acc and Precision >= Pre and Recall >= Rec and False Positive
11:    Rate >= FPR and F1 values >= F1 :
12:    End of training
13:  End if
14: End for
15: Output NIDD model

```

Alorithm 2: Lightgbm model training

In Algorithm 2, the *batch* is the number of samples, the *epochs_N* is the number of model-building exercises, and the *Acc* is the accuracy pre-set threshold. Here the *Acc* is 99.5%, the *Pre* is the pre-set threshold for accuracy, the *Pre* is 92%, the *Rec* is the pre-set threshold for the recall rate, and here the *Rec* is 92%. The *FPR* is the false positive rate pre-set threshold. Moreover, here, the *FPR* is 100%. The *F1* is the pre-set threshold for the F1 value, and here the *F1* is 92%.

Evaluation metrics

This paper uses five typical indicators to evaluate the performance of the network intrusion intelligent detection model. The five evaluation indicators are Accuracy, Precision, Recall, False Positive Rate (FPR), and F1-Measure.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (11)$$

$$FPR = \frac{FP}{FP + TN} \times 100\% \quad (12)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \times 100\% \quad (13)$$

Among them, TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives, as shown in the confusion matrix in Table 1.

In this section, we build a NIDD model based on evaluation indicators such as deep convolution generation adversarial network (DCGAN), Light Gradient Boosting Machine (LightGBM), Shapley Additive exPlanations (SHAP), and Accuracy, and evaluate related algorithms, frameworks, and models. The construction process is explained.

Model implementation

In this section, we focus on the specific process of model implementation.

Training a deep Convolutional Generative Adversarial Network (DCGAN)

Generative model training is performed by concatenating the generative model with the discriminative model. We define the goal of generative model training to be able to generate fake samples that are difficult for discriminative models to distinguish. During the training process, we use the false sample set generated by the generative model as the input layer of the discriminant model, and the probability value of the sample being a trustworthy and false sample is the output layer. The target value of

the generative model is set to 1, and the maximum possible true and false samples are generated. Analytical loss values are calculated using the cross-entropy loss function.

The input layer of the discriminant model is composed of authentic samples and false sample sets generated by the generative model. The discriminant sample's probability value is the discriminant model's output layer. The goal of discriminative model training is to correctly identify true and false samples to be considered a binary classification task and the actual sample outputs 1. Otherwise, it outputs 0. Then use the activation function to convert the probability value into the sample predicted classification label. Finally, the analysis loss value is calculated using the cross-entropy loss function.

When the loss values and number of iterations do not reach the pre-determined values, the generative and discriminant models continue to be trained alternately until the loss values and number of iterations reach the pre-determined values. The training is stopped, and the optimization gradient is updated using the Adam optimizer. When the loss value and the number of iterations reach the pre-defined values, the generated sample set is added to the training sample set to obtain the training sample set.

Feature preprocessing

The training set in this paper uses a diverse sample set consisting of the NSL_KDD dataset and the generated sample set. The main steps of processing are as follows:

- Delete rows with missing attribute values, convert text to numeric values, and encode character features and categorical labels with OrdinalEncoder.
- The mixed sample set is used as the training sample set, and the network real-time traffic data with data preprocessing is used as the test set.

LightGBM training and SHAP contribution analysis

This paper uses the Light Gradient Boosting Machine (LightGBM) for classifier construction. The classifier analyzes the loss value between the sample and the predicted result until it reaches a preset value. Contribution analysis was performed using Shapley Additive exPlanations (SHAP). The main steps are as follows:

- Define the LightGBM model, and run the model.
- Get model SHAP_VALUES.
- Feature parsing for single-sample and multi-sample.
- The samples were analyzed globally using two feature ranking methods, discriminative eigenvalues, and indistinguishable eigenvalues.

Table 1 Confusion matrix

Sample type	Predicted as a normal sample	Predicted as an attack sample
Normal sample	TN	FP
Attack sample	FN	TP

- Analyze the dependencies between the various characteristic variables of the sample.
- Analyze SHAP_VALUES after sample feature variable interaction.

Parameter optimization

Parameter optimization is a way to prevent model overfitting and improve model performance by controlling the complexity of the model. In this paper, five-fold cross-validation is used to obtain the optimal parameters of the model. Five-fold cross-validation means dividing the data into five equal parts, taking one copy of each experiment for testing, using the rest for training, and averaging the data after five experiments. In training, the grid search method (GridSearchCV) is used to find the optimal parameters for the basic parameters of the model, the parameters that affect the accuracy, the parameters that affect overfitting, and the parameters that affect the training speed. Finally, the optimal parameter prediction model based on the LightGBM algorithm is obtained after several iterative tests and parameter adjustments.

In this section, we carry out the process of deep convolution generation adversarial network (DCGAN) to solve the sample balance problem, data preprocessing, Light Gradient Boosting Machine (LightGBM) training, Shapley Additive exPlanations (SHAP) model contribution value analysis, NIDD model parameter optimization and other processes. We have a detailed analysis and explanation.

Experimental results and analysis

In this section, we focus on the experimental dataset used in training the NIDD model of intrusion intelligence detection applied to nursing homes, prediction results, analysis of the degree of influence of features, analysis of the contribution of prediction results, and model optimization and validation.

Experimental dataset

The NSL_KDD dataset. The NSL_KDD dataset is an improved version after removing many redundant data from the KDD CUP99 dataset. The dataset consists of four subsets: KDDTest+, KDDTest-21, KDDTrain+, KDDTrain+_20Percent. The KDDTest-21 and KDDTrain+_20Percent in the NSL_KDD dataset archive are subsets of KDDTrain+ and KDDTest+. There are 125,973 training data and 11,850 test data in the NSL-KDD dataset. Attack types include Denial of Service (Dos), Port Scanning and Probe (Probe), unauthorized access to local superusers (U2R), and unauthorized access to remote machines. There are four types of access

Table 2 Dataset data distribution

Attack Name	Quantity	Attack Type
Warezmater	40,287	R2L
Warezcclient	113,079	R2L
Teardrop	30,742	Dos
Spy	21,445	R2L
Smurf	48,042	DoS
Satan	91,893	Probe
Rootkit	65,086	U2R
Portsweep	85,694	Probe
Pod	96,906	Dos
Phf	117,608	R2L
Perl	69,147	U2R
Normal	72,910	
Nmap	3168	Probe
Neptune	112,950	Dos
Multihop	102,107	R2L
Loadmodule	113,266	U2R
Land	52,662	DoS
Ipsweep	51,080	Probe
Imap	12,291	R2L
Guess_passwd	71,108	R2L
Ftp_write	2294	R2L
Buffer_overflow	69,476	U2R
Back	52,211	DoS

(R2L), and Normal is the normal access data. The data distribution of the NSL_KDD dataset is shown in Table 2.

The NSL_KDD dataset provides features such as protocol name, service content name, host, and time. The NSL_KDD dataset contains 23 labels. Table 2 shows that the data volume of each category label is unbalanced if the deep convolution generation adversarial network (DCGAN) is not used to solve the problem of unbalanced sample data. This dataset will mislead the classifier, and the classification results will be skewed towards a large number of categories.

The CIC_IDC_2018 dataset. The CIC_IDC_2018 dataset contains the latest common attack scenarios: Brute-force attack, Heartbleed attack, Botnet, Denial-of-Service, Distributed Denial of Service, Web Attacks, and Infiltration of the network from inside. The file distribution of the CIC_IDC_2018 dataset is shown in Table 3.

This dataset is not a real-world attack traffic dataset. The data set uses machine learning technology to statistically analyze the network behavior of ordinary users and simulate regular user traffic. Use the above attack types to simulate anomalous traffic scenarios and configuration files to generate a comprehensive anomaly detection traffic dataset. The CICIDS-2018 dataset is used to capture the natural flow of the model using the tool

Table 3 Document content

File Name	Attack Name
Thursday-01-03-2018	Benign, Infiltration
Friday-02-03-2018	Benign, Bot
Wednesday-14-02-2018	Benign, SSH-Bruteforce, FTP-BruteForce
Thursday-15-02-2018	Benign, DoS-GoldenEye, DoS-Slowloris
Friday-16-02-2018	Benign, DoS attack-hulk, DoS attacks-SlowHTTPTest
Thursday-20-02-2018	Benign, DDoS attacks-LOIC-HTTP, DDoS-LOIC-UDP
Wednesday-21-02-2018	Benign, DDOS-LOIC-UDP, DDOS-HOIC
Thursday-22-02-2018	Benign, Brute, Force-Web, Brute, Force-XSS, SQL Injection
Friday-23-02-2018	Benign, Brute, Force-Web, burtel Force-XSS, SQL Injection
Wednesday-28-02-2018	Benign, Infiltration

CICFlowMeter-V3. From the statistical analysis perspective, 80 features were extracted, 79 independent variables. The specific data distribution of the CICIDS-2018 dataset is shown in Table 4.

The CICIDS-2018 dataset contains 15 labels. Table 4 shows that the amount of data for each category label is unbalanced if the deep convolution generation adversarial network (DCGAN) is not used to solve the problem of sample data imbalance. This dataset also misleads the classifier, causing the classification results to move closer to a more significant number of categories.

Forecast result

Standard classification prediction algorithms are K Nearest Neighbors, support vector machines, Logistic

Regression, Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine. The NIDD model proposed in this paper is constructed based on K Nearest Neighbors, support vector machines, Logistic Regression, Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine, respectively, during the construction process. Furthermore, the evaluation results are compared and verified, as shown in Fig. 3a.

We calculated the base model's time cost based on K Nearest Neighbors, support vector machines, Logistic Regression, Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine. We sum up the construction time of several base models and calculate the proportion of the construction time of several base models to the total construction time of several base models, as shown in Fig. 3b.

As seen from Fig. 3a, the accuracy rates of the base models based on K Nearest Neighbors, support vector machines, Logistic Regression, and Light Gradient Boosting Machine all exceed 95%, and they all have high detection capabilities. Among them, based on the Light Gradient Boosting Machine, the base model constructed by the algorithm has the highest accuracy, and the detection accuracy reaches 99.91%.

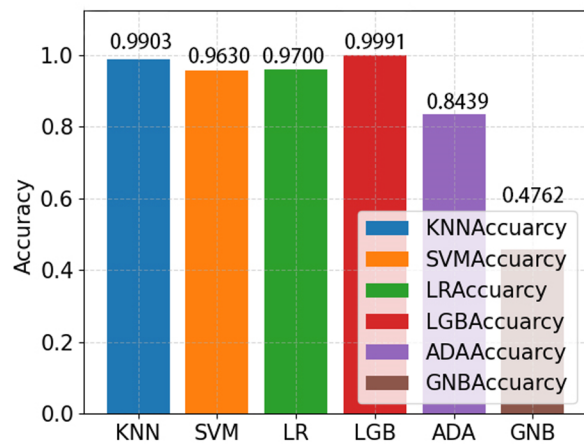
Figure 3b shows that the time spent on building the base model based on K Nearest Neighbors, support vector machines, and Logistic Regression is relatively long. It takes less time to build the base model based on Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine. Among them, the construction cost of the base model based on the Light Gradient Boosting Machine is the least. Experiments show that constructing the base model based on the Light Gradient Boosting Machine is the most efficient.

In the analysis of the experimental results of the NIDD model proposed in this paper, the indicators we use are Precision, Recall, False Positive Rate, and F1-Measure. We compare the results based on K Nearest Neighbors, support vector machines, Logistic Regression, Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine. The Precision comparison is shown in Fig. 4a. The recall comparison is shown in Fig. 4b.

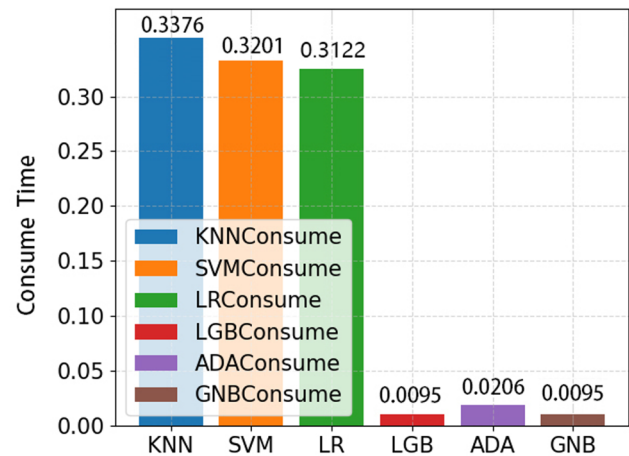
It can be seen from Fig. 4a that the accuracy rates of the base models based on K Nearest Neighbors, support vector machines, Logistic Regression, and Light Gradient Boosting Machine all exceed 80%, and they all have high detection capabilities. Among them, the base model based on the Light Gradient Boosting Machine algorithm has the highest accuracy rate, and the detection accuracy rate reaches 99.99%. It can be seen from Fig. 4b that the recall rates of the base models based on K Nearest Neighbors, support vector machines, Logistic Regression, and Light Gradient Boosting Machine all exceed 85%, and

Table 4 Dataset data distribution

Attack Name	Quantity
SSH-Bruteforce	274,497
SQL Injection	2497
Infiltration	577,996
FTP-BruteForce	124,116
DoS attacks-Slowloris	31,641
DoS attacks-SlowHTTPTest	44,943
DoS attacks-Hulk	333,820
DoS attacks-GoldenEye	11,259
DDoS attacks-LOIC-HTTP	523,392
DDOS attack-LOIC-UDP	956
DDOS attack-HOIC	251,902
Brute Force -XSS	1060
Brute Force -Web	389
Bot	287,098
Benign	117,909

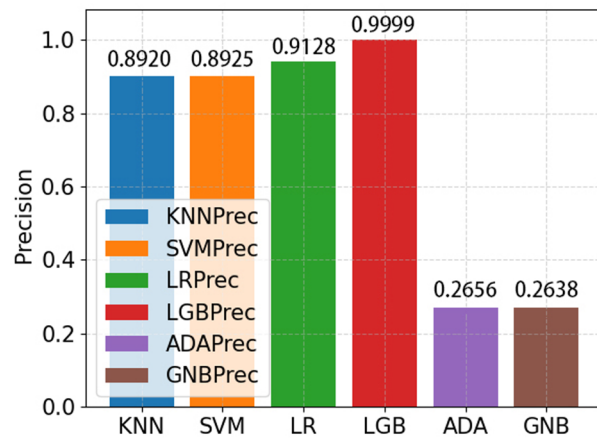


**a. Comparison of Accuracy
Evaluation Results**

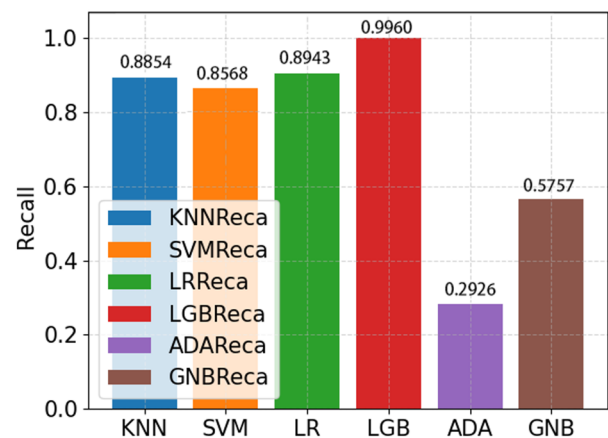


b. Time cost comparison

Fig. 3 Accuracy and time overhead comparison



**a. Comparison of
Precision Evaluation Results**



**b. Comparison of Recall
evaluation results**

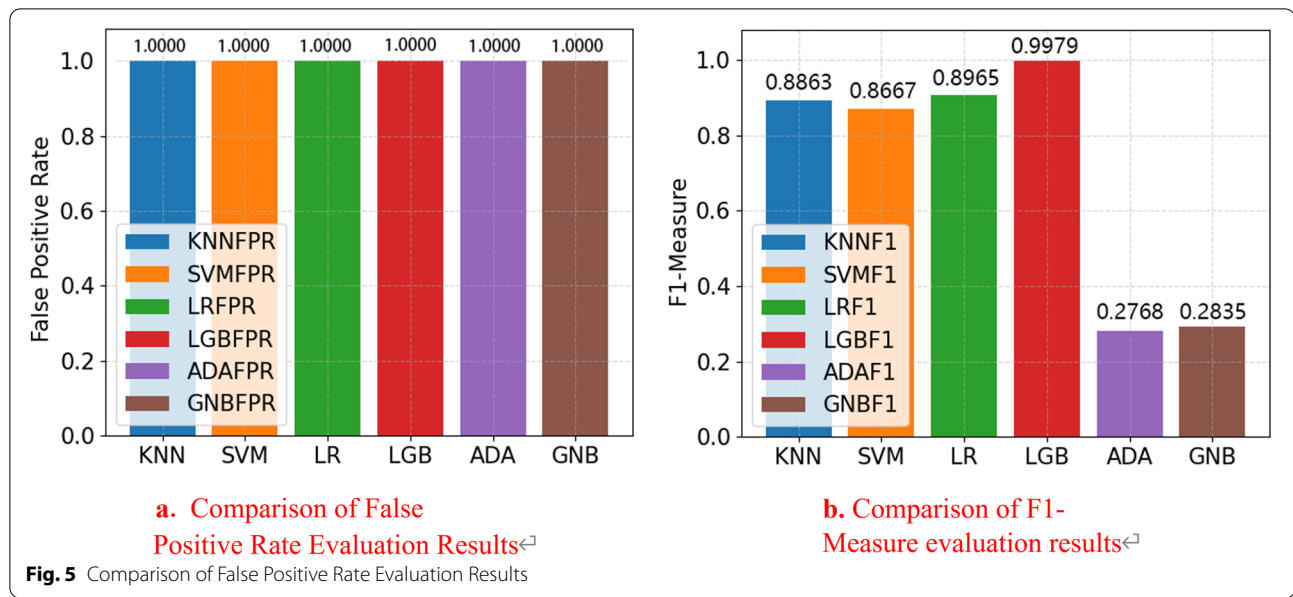
Fig. 4 Precision and recall comparison

they all have high detection capabilities. Among them, the base model based on the Light Gradient Boosting Machine algorithm has the highest recall rate, and the recall rate of detection reaches 99.60%. The False Positive Rate comparison is shown in Fig. 5a. The F1-Measure comparison is shown in Fig. 5b.

From Fig. 5a, we can see that the False Positive Rate of the base model, which is based on K Nearest Neighbors, support vector machines, Logistic Regression, Adaptive Boosting, Naive Bayes model, and Light Gradient Boosting Machine are all achieved outstanding values. It can be seen from Fig. 5b that the recall rates of the base models

based on K Nearest Neighbors, support vector machines, Logistic Regression, and Light Gradient Boosting Machine all exceed 85%, and they all have high detection capabilities. Among them, the base model based on the Light Gradient Boosting Machine algorithm has the highest recall rate, and the recall rate of detection reaches 99.79%.

The contribution of this paper is to innovatively apply the algorithm to the application scenario of intelligent network intrusion detection in nursing homes. From the above experimental results, it can be concluded that the NIDD model proposed in this paper is constructed based



on the Light Gradient Boosting Machine algorithm, and the Accuracy, Precision, Recall, False Positive Rate, and F1-Measure of intrusion detection are the highest.

Feature influence degree analysis

For the intrusion detection studied in this paper, the correlation between the independent and dependent variables is studied in Fig. 6.

In Fig. 6, the horizontal axis represents the name of the characteristic independent variable, and the vertical axis represents the influence weight value on the characteristic dependent variable.

It can be seen from Fig. 6 that the feature variables have a more significant impact on the results of network intrusion detection. The importance of the feature independent variable in the decision tree, the LightGBM algorithm uses the weight index to calculate the importance ranking of each feature variable. The importance ranking of some feature variables in this paper is shown in Fig. 7.

In Fig. 7, the horizontal axis represents the name of the characteristic independent variable, and the vertical axis represents the influence weight value on the characteristic dependent variable. Figure 7 shows some of the characteristic independent variables that ultimately impact the NIDD network intrusion detection model proposed in this paper.

Prediction result contribution analysis

In order to deeply analyze the factors that affect the intrusion detection results, we use the interpretable machine learning SHAP algorithm to analyze the SHAP value of the global feature independent variable, the importance

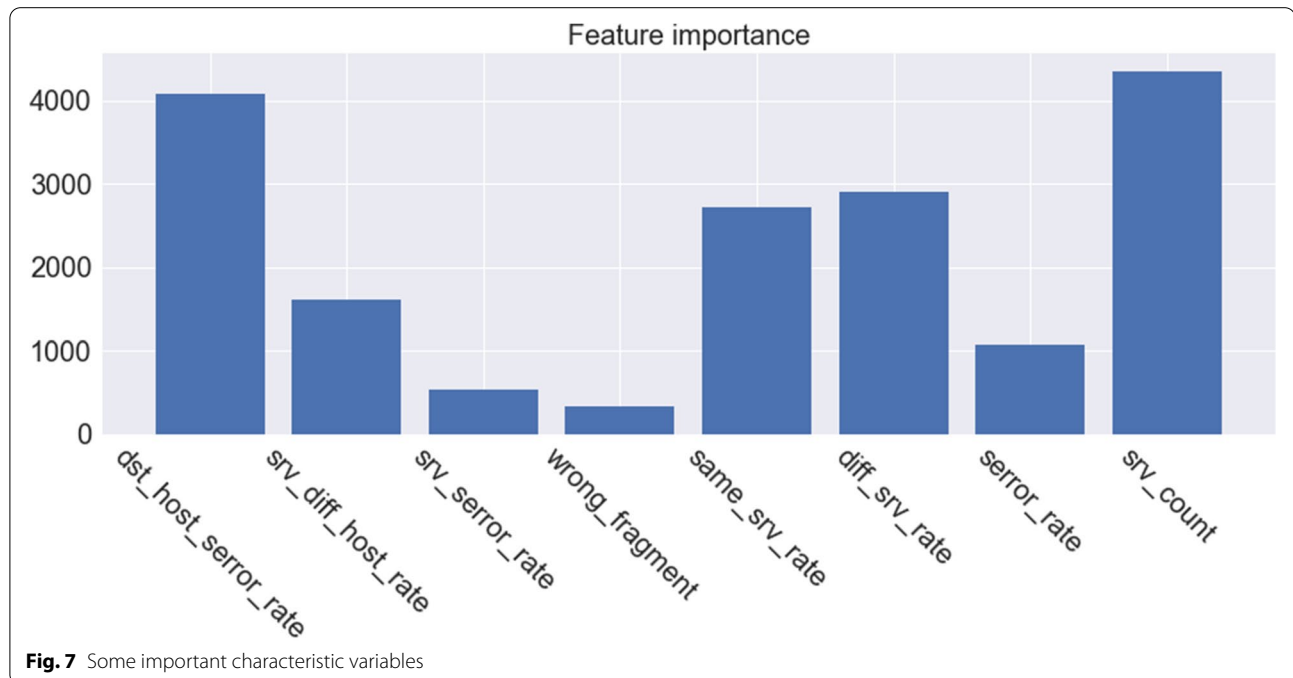
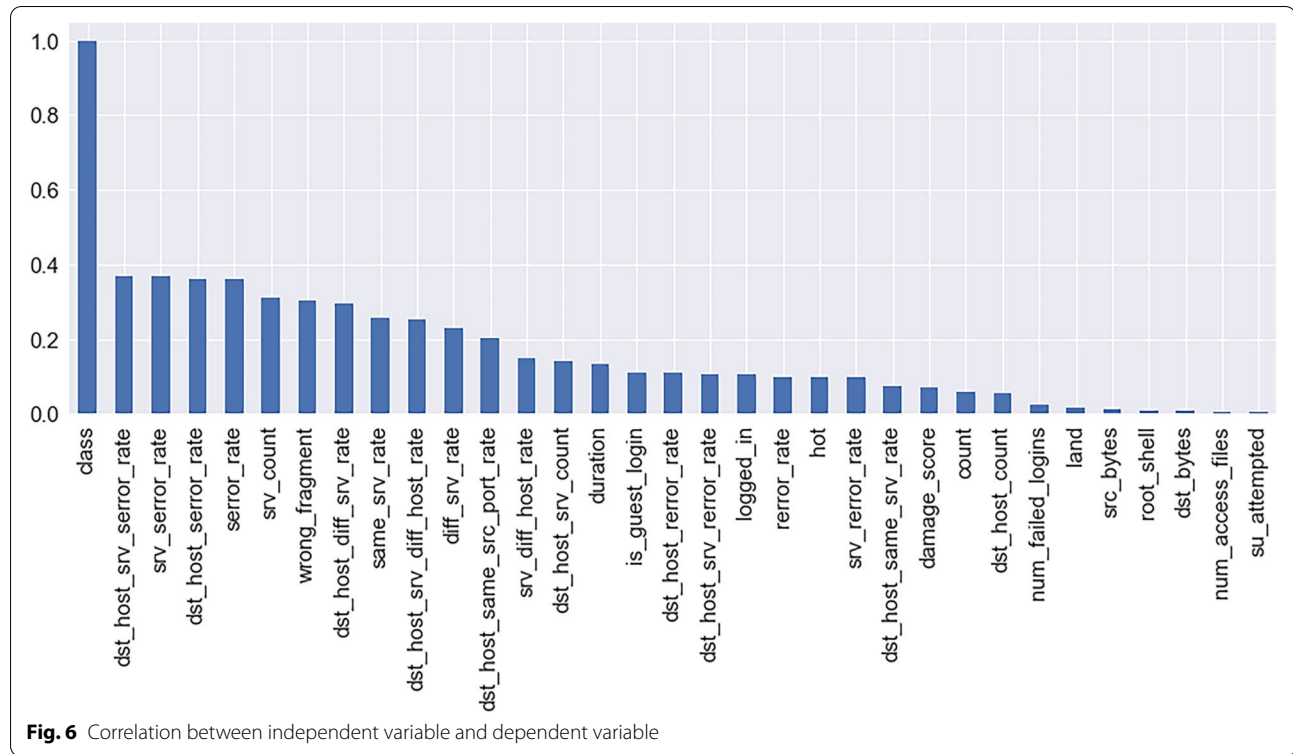
of the feature variable to the predicted result, and the interaction between some feature-independent variables and the feature-independent variables. In terms of SHAP value and other aspects, the contribution of intrusion detection results is analyzed in detail. The influence of global feature-independent variables on the prediction results is shown in Fig. 8. Each row in the figure represents a feature-independent variable, the abscissa is the SHAP value, and a point represents a sample. The bluer the color, the smaller the SHAP value of the feature independent variable. The redder the color, the larger the value of the independent variable SHAP of the feature.

From Fig. 8, it can be seen that the characteristic independent variables with higher SHAP values.

Model optimization and validation

Model optimization mainly includes sample optimization and parameter optimization. In terms of sample optimization, according to the conclusions of the previous section, this paper firstly optimizes and adjusts the sample and extracts the characteristic independent variables and dependent variables that directly impact the prediction results of the NIDD model to form a new sample. The five evaluation indicators of Accuracy, Precision, Recall, False Positive Rate (FPR), and F1-Measure are also used to evaluate the intrusion detection results of the NIDD model trained by the samples before and after optimization.

According to the above parameter optimization ideas, this paper uses GridSearchCV to find the optimal parameters and finally obtains the optimal parameters after multiple iterative tests and parameter adjustment. We use the five evaluation indicators of Accuracy,



Precision, Recall, False Positive Rate, and F1-Measure to evaluate the intrusion detection of the NIDD model before and after parameter optimization.

From Fig. 9a, we can see that the NIDD model trained with the samples after tuning has a particular

improvement in Precision, Recall, and F1-Measure of intrusion detection than the NIDD model trained with the samples after tuning. After sample optimization, the precision rate (Precision) is increased by 14%, the recall rate (Recall) is increased by 10%, and the F1

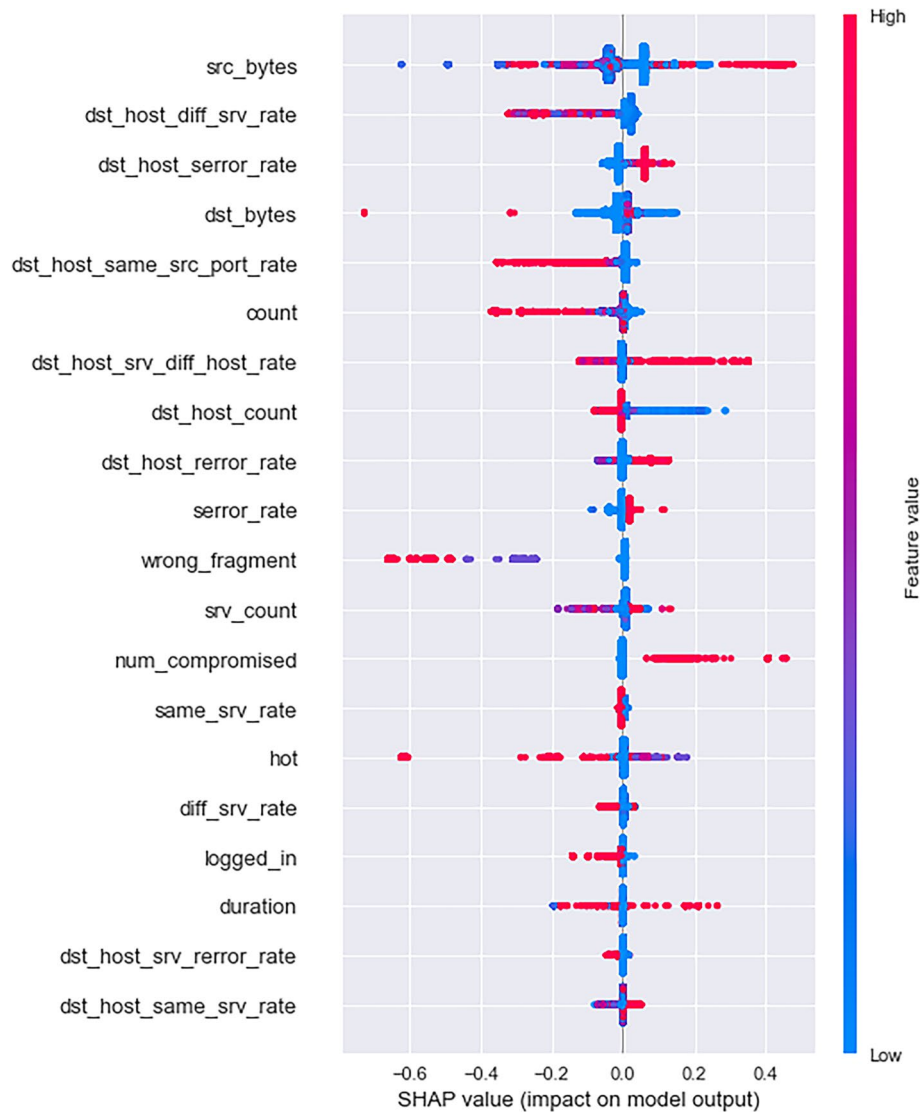


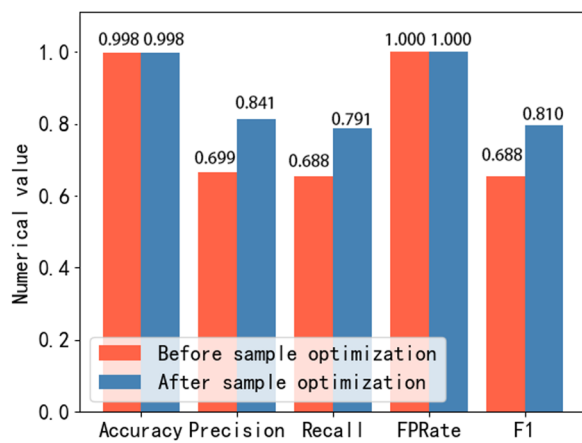
Fig. 8 Global feature independent variable SHAP value

value (F1-Measure) is increased by 12%. Figure 9b shows that the NIDD model trained after parameter tuning improved Accuracy, Precision, Recall, and F1-Measure compared to the NIDD model trained before parameter tuning. It is because the default value of the parameters in the base model before parameter tuning is not a value that has a high degree of control over the model fitting ability, model preventing overfitting, and model training accuracy. After parameter optimization, the accuracy is increased by 67%, the precision by 66%, the recall by 64%, and the F1-Measure by 67%.

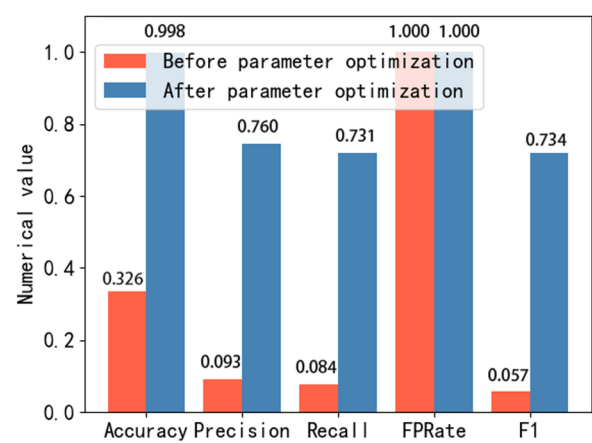
In order to verify again, the NIDD model optimization (sample optimization and parameter optimization) on the intrusion detection results of the NIDD model is proposed. We also use the five evaluation

indicators of Accuracy, Precision, Recall, False Positive Rate, and F1-Measure to evaluate the intrusion detection results of the NIDD model trained before and after the sample and parameter optimization. In order to further verify the performance of the NIDD model, We use the CIC_IDC_2018 dataset as training samples, and the experimental results are shown in Fig. 10b.

Figure 10a shows that the trained NIDD model has dramatically improved the Accuracy, Precision, Recall, and F1-Measure of intrusion detection after tuning the samples and parameters. The accuracy rate (Accuracy) after sample and parameter tuning is increased by more than 40%, the precision rate (Precision) is increased by 60%, the recall rate (Recall) is increased by 61%, and the F1 value

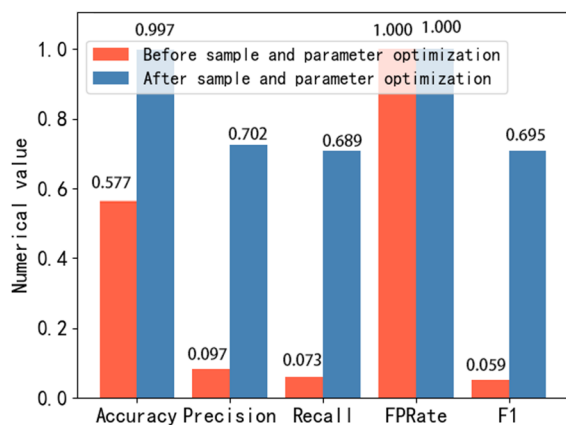


a. The evaluation results before and after sample tuning

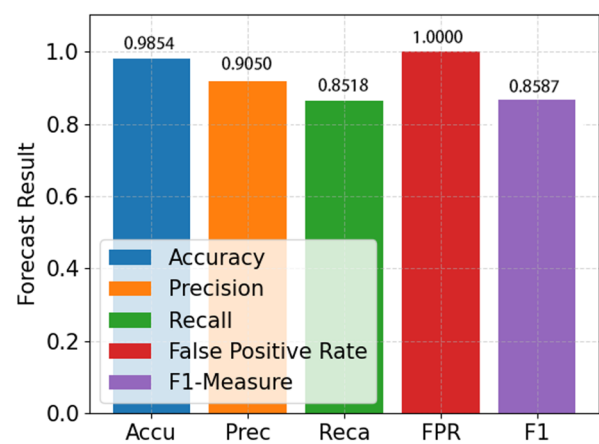


b. The evaluation results before and after parameter tuning

Fig. 9 Comparison of samples and parameters before and after adjustment



a. The evaluation results before and after sample and parameter tuning



b. Accuracy, Precision, Recall, False Positive Rate (FPR), and F1-Measure of the NIDD model on the CIC_IDC_2018 dataset

Fig. 10 Comparison of samples with parameters before and after adjustment and validation on the CIC_IDC_2018 dataset

(F1-Measure) is improved. 63%. Figure 10b shows that the NIDD model proposed in this paper is on the CIC_IDC_2018 data set, Accuracy, Precision, Recall, else Positive Rate, and F1-Measure on five evaluation indicators performance is still very good. The experimental results again prove that the NIDD model can provide a sufficient guarantee for the data communication security between the nursing robot and the cloud server and the data communication security between the IoT terminal and the cloud server.

Conclusion

In this paper, we construct the NIDD model, an intelligent intrusion detection model applied to nursing homes, based on the DCGAN and LightGBM algorithms and train the model using the NSL_KDD dataset, and finally use the interpretable machine learning SHAP algorithm to analyze the contribution of the intrusion detection results meticulously. After iterative testing and parameter tuning, the NIDD model achieved an accuracy of 99.7% for intrusion detection. In the CIC_IDC_2018 dataset,

the accuracy of the NIDD model for intrusion detection reached 98.5%, achieving practical support for the network security of nursing homes. Nevertheless, this paper completed the intrusion detection of the cloud server and did not perform intrusion detection for each IoT device, which is a limitation of the NIDD model. In the future, we hope to complete the intrusion detection for each device, which will make the network communication of nursing homes more secure and further improve the network security level of nursing homes. In addition, the data set used for the training of the NIDD model is not the network communication data of the nursing home's cloud servers, which is also a limitation of the NIDD model. The next plan is to connect this model to the cloud server network of nursing homes one after another and collect accurate network communication data from the cloud servers of nursing homes in real-time while performing intrusion detection to optimize further the NIDD model of intrusion intelligence detection model proposed in this paper for application in nursing homes.

Acknowledgements

The work of this paper is supported by the National Key Research and Development Program of China (2019YFB1405000), National Natural Science Foundation of China under Grant (No.61873309, 92046024, 92146002).

Authors' contributions

Feng Zhou and Xin Du wrote the main manuscript text and Zhihui Lu provided the idea. Wenli Li and Jie Wu prepared the data and figures. All authors reviewed the manuscript. The author(s) read and approved the final manuscript.

Funding

The work of this paper is supported by the National Key Research and Development Program of China (2019YFB1405000), National Natural Science Foundation of China under Grant (No.61873309, 92046024, 92146002).

Availability of data and materials

The NSL-KDD dataset is from <https://www.unb.ca/cic/datasets/nsf.html>. The CIC-IDC-2018 dataset is from <https://www.unb.ca/cic/datasets/ids-2018.html>.

Declarations

Ethics approval and consent to participate

I confirm that I understand Journal of Cloud Computing is an open access journal that levies an article processing charge per articles accepted for publication. By submitting my article I agree to pay this charge in full if my article is accepted for publication. I declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper. The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration (from you or one of your Contributing Authors) by another publisher. I have read the Springer journal policies on author responsibilities and submit this manuscript in accordance with those policies. All of the material is owned by the authors and/or no permissions are required. I am the author responsible for the submission of this article and I accept the conditions of submission and the Springer Copyright and License Agreement as detailed above.

Competing interests

I declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Received: 15 September 2022 Accepted: 6 November 2022

Published online: 08 December 2022

References

- Chakrabarty B, Chanda O, Islam S (2017) Anomaly based intrusion detection system using genetic algorithm and Kcentroid clustering[J]. *Int J Comput Appl* 163(11):13–17
- Saraswathi S, Suresh S, Sundararajan N et al (2011) ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented[J]. *IEEE/ACM Transact Comput Biol Bioinform* 8(2):452–463
- Kumar P, Gupta GP, Tripathi R (2020) A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of things networks[J]. *J Ambient Intelligence Human Comput* 12(10):9555–9572
- Samadi Bonab M, Ghaffari A, Soleimani Gharehchopogh F, Alemi P (2020) A wrapper-based feature selection for improving performance of intrusion detection systems. *Int J Commun Syst* 33:e4434. <https://doi.org/10.1002/dac.4434>
- Kumar V, Das AK, Sinha DUIDS (2021) A unified intrusion detection system for IoT environment[J]. *Evol Intel* 14:47–59
- Kumar V, Sinha D, Das AK (2020) et al. An integrated rule based intrusion detection system: Analysis on UNSW NBI5 data set and the real time online dataset EJ. *Cluster Comput* 23(2):1397–1418
- Agarwal A, Sharma P, Alshehri M et al (2021) Classification model for accuracy and intrusion detection using machine learning approach EJ. *Peer J Comput Sci* 7(3):Article No.e437
- Khan SH, Hayat M, Bennamoun M (2018) IEEE Trans Neural Networks Learn Syst 29(8):3573–3587 (Cost-sensitive learning of deep feature representations from imbalanced data[J])
- Douzas G, Bacao F, Last F (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences* 465:1–20. ISSN 0020-0255. <https://doi.org/10.1016/j.ins.2018.06.056>
- Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications[J]. *Neurocomputing* 70:89–50113
- Lin SZ, Shi Y, Xue Z (2018) Character-level intrusion detection based on convolutional neural networks[C] // International Joint Conference on Neural Networks (IJCNN). IEEE, New York, pp 1–8
- Liu J, Yin L, Hu Y et al (eds) (2018) A novel intrusion detection algorithm or industrial control systems based on CNN and process state transition[C] // IEEE International Performance Computing and Communications Conference. IPCCC, Venice, pp 1–8
- Mohammad S, Rami K, Abdulrahman Y (2020) DoS and DDoS attack detection using deep learning and IDS[J]. *The International Arab. J Inform Technol* 17(4A):655–661
- Kasongo SM, Sun Yanxia (2020) A deep learning method with wrapper-based feature extraction for wireless intrusion detection system. *ScienceDirect[J] Computers&Security* 92:101752
- Min EX, Long J, Liu Q et al (2018) TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest[J]. *Secur Communication Networks* 2018:1–9
- Wang XS, He K, Song CB et al (1904) AT-GAN: a generative attack model for adversarial transferring on generative adversarial nets[J]. *arXiv Preprint, arXiv*, p 07793
- Hu WW, Tan Y Generating adversarial malware examples for black-box attacks based on GAN. *arXiv Preprint* <https://doi.org/10.48550/arXiv.1702.05983>
- Muhannadu MS (2019) Generative adversarial networks for launching and thwarting adversarial attacks on networks intrusion detection systems[C] // 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, Tangier, pp 78–83
- Salem M, Taheri S, Yuan J (2018) Anomaly generation using generative adversarial networks in host-based intrusion detection[C] // IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference(UEMCON). IEEE, Seattle, pp 683–687
- Ramasamy LK, Khan F, Shah M, Prasad BVVS, Iwendi C, Biamba C (2022) Secure Smart Wearable Computing through Artificial

- Intelligence-Enabled Internet of Things and Cyber-Physical Systems for Health Monitoring. *Sensors* 22:1076. <https://doi.org/10.3390/s22031076>
21. Onyema EM, Dalal S, Romero CAT et al (2022) Design of Intrusion Detection System based on Cyborg intelligence for security of Cloud Network Traffic of Smart Cities. *J Cloud Comp* 11:26. <https://doi.org/10.1186/s13677-022-00305-6>
 22. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–44. <https://doi.org/10.1145/3422622>
 23. Radford A, Metz L, Chintala S Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv Pre-print <https://doi.org/10.48550/arXiv.1511.06434>
 24. Chen TQ, Guestrin C (2016) XGBoost: A scalable tree boosting system[C]// *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA, :785–794. DOI:<https://doi.org/10.1145/2939672.2939785>
 25. Xu Y, Zhao X, Chen Y, Yang Z (2019) Research on a Mixed Gas Classification Algorithm Based on Extreme Random Tree. *Applied Sciences* 9(9):1728. <https://doi.org/10.3390/app9091728>
 26. Guolin K, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Tie-Yan L (2017) LightGBM: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, pp. 3149–3157
 27. Lundberg SM, Lee S (2017) A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, pp. 4768–4777

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)