Journal of Engineering
and Applied Science

RESEARCH Open Access

# Artificial intelligence-based adaptive anomaly detection technology for IaaS cloud virtual machines

Guoming Jiang[1*]

*Correspondence:
Guoming_Jiang2023@outlook.com

[1] Information Technology Center, Zhejiang Sci-Tech University, Hangzhou 310018, China

**Abstract**

As infrastructure-as-a-service clouds quickly grow, an increasing number of businesses and people are moving their application development to the cloud. The purpose of the research is to solve the problem of identifying memory anomalies in cloud virtual machines and improve the accuracy of the model in detecting abnormal situations. This paper presents a model for detecting virtual machine anomalies in IaaS cloud platform. The model considers the unique properties of monitoring metrics as time-series data and proposes an approach based on four important virtual machine monitoring metrics. The study also develops an adaptive anomaly detection system based on deep Q-network algorithms and migration learning principles for the variety of VM monitoring data in the cloud. The testing findings reveal that utilizing a Zoom layer with a 2-kernel size can increase detection accuracy to 96.7%. This demonstrates that a portion of the experimental data can extract the temporal features using the Zoom layer and different kernel sizes. The research model for anomaly detection had a classification accuracy of 99.8%. The deep Q-network model's final anomaly detection accuracy varies from 96.7 to 98.6%. The outcomes of the research improved the system's security and dependability, showed the worth of the overall framework design, and significantly decreased the number of resources needed for system operation and maintenance.

**Keywords:** Artificial intelligence, IaaS, Migration learning, Reinforcement learning, Anomalies, Detection

## Introduction

As the system architecture and service types in the cloud computing environment become more complex, the frequency of anomalies of various types of systems faced by cloud platforms is also increasing. Such anomalies will lead to cloud service downtime, data loss, and other failures, increasing the risk of loss for users [1, 2]. In the cloud computing environment, VM exception monitoring depends on analyzing logs and key performance indicators. With the increase of monitoring data, this consumes not only more system resources but also anomaly detection models that rely on historical data often have false positives in the face of new anomalies. The cloud system contains a large number of different types of server resources, and although virtualization technology combines them into a resource pool, different virtual machines still have different performance indicators when anomalies occur [3]. Infrastructure as a Service (IaaS) is an IT infrastructure organized through a network, providing users with a variety of different specifications of virtual machines; users can deploy any program they want in the rented virtual machines. However, such programs can also have various vulnerabilities, increasing the risk of damage to virtual machines and cloud platforms [4]. Memory anomalies often occur in virtual machines in IaaS cloud, and the occurrence of such anomalies is often ignored because of the use of users and is often accompanied by changes in other performance indicators. Therefore, it is necessary to design an anomaly detection method that can adapt to different scenarios and user needs according to key indicators such as CPU and memory. Deep neural networks can learn the unique data features of various anomalies and identify these anomalies. Among them, convolutional neural network (CNN) models are very good at automatically extracting important features from data. The deep Q-network (DQN) algorithm is a method based on reinforcement learning, which can not only detect anomalies but also learn through interaction with the environment and constantly adjust its detection and response strategies according to the behavior and feedback of the cloud platform to achieve continuous optimization [5]. Therefore, an intelligent adaptive anomaly detection model based on CNN model and deep Q-network algorithm is proposed from the perspective of IaaS cloud virtual machine memory anomaly-related problems. The study hopes to make the CC system more secure and stable. This lowers system overhead and the likelihood of abnormalities and failures. There are five primary sections to the research. The context and importance of the research on VMAD for IaaS clouds are introduced in part 1 of the essay. The second section provides an overview of VMAD, which is a thorough examination of the outcomes attained by specialists and academics both domestically and internationally in the field of VM detection. The study's methodology is covered in the " Methods" section, which is broken down into two main subsections. In the " AI-based self-adaptive AD approach for IaaS Cloud VMsa" paragraph, the study builds a cloud-based VM memory AD model based on four key VM monitoring parameters. The paper develops an adaptive anomaly detection (AD) system based on ML principles and the deep Q-network (DQN) algorithm in " AI-based VM memory AD model construction" subsection. The research approach is presented in the " Discussion" section along with an analysis of the experimental findings. Also given are the research methodology's flaws and suggestions for future research.

## Related work

As artificial intelligence (AI) advances quickly, numerous machine learning applications are becoming more and more common in AD-related sectors. To identify and categorize errors in intricate aviation systems, Ning et al. presented an automatic encoder. The effectiveness of the method in characterizing the state of aviation systems and reliably identifying various defect kinds has been demonstrated [6]. Zhang et al. developed an AD approach based on multivariate data streams. The technology offers more than 75% accuracy in fault identification, according to experiments, and may be used to build online models [7]. To track harmful activities on surveillance networks, Gayathri M. et al. developed an intrusion detection model based on a Gaussian model with a straightforward Bayesian methodology. Anomalies in the system were more accurately detected by the model than by the Gaussian model, according to experiments [8]. To enhance the effectiveness of conventional manual log checking methods, Chen et al. suggested a system log detection approach based on a symmetric organized dual long- and short-term memory network. The outcomes demonstrate that the method has some advantages for log identification and resolves the issue of long- and short-term memory networks' low prediction ability on long sequences [9].

In order to overcome the inapplicability of current log-based AD nonindustrial systems, the researchers Jing H. et al. recommended using human feedback to adjust the structure of the detection model in order to reduce false alarms. The study found that the technique significantly increased detection accuracy by detecting 70% fewer false alarms [10]. A human–computer interaction strategy for streaming AD was put forth by Li et al. and is based on an online adaptive forest algorithm. Studies have demonstrated that the method's added feedback usefulness can boost AD systems' performance while requiring fewer visitors [11]. To deal with the issue of virtual networks in highly dynamic contexts that produce potentially perplexing AD algorithms, Spiekermann et al. suggested a packet-based AD technique. Studies have demonstrated the method's efficacy [12]. To address the issues brought on the VM migration to cloud intrusion detection systems, Ibrahim et al. created a CC-oriented adaptive intrusion detection system. Studies have revealed that this system performs intrusion detection better than existing detection methods [13].

The identification of them varies depending on the range of data types present in the CC environment, according to a synthesis of national and international research. As a result, detection methods are less generic, and the outcomes of detection vary widely. Real-time monitoring and AD are required for VMs in CC, as detection based on previous data is prone to inaccuracy. Additionally, the algorithm's complexity and resource utilization must be considered. Current algorithms struggle to keep up with the complex and dynamic real-time operation of the cloud system because they are unable to promptly and accurately recognize novel situations and abnormalities as they are encountered. The paper suggests AI-based adaptive AD technology for IaaS cloud VMs to address these problems.

## Methods

### AI-based self-adaptive AD approach for IaaS cloud VMs

Memory anomalies are a common type of abnormal memory in VMs, from which deep neural networks can learn the unique data characteristics of various types of anomalies as they occur, in order to identify them. Therefore, the study classifies anomalies using various types of deep learning algorithms based on four key performance metrics: CPU, memory, hard disk, and network. The Zoom-CNN model is also designed to target the VM memory for AD. In addition, the research is based on the DQN algorithm to control the migration training module to achieve the best matching state of the VM and to realize the overall adaptive adjustment of the system.

## AI-based VM memory AD model construction

CC is a new model of resource sharing that incorporates technologies from various fields such as virtualization, distributed computing, and networking [14]. CL provides a large number of computing and storage server resources in a system packaged as abstract virtual resources through virtualization technology. Cloud users can access the purchased services through the network, regardless of time and space constraints. According to the hierarchy, CC includes three different layers of service models, as shown in Fig. 1.

Figure 1 shows that CC consists of Infrastructure as a Service (IaaS), Platform as a Service PaaS, and Software as a Service SaaS. PaaSCL companies offer server platforms as a service, whereas SaaSCL vendors offer application software and data as a service. IaaSCL providers consider IT infrastructure as a service, and users only need to purchase IaaS services to use, through the network The purpose of AD is to identify abnormal patterns or to mine data for values whose logic or characteristics do not match [15]. The cases where anomalies are detected are usually referred to as positive classes and those where they are detected as negative classes. Based on the correctness of the detection results, four cases exist, namely true anomalies, false anomalies, true normals, and false normals. According to such AD cases, four AD performance evaluation indicators can be obtained. For example, Eq. (1) is the expression for the calculation of the accuracy rate.

$$R_a = \frac{N_{TP} + N_{TN}}{N_T} \tag{1}$$

In Eq. (1), $N_{TP}$ is the number of samples detected as true abnormal, $N_{TN}$ represents the number of samples detected as false normal, and $N_T$ represents the total number of samples detected. The false alarm rate represents the ratio of the number of normal samples incorrectly detected as abnormal in the test to the total number of normal samples. The expression is calculated as shown in Eq. (2).

$$R_{FP} = \frac{N_{FP}}{N_{FP} + N_{TN}} \tag{2}$$

In Eq. (2), $N_{FP}$ represents the number of samples with false abnormal test results. The expression is calculated as shown in Eq. (3).

$$R_R = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{3}$$

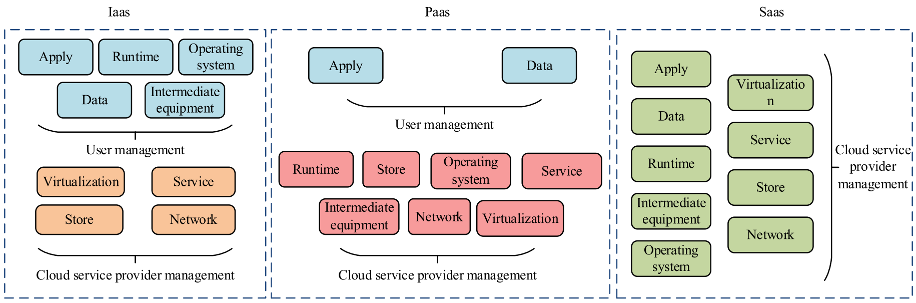


**Fig.1** Three service models of cloud computing

In Eq. (3), $N_{FN}$ represents the number of samples with false normal test results. The larger the value, the better the detection performance, and the expression is calculated as shown in Eq. (4).

$$R_p = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{4}$$

The receiver operating characteristic curve (ROC) curve, which is derived as indicated in Eq. (5) [16], can be used to represent the relationship between the false alarm rate and the positive alarm rate.

$$AUC = \int_0^1 R_{TP}(R_{FP})dR_{FP} \in [0, 1] \tag{5}$$

The prerequisite for CC management of hardware resources is virtualization, which can be used to run multiple virtual servers, or VMs, on physical servers [17]. The various types of monitoring metrics data of VMs in the cloud are sampled at certain intervals. The sampled data has a time sequence characteristic, i.e., time-series data. The recurrent neural network (RNN) retains certain historical data information when processing time-series data, which can achieve short-term information persistence. The expression of the computation of the hidden layer of the RNN after the RNN receives the input sample $x_t$ at time $t$ is shown in Eq. (6).

$$s_t = f(Ux_t + Ws_{t-1}) \tag{6}$$

In Eq. (6), $o_t$ is the output value, and $f$ is the activation function. $U$ denotes the input $x$ weight matrix, and $s_{t-1}$ is the previous value BBB, which is used as the weight matrix for this input. The computational expression of the RNN hidden layer is shown in Eq. (7).

$$o_t = g(Vs_t) \tag{7}$$

In Eq. (7), $s_t$ represents the memory value of the moment, and $V$ is the weight matrix of the output layer. If Eq. (6) is continuously substituted into Eq. (7), Eq. (8) exists.

$$o_t = Vf\left(Ux_t + Wf\left(Wf(Ux_{t-3} + \cdots) + Ux_{t-1}\right)\right) = g(Vs_t) \tag{8}$$

Although RNNs perform better on chance all sequence problems, the network can only process one time step at a time, meaning that RNNs are extremely computationally intensive. CNNs, on the other hand, are capable of massively parallel processing, and they can be structured with multiple layers of networks to obtain sufficiently large sensory fields to save a great deal of time, such that is the basic idea behind temporal convolutional network (TCN) [18]. The CNN convolution and pooling process is shown in Fig. 2.

As in Fig. 2, the CNN's convolutional kernel processes the input features one at a time, does a direct matrix multiplication summation, and adds the amount of deviation as the output. The effect of its action with the region near the input data is just enough to satisfy the research on the extraction of correlation features between VM performance metrics. To improve the AD performance, the study constructed a Zoom-CNN model based on CNN and TCN networks. The ReLU function is computationally simple and

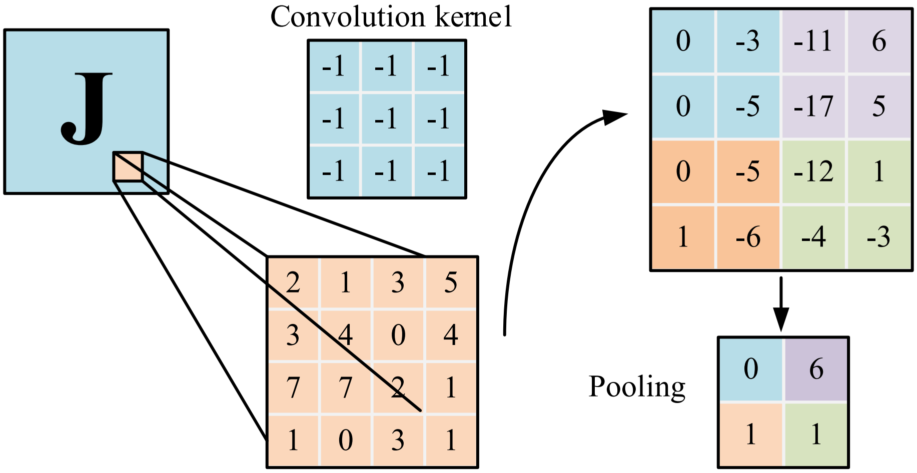


**Fig. 2** CNN convolution and pooling process

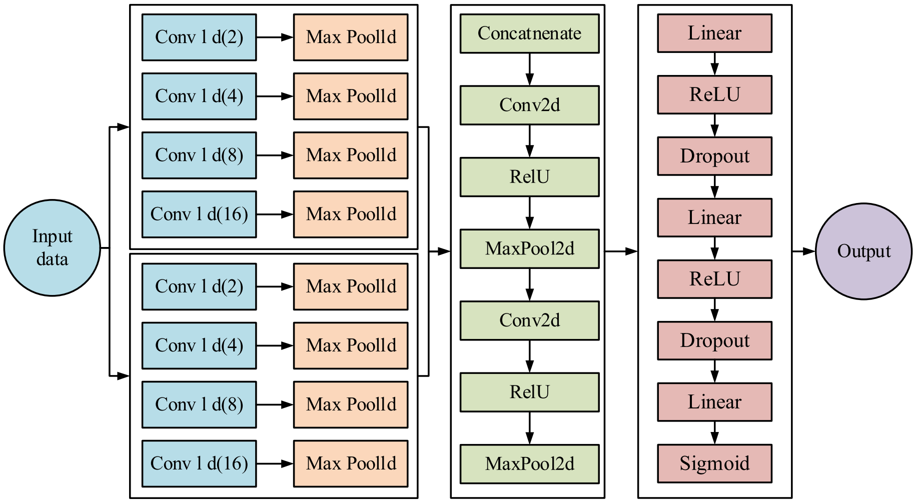


**Fig. 3** VM memory anomaly detection model based on Zoom-CNN algorithm

converges faster to avoid gradient disappearance. Therefore, as stated in Eq. (9), the study used it as the Zoom-CNN model's activation function.

$$\max(0,x) = f(x) \tag{9}$$

The final output of the network is $y$, which completes the classification task, and its computational expression is shown in Eq. (10).

$$y = \begin{cases} 1, h(X) \geq 0.5 \\ 0, h(X) < 0.5 \end{cases} \tag{10}$$

In Eq. (10), $X$ denotes the vector. The Zoom-CNN algorithm-based in-memory AD model for VMs in the cloud is shown in Fig. 3.

The Zoom layer, CNN convolutional layer, and fully connected layer are the three primary components of the model, as shown in Fig. 3. The first part is composed of multiple kernel sizes and 1-dimensional convolution with different step sizes. The first part consists of multiple kernel sizes and 1-dimensional convolutions of different step sizes, mainly for multidimensional temporal feature extraction of individual surveillance data. The second part consists of two two-dimensional convolutional layers with a kernel size of 3. It mainly implements correlation feature extraction for different performance metrics of VMs within training utilization, CPU utilization, hard disk usage, and network throughput [19]. The third part is the fully connected layer, whose input is the image processed by the convolutional layer. After passing the ReLU activation function, the classification of anomalies is finalized.

### Adaptive AD system design for IaaS cloud

In addition to the AD of the VM memory, the system as a whole need to be tested. During the operation of an IaaS cloud, the user and the VMs in the cloud are constantly interacting and using the various resources in them. As a result, a lot of monitoring data, logs, etc. are generated. These data can reflect the operational status of the VMs and thus react to abnormalities in the VMs. Based on virtualization technology, each physical server in the cloud data center can run multiple VMs of different configurations independently, and each VM shares the hardware resources of the server to which it belongs, making better use of memory, processors, hard disks, and networks through rational allocation and scheduling. Figure 4 illustrates a suggested adaptive AD approach that combines migration learning (ML) and reinforcement learning (RL) to address the AD problem of VMs in IaaS clouds.

The AD module, the model training module, and the central control module are the three primary hierarchical modules of the model, as depicted in Fig. 4. The goal of the AD module, which is situated at the CC system's edge, is to gather and preprocess the VM's CPU and memory in real time. The AD model is also tested, and the results are sent to the model training module. The study uses four performance metrics of VMs, based on the Zoom-CNNAD model, to perform real-time monitoring of memory

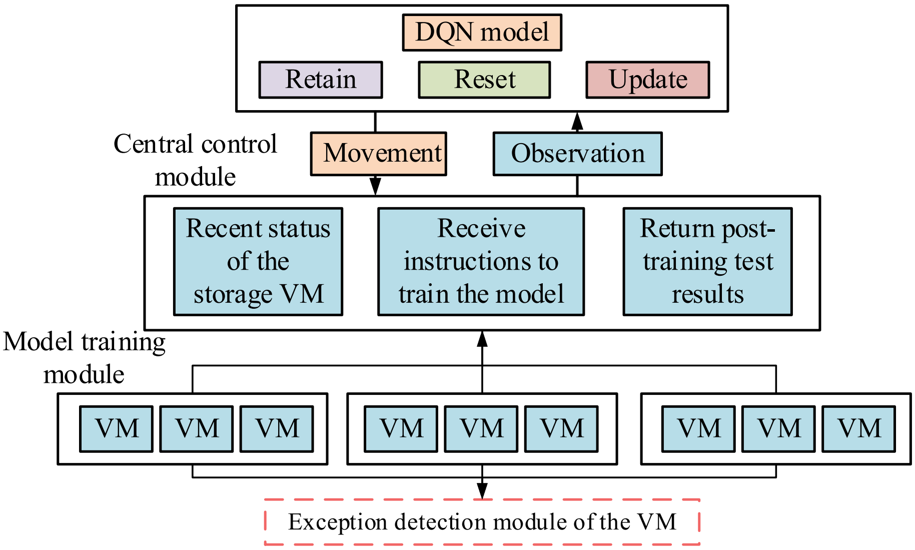

DQN model
Retain
Reset
Update
Central control module
Movement
Observation
Recent status of the storage VM
Receive instructions to train the model
Return post-training test results
Model training module
VM
VM
VM
VM
VM
VM
VM
VM
VM
Exception detection module of the VM

**Fig. 4** Adaptive anomaly detection model combining transfer learning and reinforcement learning

anomalies in target VMs. The migration training module corresponds to multiple target VMs, which are located at the edge server nodes. The framework structure of the migration training module is shown in Fig. 5.

As in Fig. 5, the role of the migration training module is to receive the monitoring indicator data with AD results and to save the parameters of the recent VMs. It will be compressed and sent to the central control module to receive instructions from this module. According to the instructions, the module's parameters are reset, model training is performed, etc., and the VM state results are fed back to the central control module. Based on ML principles, the model is first trained using existing data. The pre-trained wake-ups are placed into the migration training module to obtain sufficient feature information. The definition of ML is shown in Eq. (11).

$$D = \{\chi, P(X)\} \tag{11}$$

In Eq. (11), $\chi$ is the feature space, and $P(X)$ represents the edge probability distribution. Where, $X = \{P(X)\} \in \chi$. AD in different user models and VM environments can be viewed as a domain problem. The expression is shown in Eq. (12).

$$T = \left\{y, f(\cdot)\right\} \tag{12}$$

In Eq. (12), $y$ represents the label space, and $f(\cdot)$ AA is the target prediction function. The application scenarios of VMs in the cloud are also highly complex and often lack sufficient labeling data when specific to a particular aspect of the application. The study is based on RL theory to design the central control module [20]. The migration training module is also observed to enable adaptive control of the AD model. The system's core server houses the central control module, which corresponds to numerous model training modules. Figure 6 depicts the foundation for this module.

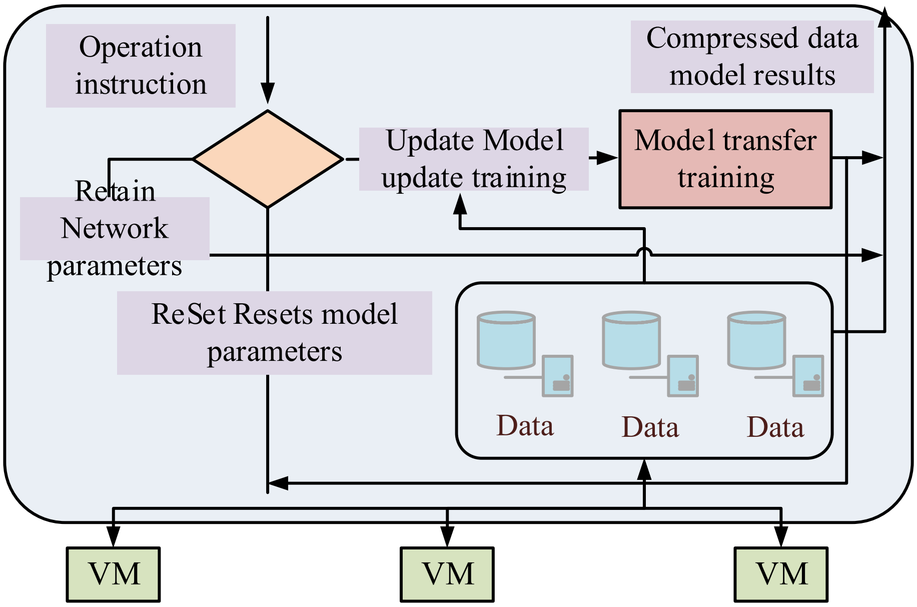

Operation instruction
Compressed data model results
Update Model update training
Model transfer training
Retain Network parameters
ReSet Resets model parameters
Data
Data
Data
VM
VM
VM

**Fig. 5** Framework structure of transfer training module

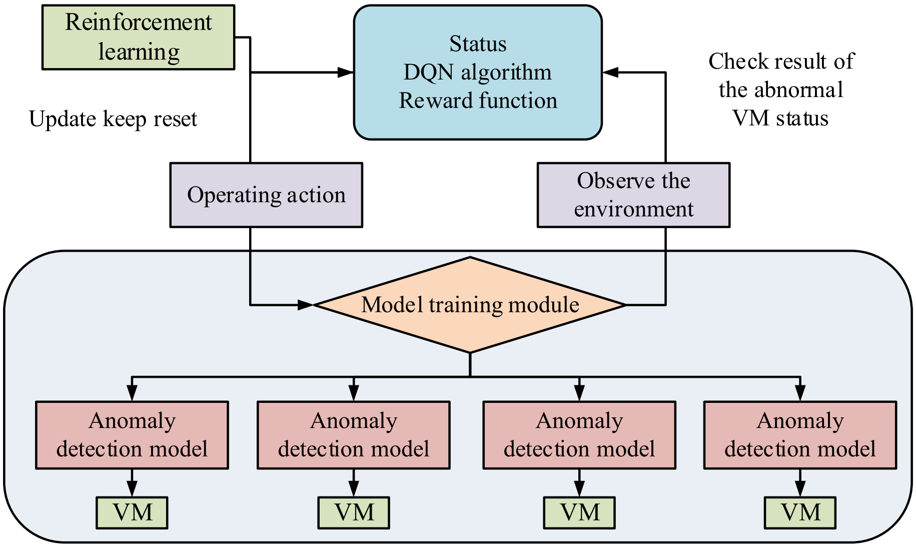


**Fig. 6** Structure of the central control module

This module's function, as shown in Fig. 6, is to accept the monitoring parameters, AD results, and other data given by the model training module. Calculate rewards through a reward function based on state parameters, detection results, and other content. Based on RL theory, the migration training module is augmented and controlled so that adaptive control of the AD model is achieved. Common RL is based on Markovian decision processes and is widely used in fields such as automatic control [21]. In the study, the reward function is then based on monitoring data and detection results. Its structure is constructed in proportion to the detection accuracy as a percentage of $\alpha_1 = 0.4$, the loss value as a percentage of $\alpha_2 = 0.3$, and the VM compression to a percentage of $\alpha_3 = 0.3$. The percentage of correct predictions in the batch sample is the accuracy rate. The expression is calculated as in Eq. (13).

$$R_1 = \frac{TP + TN}{Batch\_size} \tag{13}$$

The smaller the loss value, the better the detection and classification, which is expressed as shown in Eq. (14).

$$R_2 = 1 - \frac{1}{Loss} \tag{14}$$

The lower the value set, the higher the reward achieved, which preferably takes a value in the range 0–1 and is expressed in the calculation of Eq. (14).

$$R_3 = -\frac{1}{6 * Batch * Length * n} \sum_{i}^{Batch} \sum_{j}^{Length} \sum_{k}^{n} x_{ijk} + \frac{1}{2} \tag{15}$$

Therefore, the final reward function is shown in Eq. (16).

$$R = R_1 * \alpha_1 + R * \alpha_1 + R_{23} * \alpha_1 \tag{16}$$

In order to prevent the reward function from affecting the reward for the present action, a discount factor of 1, or the discount rate $\gamma \in [0, 1]$, is introduced here for the reward for future acts. The cumulative reward attained over time, assuming a total of $n$ acts during the process, is displayed in Eq. (17).

$$R_t = r_t + \gamma \cdot r_{t+1} + \cdots = \sum_{i=0}^{n} \gamma^k \cdot r_{t+k} \tag{17}$$

In the RL framework, an intelligent system can learn ways to perform decision tasks by interacting with the external environment. In the learning process, the environment is first observed to obtain the state $S_t$. and selects 1 action $A_t$, from a limited set of actions, to be executed in the task, and the intelligent body obtains an environmental reward $R_t$ through predefined. The observation of the environment state is repeated until learning is complete.

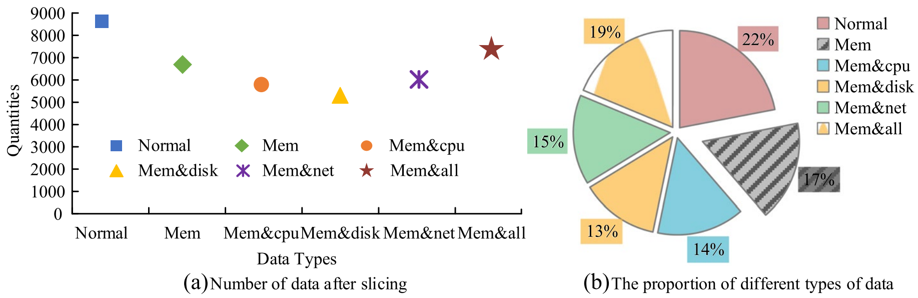
## Results and discussion

The study uses a convolutional neural network to extract correlation features between indicators and a Zoom layer to extract temporal features of monitoring indicator data. A convolutional kernel of kernel size 3 × 3 is chosen, and the step size and padding are both set to 1. The maximum pooling layer is chosen with a kernel size of 2 × 2. After the Zoom layer and two convolutional layers + the maximum pooling layer, the final output dimension is 64 × 8 × 15. The 64 time-series data from the monitoring metrics, which include CPU, RAM, hard disc, and the Zoom layer, can be chosen from four alternative 1-dimensional convolutional layer step sizes—2, 4, 6, 8, and 16—with the same kernel size. Table 1 displays the experimental setting and associated parameters.

By deploying plug-ins on IaaS virtual machines, the performance data of target virtual machines is collected at a fixed frequency. The raw data collected in real time includes CPU utilization, disk usage, free memory, and network traffic. Then establish a separate server as a data receiver to monitor the latest data in real-time. The

**Table 1** Experimental environment and related parameter settings

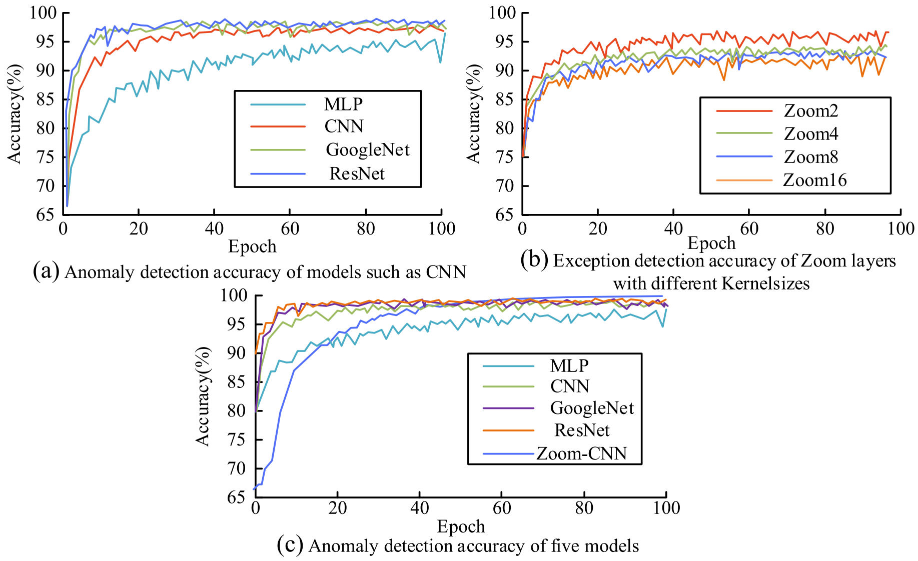
| Experimental environment | | Collected monitoring data indicators | | | |
|---|---|---|---|---|---|
| **Operating system** | **CentOS 7.5** | **Monitoring type** | **Monitoring index** | **Collection unit** | **Acquisition frequency** |
| CPU | Intel(R) Xeon(R) CPU E5-2407 2.20GHz4 core 4 threads | CPU | CPU usage | % | 12/min |
| Physical memory | 96G | Internal memory | Free memory space | Byte | 12/min |
| Physical hard disk | 2 T | Hard disk | Hard disk space | Byte | 12/min |
| Deep learning framework | PyTorch 1.10.1 | Network | Network traffic | Byte/s | 12/min |



(a) Number of data after slicing

(b) The proportion of different types of data

**Fig. 7** The quantity and proportion of each type of data



(a) Anomaly detection accuracy of models such as CNN

(b) Exception detection accuracy of Zoom layers with different Kernelsizes

(c) Anomaly detection accuracy of five models

**Fig. 8** Anomaly detection accuracy results of various models

monitoring frequency of the collection is 12 times per minute, and the data collection is carried out on 10 target servers for 5 days, and the monitoring data is more than 800,000 pieces. After processing, the quantity and proportion of each type of data are shown in Fig. 7.

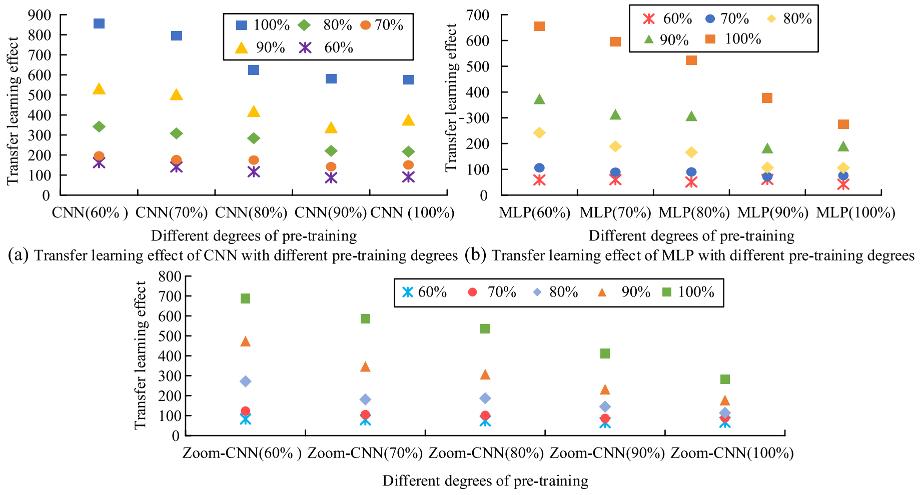
After the raw performance metrics are processed, 80% of the data is used for training, and the rest is used for testing. The researched Zoom-CNN-based VMAD model was compared with other models for AD accuracy, and the results are given in Fig. 8. The aim was defined as a binary classification of normal and abnormal samples.

Figure 8a shows that the MLP model can achieve an accuracy of 94% in anomaly detection of experimental data, while the CNN model can achieve 97.8% correct AD rate. It shows that both models have a good classification effect on the experimental data. Better training results can be achieved when using GoogleNet and ResNet networks for training, both of which can achieve a detection accuracy of 98% or more. Figure 8b demonstrates that the Zoom layer's detection accuracy can reach 96.7% when using a 2-kernel size. When the kernel size of the Zoom layer is 4, its classification accuracy is 93.8%. This indicates that using Zoom layers of different kernel sizes are able to extract temporal features from parts of the experimental data. The

Zoom-CNN model developed for the study, as shown in Fig. 8c, obtains 99.8% classification accuracy for AD, and as the number of training sessions grows, the final study model achieves greater AD accuracy than all other models. For each stage of the pre-training dataset reaching 60–100%, the model parameters of the CNN model, the MLP model, and the Zoom-CNN model were stored. The parameters were then used on a separate data set with 10 randomly selected batches of data. The training was then continued until the training batches required to reach the corresponding accuracy were reached, and the results are shown in Fig. 9.

The CNN model requires fewer training sessions than the MLP and Zoom-CNN models to achieve high accuracy and data representation. The Zoom-CNN model, on the other hand, is slow to improve at first, but it achieves higher detection accuracy. On pre-trained models, the best results can be achieved using the minimum number of training iterations when the model achieves 90% accuracy. Therefore, when conducting overall experiments on adaptive detection systems, model parameters that reach 90% accuracy in pre-training should be used. In summary, the effectiveness of AI can be demonstrated, and the consumption required to retrain the model can be effectively reduced by pre-training the model when performing similar tasks. Figure 10 displays, following one cycle of DQN model training, the detection accuracy of the AD model in the adaptive AD system for the experimental data.

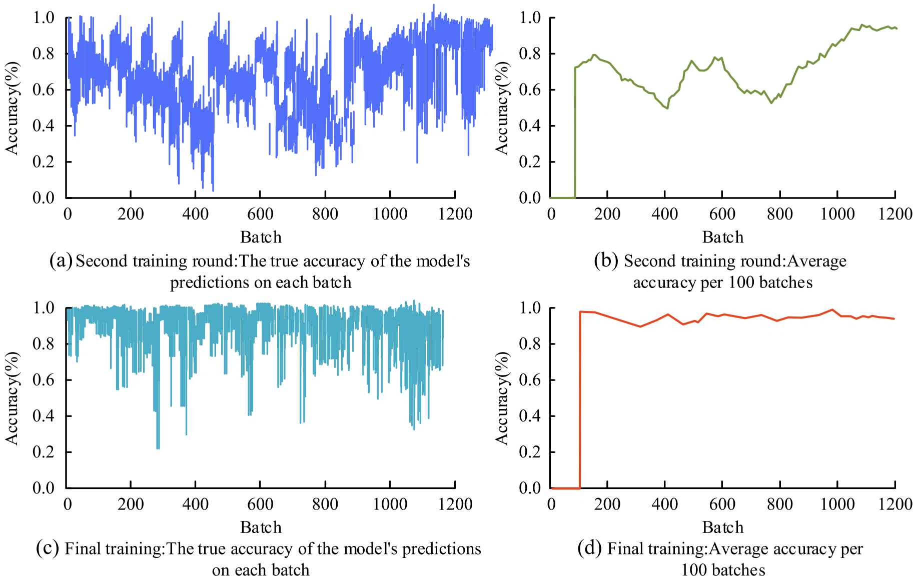
Figure 10a and b shows that the DQN model won't begin learning until there is enough data stored for observation. The DQN model will begin updating its AD model after a training cycle with a greater and better detection accuracy than the prior period of data. Figure 10c and d shows that the AD module was able to handle fresh data with a greater average detection accuracy following numerous training rounds. The DQN network's final AD accuracy ranges from 96.7 to 98.6%. The trials also employed fault injection to deliberately produce anomalies, and Fig. 11 illustrates how the monitoring data was gathered, together with manual processing, to determine the detection model's accuracy for anomalies in the virtual machine.



(a) Transfer learning effect of CNN with different pre-training degrees (b) Transfer learning effect of MLP with different pre-training degrees

(c) Transfer learning effect of Zoom-CNN models with different pre-training degrees

**Fig. 9** Transfer learning effects of models with different pre-training degrees



(a) Second training round:The true accuracy of the model's predictions on each batch

(b) Second training round:Average accuracy per 100 batches

(c) Final training:The true accuracy of the model's predictions on each batch

(d) Final training:Average accuracy per 100 batches

**Fig.10** DQN model training effect

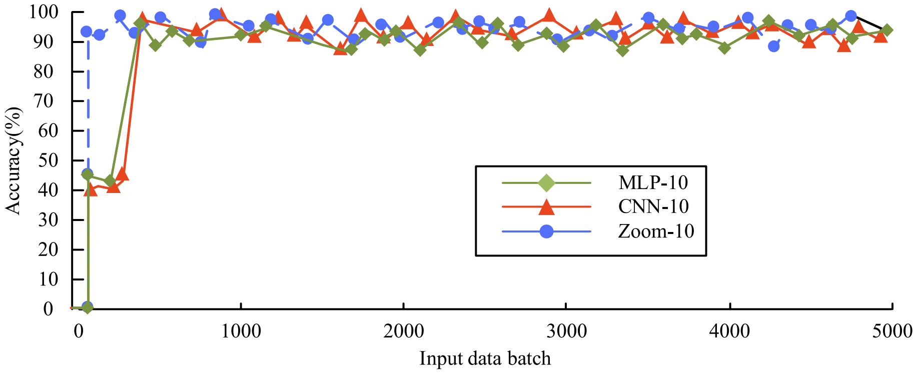
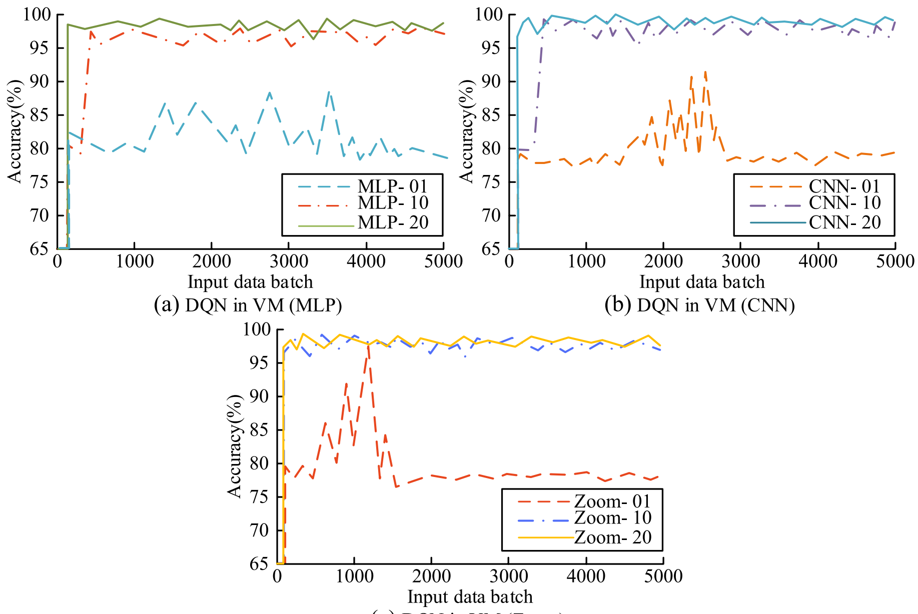


**Fig. 11** Detection results on the target virtual machine

Figure 11 shows that during a period of data training, the detection accuracy of the AD module based on all three models improved. The Zoom-CNN model produces somewhat better outcomes than the CNN model among them. In most instances, the MLP model was the source of both the CNN model and the Zoom-CNN model. The Zoom-CNN model not only achieves higher accuracy but also has stronger generalization than the other two models, proving the effectiveness of the model design. When the model was set for migration training, 10 training batches were performed each time, with a total of 200 batches making the detection accuracy of the module above 90%. The detection accuracy of the three models is shown in Fig. 12.

From Fig. 12a, it can be seen that the MLP model has the lowest detection accuracy for training one time when AD is performed on VMs. With the increase of batch, its accuracy variation ranged from 72.3 to 88.9%. The model has the highest detection accuracy for 20 times of training, with its accuracy variation ranging from 93.4–98.6%. Figure 12b demonstrates that when AD is applied to VMs, the CNN model has



(a) DQN in VM (MLP)

(b) DQN in VM (CNN)

(c) DQN in VM (Zoom)

**Fig. 12** Detection results of different times of model transfer training

**Table 2** Comparison results of performance indicators of each anomaly detection method

| Method | Precision/% | t/P | Recall/% | t/P | F1 score | t/P |
|---|---|---|---|---|---|---|
| Research algorithm | 97.42 ± 0.26 | / | 93.21 ± 0.33 | / | 98.13 ± 0.26 | / |
| References [7] | 84.42 ± 1.02 | 39.054/ < 0.05 | 88.01 ± 0.84 | 18.220/ < 0.05 | 89.22 ± 0.76 | 35.078/ < 0.05 |
| References [8] | 87.79 ± 0.74 | 38.825/ < 0.05 | 90.31 ± 1.30 | 6.837/ < 0.05 | 91.24 ± 0.37 | 48.181/ < 0.05 |
| References [11] | 88.63 ± 1.63 | 16.840/ < 0.05 | 91.26 ± 0.45 | 11.051/ < 0.05 | 91.74 ± 0.98 | 19.929/ < 0.05 |
| References [12] | 89.84 ± 0.73 | 30.932/ < 0.05 | 90.12 ± 1.22 | 7.732/ < 0.05 | 92.01 ± 1.02 | 18.386/ < 0.05 |
| References [13] | 89.33 ± 0.67 | 35.597/ < 0.05 | 90.36 ± 1.07 | 8.049/ < 0.05 | 90.86 ± 1.31 | 17.214/ < 0.05 |

022, 40(8): 2440–2455

the lowest detection accuracy for training one time. As the batch increases, its accuracy variation range is greater than that of the MLP model, at 73.8–91.7%. The model had the highest detection accuracy for 20 training sessions, with its accuracy variation ranging from 97.3–99.2%. Figure 12c demonstrates that when the VM is AD'd for one training, the Zoom model has the lowest detection accuracy. As the batch increases, its accuracy varies the most in the range of 77.3–97.5%. The model has the highest detection accuracy for 20 training times, and its accuracy variation range is the most stable, between 97.3 and 99.8%. SPSS 23.0 data analysis software was used for analysis. If the measurement data conform to normal distribution, it is expressed by means of mean ± standard deviation. Comparison between the two groups is conducted by testing. When $P < 0.01$, it can be considered that the data comparison has significant statistical significance. In order to further evaluate the superior performance of the research method, the precision, recall, and F1 score of the research method were compared with the anomaly detection methods mentioned in literatures [7, 8, 11, 12], and [13], respectively. Table 2 shows the comparison results of the performance indicators of each anomaly detection method.

As can be seen from Table 2, the precision, recall, and F1 score values of the research method are all higher than those of the other six anomaly detection methods, and the data of the research method and other methods have significant statistical significance ($P < 0.05$).

## Discussion

To realize the anomaly judgment of the target virtual machine, an improved Zoom-CNN anomaly detection model is designed based on TCN model and CNN. The classification accuracy of Zoom-CNN model is as high as 999.8%. The improved model effectively reduces the monitoring requirements of target VMS and the overall O&M resource consumption of the system. This method is better than the CNN-based system packet load anomaly detection method proposed by Song J. et al. [22]. The reason is that the model combines the temporal learning ability of TCN with the spatial extraction ability of CNN, thus improving the quality of feature representation. Secondly, a transfer training module based on transfer learning is proposed to realize parameter reset or training update of the model. Through transfer learning, only less than 200 batches of small batch data are needed for fast training to achieve better detection results. This method makes the model better match the target virtual machine. This result is similar to that of an unsupervised KPI anomaly detection method proposed by Zhang S. et al. [23]. This may be because there may be similar anomaly features among different virtual machines, and transfer learning can make use of the prior knowledge learned in one or more source tasks to quickly adapt to the new environment and help the model identify anomalies more accurately in the new environment. It can improve the generalization ability of the model while reducing the training time and calculation cost. Finally, an adaptive anomaly detection method based on DQN algorithm is proposed. The accuracy range of this model is the most stable, ranging from 97.3 to 99.8%. The precision, recall, and F1-score values of the research method are higher than those of other six advanced anomaly detection methods. This may be because the model can design a reward mechanism based on the VM state, fault identification results, and loss feedback from transfer learning. The transfer learning control is optimized to ensure the best matching state between anomaly monitoring and VM by reward driving, so as to realize the automatic adjustment of the system.

## Conclusions

To improve the efficiency of memory anomaly detection in IaaS cloud virtual machines and ensure the service quality of cloud users, an intelligent adaptive IaaS cloud virtual machine anomaly detection method is proposed. In this paper, memory, network, CPU, and hard disk are used as input data, and based on the improved Zoom-CNN model, it is used to determine the anomaly of target virtual machine. Based on transfer learning, the parameters of anomaly detection model are updated and trained. Finally, DQN algorithm is proposed to construct the central control module. The results show that the classification accuracy of Zoom-CNN model for anomaly detection is 99.8%. With the increase of training times, the accuracy of anomaly detection of the final research model is higher than that of other models. After repeated training, the anomaly detection module has been able to cope with new data with a high average detection accuracy, and the final anomaly detection accuracy of DQN network ranges from 96.7 to 98.6%. When Zoom model is used to detect anomalies in virtual machines, the detection accuracy of 20 training times is the highest, and the accuracy range is the most stable, ranging from 97.3 to 99.8%. The precision, recall, and F1 score values of the research method were higher than those of the other six advanced anomaly detection methods, and the data of the research method were statistically significant compared with those of other methods ($P < 0.05$). This shows that compared with the existing anomaly detection methods, the research method has stronger generalization and superiority in data anomaly detection. Although the research method has a high detection rate, limited by the experimental environment, the main target of the research is web service type virtual machine, and more training models for different virtual machines will be designed in the future. In addition, the proposed method currently does not provide more precise control over the transfer training of the model, so the model transfer training will be improved in the future, and more detailed regulation will be added, such as selecting key parameters and deciding the number of iterations to better adapt to the environment.

**Abbreviations**

| | |
|---|---|
| CC | Cloud computing |
| CL | Cloud service |
| IaaS | Infrastructure as a Service |
| VMs | Virtual machines |
| DQN | Deep Q-network |
| CNN | Convolutional neural network |
| AD | Anomaly detection |
| AI | Artificial intelligence |
| TCN | Temporal convolutional network |
| RL | Reinforcement learning |
| ML | Migration learning |

**Acknowledgements**
Not applicable

**Author's contributions**
GMJ wrote original draft; participated in literature search and analyses, evaluations, and manuscript preparation; and wrote the paper.

**Funding**
Not applicable.

**Availability of data and materials**
The data involved in this paper will be made public as required.

## Declarations

**Competing interests**
The author declares no competing interests.

Received: 8 August 2023 Accepted: 21 April 2024
Published online: 30 April 2024

**References**

1. Thilagam T, Aruna R (2021) Intrusion detection for network-based cloud computing by custom RC-NN and optimization. ICT Express 7(4):512–520
2. Mohammed CM, Zeebaree SRM (2021) Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS: a review. Int J Sci Bus 5(2):17–30
3. Son Y, Lee YS (2022) A smart contract weakness and security hole analyzer using virtual machine based dynamic monitor. J Logist Inform Serv Sci 9(1):36–52
4. Bisht PS, Mishra P, Chauhan P, Joshi RC (2023) HyperGuard: on designing out-VM malware analysis approach to detect intrusions from hypervisor in cloud environment. Int J Grid Util Comput 14(4):356–367

5. Wang Y, Mao M, Chang L, Hatziargyriou ND (2023) Intelligent voltage control method in active distribution networks based on averaged weighted double deep Q-network algorithm. J Modern Power Syst Clean Energy 11(1):132–143
6. Ning S, Sun J, Liu C, Yi Y (2021) Applications of deep learning in big data analytics for aircraft complex system anomaly detection. J Risk Reliability 235(3):923–940
7. Zhang Q, Han J, Cheng LI, Zhang B, Gong Z (2022) Approach to anomaly detection in microservice system with multi-source data streams. Zte Technol English 20(3):85–92
8. Gayathri M, Pramila PV (2022) Analysis of accuracy in anomaly detection of intrusion detection system using Naive Bayes algorithm compared over Gaussian model. ECS Trans 107(1):13977–13991
9. Chen Y, Luktarhan N, Lv D (2022) LogLS: research on system log anomaly detection method based on dual LSTM. Symmetry 14(3):454–454
10. Jing H, Tong J, Yifan WU, Chuanjia H, Ying LI (2021) FeedbackAware anomaly detection through logs for largescale software systems. Zte Technol English 19(3):88–94
11. Qingyang LI, Zhiwen YU, Huang XU, Guo B (2023) Human-machine interactive streaming anomaly detection by online self-adaptive forest. Front Comput Sci China English 17(2):145–156
12. Spiekermann D, Keller J (2021) Unsupervised packet-based anomaly detection in virtual networks. Comput Netw 2:108017. https://doi.org/10.1016/j.comnet.2021. 192(Jun.19):1-17
13 Ibrahim NM, Zainal A (2019) An adaptive intrusion detection scheme for cloud computing. Int J Swarm Intell Res 10(4):53–70
14. Jangjou M, Sohrabi MK (2022) A comprehensive survey on security challenges in different network layers in cloud computing. Arch Comput Methods Eng 29(6):3587–3608
15 Mutulu PM, Kahonge AM (2021) A multi-tenancy cloud trust model using quality of service monitoring: a case of Infrastructure as a Service (IaaS). Int J Comput Appl 174(27):41–46
16. Martin M, Kleinhenz MD, Schwartzkopf-Genswein KS, Melendez D, Marti S, Pajor EA (2022) Characterizing the diagnostic sensitivity and specificity of pain biomarkers in cattle using receiver operating characteristic curves. J Dairy Sci 105(12):9853–9868
17. Ganesan S, Ganesan S (2021) A multi-objective secure optimal VM placement in energy-efficient server of cloud computing. Intell Autom Soft Computing 29(3):387–401
18. Bi J, Zhang X, Yuan H, Zhang J, Zhou M (2021) A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. IEEE Trans Autom Sci Eng 19(3):1869–1879
19. Masood F, Masood J, Zahir H, Driss K, Mehmood N, Farooq H (2023) Novel approach to evaluate classification algorithms and feature selection filter algorithms using medical data. J Comput Cogn Eng 2(1):57–67
20. Yunxiu Z, Kai XU (2023) Recognition and interfere deceptive behavior based on inverse reinforcement learning and game theory. Syst Eng Electronic English 34(2):270–288
21. Ye MA, Tianqing C, Wenhui F (2021) A single-task and multi-decision evolutionary game model based on multi-agent reinforcement learning. Syst Eng Electron English 32(3):642–657
22. Song JY, Paul R, Yun JH, Kim HC, Choi YJ (2021) CNN-based anomaly detection for packet payloads of industrial control system. Int J Sens Netw 36(1):36–49
23. Zhang S, Zhong Z, Li D, Fan Q, Sun Y, Zhu M, Liu YL, Yang H, Zou Y (2022) Efficient KPI anomaly detection through transfer learning for large-scale web services. IEEE J Sel Areas Commun 40(8):2440–2455

## Publisher's Note

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