

Article

Design and Verification of a Security Policy to Defense DDoS Attack for Cloud Robot

Boyi Liu ^{1,2}, Jieren Cheng ^{1,4,*}, Yijie Li ¹, Xiangyan Tang ^{1,*} and Jiquan Chen ^{1,3}

¹ School of Information Science and Technology, Hainan University, 570228, Haikou, China

² University of Chinese Academy of Science, 100000, Beijing, China;

³ School of Software and Microelectronics, Peking University, 100000, Beijing, China

⁴ State Key Laboratory of Marine Resource Utilization in South China Sea, 570228, Haikou, China

* Correspondence: cjr22@163.com, liuboyilby@163.com; Tel.: +86 15108910688

Abstract: Cloud robot is becoming popular and security of cloud robot is important. However, the researches of cloud robot safety are a few. This work develops a security policy to defense DDoS attack of cloud robot. In this policy, complex, but accurate calculation models are deployed on the cloud, simple but efficient calculation models are deployed on the robot. In the cloud, there are master server and standby server. The master server transfers parameters of complex but accurate models to the standby server periodically and the master executes the start-stop backup policy. Specifically, this work proposes and proves an algorithm to dynamically adjust the interval of parameter transfer. According to a PDRA feature of Netflow, when DDoS attacks, the master server sends warning signals to the robot and standby server. The robot runs local models to avoid stopping work until it is connected to the standby server. Then, standby server provides service to the robot until the master server recover. Finally, this work implemented a gesture recognition cloud robot based on convolutional neural network, hidden Markov model and PDRA feature of Netflow to verify the policy. Experiment shows that the security policy to defense DDoS attack for cloud robot is effective.

Keywords: Cloud robotic; Security policy; DDoS

1. Introduction

Since the concept of the cloud robot was proposed by Dr. Kuffner of Carnegie Mellon University (now working at Google company) in 2010 [1], the research on cloud robots is rising gradually. Cloud robots are the combination of cloud computing and robotics. Like other network terminals, the robot itself does not need to store all the information or has a strong computing power. It only needs to demand from the cloud, and the cloud is responsive and satisfied. The idea of connecting a robot to an external computer appeared in 1990s. University of Tokyo Inaba proposed the concept of remote brain. Cloud robots will further explore this concept and explore the way to achieve cheaper computing and interconnect with ubiquitous networks. Since the concept of cloud robot was put forward, it has aroused the interest of many IT companies, scientific research institutions and researchers at home and abroad.

Cloud robot has become an important direction of research at home and abroad. The application of cloud robots is increasing with the research of cloud technology. At the beginning of 2011, the 4a study program of RoboEarth [2] was initiated by the Eindhoven university of Technology, Swiss Federal Institute of Technology Zurich, Technische Universitaet Mnchen, Saragossa university, Universität Stuttgart, and 35 scientists from Philips. Trying to get the robots to share information and store their discoveries helps the robots build their own Internet and Wikipedia. In early 2014, RoboEarth project was launched by scientists at the Eindhoven University of Technology. Four robots collaborated in a simulated hospital environment to take care of the patient through interaction with the cloud backend server, which enables them to share information and learn from each other. Google

engineers have developed robot software based on the Android platform, which can be used for remote control based on the Lego mind-storms, iRobot Create and Vex Pro, etc. [3]. The ASORO lab in Singapore has built a cloud computing architecture that allows robots to build 3D maps of the current environment. In addition, its building speed is much faster than the robot's onboard computers [4]. KEHO B et al of the university of California, Berkeley, based on cloud platform, used Willow Garage's PR2 robot and valley song target recognition engine to complete the 3D robot fetching task [5]. Researchers at France's Toulouse system analysis and structural laboratory created a "user manual" that can be stored in each target to help the robot complete its operational tasks [6]. In the children's hospital in Italy, the Nao robot of French Aldebaran relies on cloud architecture to perform speech recognition, face detection and other tasks, which facilitates the interaction between robots and patients [7]. KAMEI K et al. proposed a mall wheelchair robot, sharing map information through the cloud, positioning and navigation using cloud architecture, and helping disabled or disabled elderly people visit the mall [8].

Through the design of an autonomous low maintenance infrastructure in the cloud and can provide the resources based on supply and demand, and create an intelligent engine in cloud infrastructure, real-time resource scheduling and management of the robot, simplify the robot hardware facilities, robot knowledge sharing through the cloud, and ultimately the formation of the cloud robot system. The cloud robot becomes more and more popular, but few people consider its security. There are no papers about cloud robot security in google scholar or some other platforms. Most of cloud robots require accurate results from cloud to work. So, it will bring huge losses [9] if the cloud is attacked. When the cloud is attacked, the driverless cars will stop and cause traffic jams. It will even result in more serious consequences. If the cloud robot can play Go like AlphaGo Zero [10], it will lose the game when the cloud is attacked. If the cloud robot used in military [11], it is possible to stop working or even to attack the friendly army and lead to war failure.

There are many ways of cyber-attacks, DDoS is one of the most common. If DDoS attacks the server of the cloud robot, the robot will stop work. Which may cause great loss. Recently, there are a various of data on the network because of the development of big data. For years, Distributed Denial of Service (DDoS) attack has been one of the greatest threats to network security.

In recent years, the technologies that use cloud computing to protect network security tends to mature. Based on the consideration of the operation and maintenance costs of physical devices, more and more enterprises and organizations are using their global services provided by cloud computing technologies to migrate their enterprise applications to the cloud environment to effectively defense DDoS attacks. Reference [12] proposed to use the public cloud defense DDoS attack to classify the visiting data by historical records, thus filtering the malicious IP source address. Reference [13] used Netflow protocol to collect data traffic and detect DDoS attacks in a self-learning manner. Reference [14] presented the design, implementation, and evaluation of dynamic PID (D-PID), a framework that uses PIDs negotiated between the neighboring domains as inter-domain routing objects. The authors in [15] proposed a novel DDoS attack detection system based on Spark framework. The authors in [16] proposed a multi-queue SDN controller scheduling algorithm based on time slice allocation strategy.

Cloud robot has many challenges [17], safety is important. This work addresses the safety problem of the cloud robot by the example of DDoS attack. This work designed a security policy to defense DDoS attack for cloud robot and implemented a gesture recognition robot to verify.

2. Design of a security policy to defense DDoS attack for cloud robot

2.1 Cloud robot system design matched with the security policy

As shown in Fig.1, when there is no network attack, data will be collected by sensors of the robot. Then, it will be uploaded to the master server. Relying on powerful computing power, the master server drives complex models (such as Deep Learning Models) to get accurate results immediately. The master server sends the calculation results to the robot. As the system running, the master server transfer parameters of complex models to the standby server. What's more, the master server share

knowledge with the cloud platform based on transfer learning or some other algorithms to improve the performance of complex models. With the system running like this, the server gives a full play to the advantage of computing power. And the robot will make full use of the ability of information acquisition. Not only does the robot get accurate results, but also it removes heavy computing equipment.

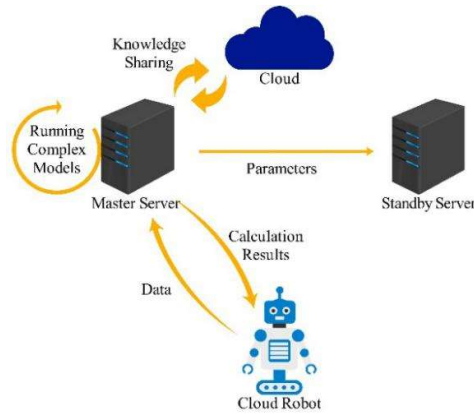


Figure 1. Design of the cloud robot against DDoS attack

The advantages of this scheme are as follows:

Guaranteeing the advantage of the cloud robot: computing in the cloud. The computation speed is much faster than that is only in the robot. Heavy equipment removed from robots.

Adding the knowledge sharing in the cloud. The math model does not only learn from the data collected by the robot, but also from the data from the cloud. With the development of transfer learning and some other machines learning method, this function module will play an increasingly important role.

Adding the standby server. Instead of using “One server-robot” mode, the greatest degree of accuracy will be guaranteed when the DDoS attacks occur.

This mode needs strong communication support. Remote communication speed has developed a lot in recent years. As a result, it has already satisfied with the communication requirements. And with the rapid development of the communication, this mode will be more effective and stable in the future.

2.2 Dynamically parameters transfer algorithm in the security policy

According to the characteristics of Poisson distribution, we first assume that the frequency of DDoS attacks obeys Poisson distribution in the process of transferring, parameters will run in a reasonable cycle better. So, this work presents a setting method of the frequency of the parameters transfer from master server to the standby server in unit time. Firstly, this work proves that the frequency of DDoS attacks obeys the Poisson distribution. Then, stationary of this process are proved. Finally, according to the nature of stationary, this work presents the setting method of the frequency of the parameters transfer from master server to the standby server.

ion [18]. Then we proved it: A random process $\{N(t), t \geq 0\}$ is called a counting process. $N(t)$ is random variable, representing the number of DDoS attacks. If it satisfies the following conditions, this counting process is called the Poisson process with the rate $\lambda (\lambda > 0)$ [19].

$N(0) = 0$; $\{N(t)\}$ is an independent incremental process;

For any $t, s \geq 0, N(s, t+s]$ obey the Poisson distribution with parameter λt . That is:

$$P(N(s, t+s] = k) = \frac{\lambda t^k}{k!} e^{-\lambda t}; k = 0, 1, \dots \quad (1)$$

Frequency of the occurrence of DDoS is regarded as random variety $N(t)$, among this, $t \geq 0$. When $t \geq 0$, $N(0) = 0$ obviously. So, the condition 1 is satisfied.

$N(t) - N(s) (0 \leq s \leq t)$ is the increment in $(s, t]$. The occurrence of DDoS attack is mutually independent. So, as for any positive integer n and any $0 \leq t_0 \leq t_1 \leq t_2 \leq \dots \leq t_n$. There are n increments: $X(t_1) - X(t_0), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})$, which are interdependent. So, we can call $\{N(t), t \geq 0\}$ processes with independent increments and it is satisfied with condition (2). In a sufficient small-time interval h , only one DDoS attack event can occur. The probability of two or more occurrences is extremely low and negligible. That can be expressed as this:

$$P\{N(h) \geq 2\} = o(h); h > 0 \quad (2)$$

In a sufficient small-time interval h , it is assumed that the probability of 1 DDoS attack is proportional to the length of the time interval of h . That can be expressed as this:

$$P\{N(h) = 1\} = \lambda h + o(h); h > 0 \quad (3)$$

In the formula 3, $\lambda (\lambda > 0)$ is the rate of the counting process.

We can prove that the random variable $N(t)$ obeys the Poisson distribution by simply verifying formula (2) and (3) satisfy the formula (1).

This work makes the $P_n(t) = P\{N(t) = n\}$ and $h > 0$. Then we can get:

$$\begin{aligned} P_0(t+h) &= P\{N(t+h) = 0\} \\ &= P\{N(t) = 0, N(t+h) - N(t) = 0\} \\ &= P\{N(t) = 0\} \cdot P\{N(t+h) - N(t) = 0\} = P\{N(t) = 0\} \cdot P\{N(h) = 0\} \end{aligned}$$

From the equation (2) and (3), we can get:

$$P_0(t+h) = P_0(t)[1 - \lambda h + o(h)]$$

Get:

$$\frac{P_0(t+h) - P_0(t)}{h} = -\lambda P_0(t) + \frac{o(h)}{h} \quad (5)$$

Let $h \rightarrow 0$, take the limit and then we can get the differential equation:

$$P'_0(t) = -\lambda P_0(t) \quad (6)$$

From $P_0(0) = P\{N(0) = 0\} = 1$, we can get:

$$P'_0(t) = e^{-\lambda t} \quad (7)$$

Similarly, for $n \geq 1$, a differential equation can be obtained:

$$P'_n(t) = -\lambda P_n(t) + \lambda P_{n-1}(t) \quad (8)$$

When $n=1$, from $P_0(t) = e^{-\lambda t}$ and $P_1(0) = 0$, we can get:

$$P'_1(t) = \lambda t e^{-\lambda t} \quad (9)$$

According to mathematical induction and $P_n(0) = 0$, we can get:

$$P_n(t) = \frac{e^{-\lambda t} (\lambda t)^n}{n!} \quad (10)$$

The formula (10) can prove that this random eligible constraint (3), so we can deem that the random $N(t)$ obeys Poisson distribution. Therefore, the occurrence frequency of DDoS attacks obeys Poisson distribution. Then, we can easily draw the conclusion that the occurrence frequency of DDoS attacks and DDoS disappears are both obey Poisson distribution. Finally, we get that the frequency of the state of the server is overturned obeys Poisson distribution.

Now, we have known that the random process $N(t)$ (The state of the server is overturned) obeys the Poisson distribution. Then, this work proves the stationary of the process.

We introduce s as a time point. So, $N \sim P(\lambda(t-s))$ in $[s, t]$.

$$\text{That is: } P(N=k) = \frac{(n(t-s))^k}{k!}$$

We suppose:

$$p(x(0)=1) = p(x(0)=-1) = \frac{1}{2} \quad (11)$$

That is:

$$P(n=k) = \frac{(\pi(t-s))^k}{k!} \quad (12)$$

Then:

$$E(X(t)) = 1 \times P(N \text{ is even}) + (-1) \times P(N \text{ is odd})$$

$$P(N \text{ is even}) = \sum_{k=0}^{\infty} \frac{(\pi(t-s))^{2k}}{(2k)!} e^{-\pi(t-s)} \quad (13)$$

$$P(N \text{ is odd}) = \sum_{k=0}^{\infty} \frac{(\pi(t-s))^{2k+1}}{(2k+1)!} e^{-\pi(t-s)} \quad (14)$$

We know:

$$e^{(x)} = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (15)$$

We can get:

$$e^{(\lambda(t-s))} = \sum_{k=0}^{\infty} \frac{(\lambda(t-s))^k}{k!} \quad (16)$$

$$e^{(-\lambda(t-s))} = \sum_{k=0}^{\infty} \frac{(\lambda(t-s))^k}{k!} (-1)^k \quad (17)$$

From the above formulas, we can get:

$$E(x(t)) = e^{(-\pi t)} \quad (18)$$

Then we check second moment, and we can easily get:

$$E(x(t)x(s)) = 1 \cdot P(N(t-s) \text{ is even}) - P(N(t-s) \text{ is odd}) = e^{(-\pi|t-s|)} \quad (18)$$

Whether the first moment or the second moment, they are only dependent on the $|t-s|$, so the process is stationary. Because attacks and non-attacks always come in pairs, so the process of DDoS attack is stationary also. According to the feature of stationary: mathematical expectations in unit time remain unchanged. Then, we can get the reasonable parameters transfer cycle:

$$Et = \frac{m(T_1) + m(T_2) + \dots + m(T_n)}{nT} \quad (19)$$

$$T_{\text{transfer}} = \frac{T - T_{\text{attack}}}{E_t - Num_{\text{attack}}} \quad (20)$$

In the formula (19), E_t is the mathematical expectation of the DDoS attack frequency. T is a cycle time. n means the n cycle. m is the number of DDoS attacks in the N cycle. T_{transfer} is the time required for the next parameter transfer. T_{attack} is the time of the last DDoS attack during the cycle. Num_{attack} is the number of DDoS attacks during this cycle. By this way, the system can transfer the parameters dynamically based on the frequency of DDoS attacks. The parameters can be transferred legitimately.

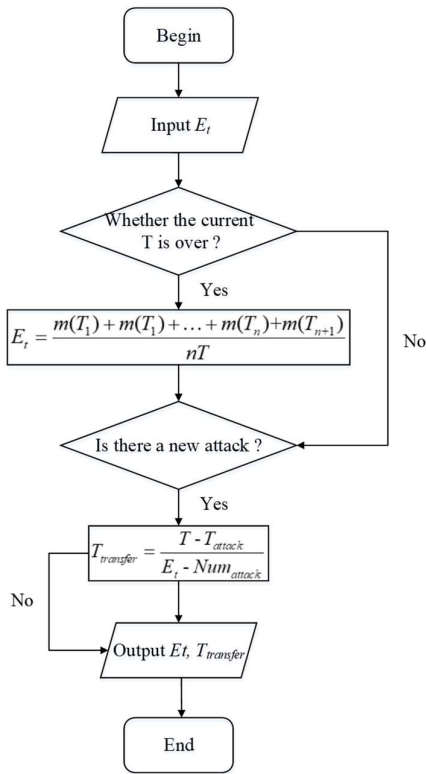


Figure 2. Flow chart of the dynamically parameters transfer algorithm

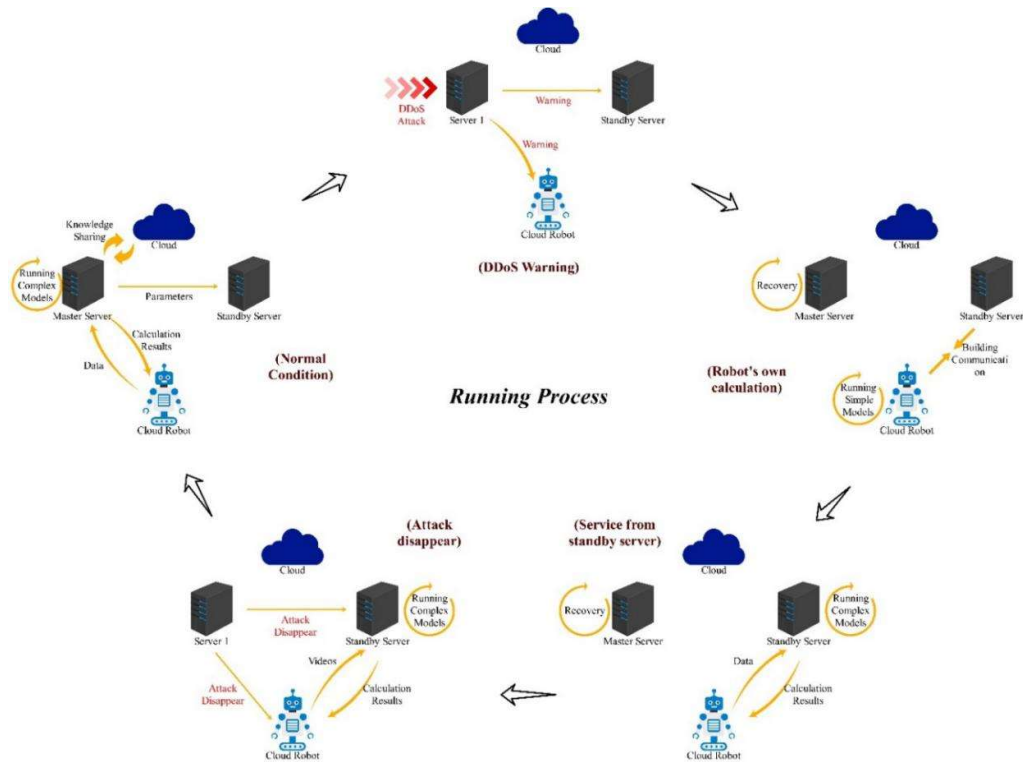
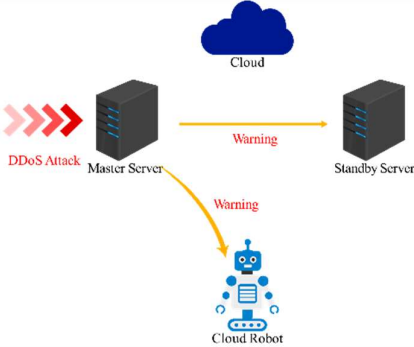
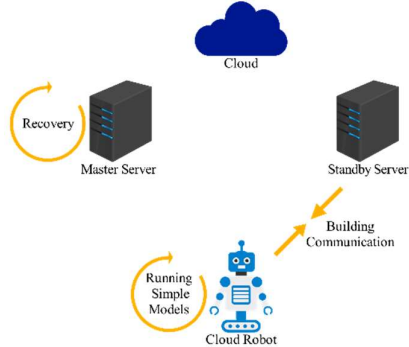
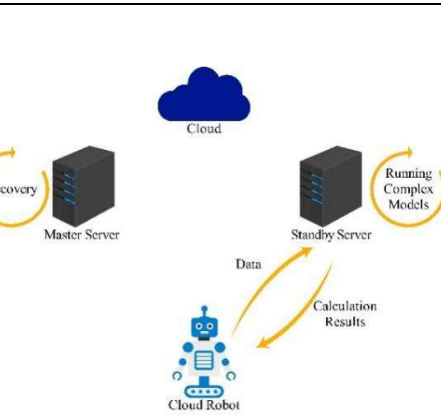
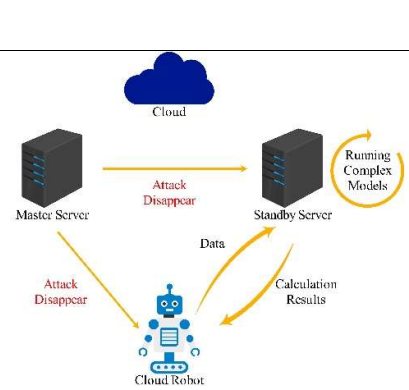


Figure. 3 Running processes of the cloud.

Periodically passing parameters is one of the policies. Another policy is start-stop backup. This policy makes backup from the master server to the standby server when the former start or stop work. Which can increase the robust of the security policy.

2.3 Cloud robot system running process of the security policy

Table 1. Running processes of different conditions and its working principle

Running Process	Working Principle
	<p>When there is DDoS attack, the master server will send warning signals to the robot and standby server, as illustrated in Fig.2. Considering the attack is quick, the probability of a parameters transfer completing in an instant is very small. So, this work doesn't design the parameters transfer process when there is DDoS attack warning.</p>
	<p>After warning, the master server need to be repaired. The robot runs simple models. At the same time, it will build communication with the standby server, as illustrated in Fig.3. By this way, the robot won't stop working even though there is network attack. Results from simple models are usually more accurate than complex models. But the inaccurate losses are far less than the losses due to stop working. To say the least, the simple models have developed a lot and they don't have too many errors.</p>
	<p>When the communication between the robot and standby server has built, the run model between the robot and the standby server is the same as with the master server, as illustrated in Fig.2. We don't know when the master server will be recover, so we have to prepare a standby server. Accuracy sometimes is import, so this work just regards the process of running simple models as a transition. The standby server will provide computing service for the robot when it has built communications with the robot. Relying on the parameters transferred timely from the master server, the standby server can set up an accurate model immediately. Then the robot can get real-time and accurate result.</p>
	<p>When the master server has been recovered, the master server sends the signal that there is no DDoS attack to the robot and the standby server. The priority of the master server is higher than that of the standby server. The robot transfer data to the master server when the master server provide service.</p>

Running processes of the system designed for the cloud robot include the normal condition, the DDoS warning stage, the robot's own calculation stage, the service from standby server stage and the attack disappear stage (as illustrated in Fig.3). When there is no DDoS attack, the system runs as normal. When there is DDoS attack, the robot turns to the DDoS warning stage. Then, the robot runs simple models with supports from the computing equipment of itself, that is the robot's own calculation stage. But it is not a permanent solution, so this work provides the service from the standby server to the robot. in the next stage, that is the standby server stage. After the recovery of the master server, the master server sends safety signal to the robot and standby server, that is attack disappear stage. Run to this stage, the process of the robot defenses DDoS attack has completed. The cloud robot will turn to the normal condition finally. The process of every condition is illustrated in Table 1.

3. Implementation of a gesture recognition cloud robot against DDoS attack to verify the policy

This work created a cloud gesture recognition robot against DDoS attack based on the scheme above. Considering we only need to verify the feasibility of the system. So, this work chose the simplest robot: manipulator. The robot control system runs in Jetson TX1. The complex model runs in master server is Deep CNN (Deep Convolutional Neural Network). The simple model runs is HMM (Hidden Markov Model). The DDoS detection method is "DDoS attack detection based on abnormal network flow with sliding window trending prediction".

3.1 Gesture recognition based on deep CNN in the master server

The convolution network is inspired by the animal visual nervous system. The eye is an imaging system that images through the pupil and the lens in the retina, which is the optical system of the vision. The retina is filled with a large number of photoreceptor cells, which converts light stimulation into nerve impulses, starting from these cells and entering the visual nervous system. Visual impulse is transmitted to the primary visual cortex of the brain through the visual pathway (Visual nerve > Optic chiasma > Optic tract > Optic radiology). From the beginning of the visual cortex, for image processing is layered, this layer is from V1 to V5 (V means Visual), a part of the specific information of each layer, such as the V1 of some edge texture response, V4 of some simple shapes are the order response, but the signal transmission is from V1 > V5, note that this is probably not a sequence of layers of simple connection, such as V2 and all other layers are connected.

For the input image X , the convolution neural network extracts and reduces the feature of the image by alternately connected convolution layer and lower sampling layer. In this paper, the feature graph of layer i in the network is represented by H_i ($H_0 = X$), and if the i layer is a convolution layer, then H_0 can be described as:

$$H_i = f(w_i \otimes H_{i-1} + b_i) \quad (21)$$

In formula 2-1, W_i represents the weight of the convolution kernel of the i layer in the convolution network. The characteristics of H_{i-1} layer W_i and the network $(i-1)$ convolution (" \otimes "), and then the offset vector sum b_i and current layer, finally through the nonlinear function $f(x)$ by incentive to get H_i . The feature graph H_i of the current layer.

The role of down sampling is to reduce the dimension of the feature graph, and to ensure that the feature is invariable to a certain extent. Therefore, the down sampling layer is usually located behind the volume layer. Then we use the specific down sampling method (such as Max pooling, average pooling and stochastic pooling) to sample the feature map (also called "pooling"), and finally get the H_i after sampling.

$$H_i = \text{subsampling}(H_{i-1}) \quad (22)$$

The input image X performs feature extraction and dimensionality reduction through multiple levels of convolution layer and down sampling layer. Finally, the feature expression is classified by the fully connected network, and the mapping from input image to category probability distribution Y is completed.

$$Y(m) = P(L = l_m | H_0; (W, b)) \quad (23)$$

In formula 23, M is an index for the label category. The training process of the convolution neural network is the process of minimizing the loss function $L(W, b)$ of the network. Negative Log likelihood (NLL) and Mean Square Error (MSE) are common loss functions in convolution neural networks.

$$NLL(W, b) = -\sum_{m=1}^{|Y|} \log Y(m) \quad (24)$$

$$MSE(W, b) = \frac{1}{|Y|} \sum_{m=1}^{|Y|} (Y(m) - \bar{Y}(m))^2 \quad (25)$$

At the same time, in order to reduce the over fitting of network parameters, L2-regularization is usually added to the final loss function $E(W, b)$:

$$MSE(W, b) = \frac{1}{|Y|} \sum_{m=1}^{|Y|} (Y(m) - \bar{Y}(m))^2 \quad (25)$$

$$E(W, b) = L(W, b) + \frac{\lambda}{2} W^T W \quad (26)$$

The weight decay λ in formula 26, which is used to control the over fitting strength of the whole network model. For training samples, the common way of training is to use iterative Stochastic Gradient Descent (SGD) to reduce the loss function by changing the weights of convolution kernel in convolution neural network and corresponding level vectors

$$W_i = W_i - \eta \frac{\partial E(W, b)}{\partial W_i} \quad (27)$$

$$b_i = b_i - \eta \frac{\partial E(W, b)}{\partial b_i} \quad (28)$$

In formula 27 and 28, the super parameter λ is the learning rate, which is used to control the speed of the gradient descent. After the iterative gradient descent training, the loss of the whole model tends to be stable, and the training is stopped. The values of the (W, b) in the convolution neural network are also determined by training and will no longer be updated before the network is transmitted to the network.

3.2 Gesture recognition based on deep HMM in the standby server

Hidden Markov model (Hidden Markov Model, referred to as HMM) is the simplest structure dynamic Bayesian network (dynamic Bayesian network), which is a famous directed graph model, mainly used for data modeling, is widely used in speech recognition, Natural Language Processing etc...

Hidden Markov model is a further development of Markov model. It looked at a series of general stochastic process state constantly changing time t state with q_t said, he can be N for turntable set $S = [s_1, s_2, \dots, s_n]$, the characteristic of Markov model is mainly expressed by "transfer probability". The probability of the latter state is determined by its previous state, namely $P_t[q_t / q_{t-1}, \dots, q_1]$. If this probability is determined only in the previous state, that is, $P_t[q_t / q_{t-1}]$, it is called the first order Markov process.

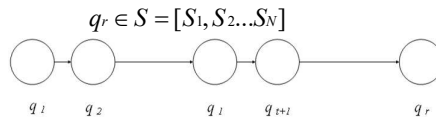


Figure 4. (a) Markov process

$$q_T \in S = [S_1, S_2, \dots, S_N]$$

$$O_T \in v = [V_1, V_2, \dots, V_N]$$

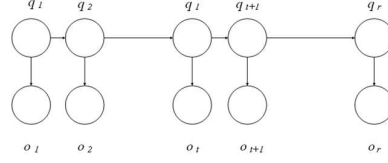


Figure 4. (b) Hidden Markov process

The arrow in the graph represents the dependency between variables. At any time, the value of observation variables depends only on state variables, that is, X_t is determined by y_t and has nothing to do with the values of other state variables and observation variables. At the same time, y_{t-1} state t at time t depends only on the time $t-1$, has nothing to do with the rest of the $n-2$ state, which is called Markov chain, namely, the state in the next time is only determined by the current state does not depend on any previous state. Based on this dependence, the joint probability distribution of all variables is:

$$P(x_1, y_1, \dots, x_n, y_n) = P(y_1)P(x_1 | y_1) \prod_{i=2}^n P(y_i | y_{i-1})P(x_i | y_i) \quad (29)$$

The characteristics of the hidden Markov process can be characterized by the following set of parameters:

(1) The probability that the transfer probability $a_{i,j} = \Pr[s_j | s_i]$ is transferred from the state i to the state j . Because there are a total of N possible states, $a_{i,j}$ has $N * N$ possible values, and they are expressed in matrices

$$A = [a_{i,j}] \text{ and } \sum_{j=1}^N a_{i,j} = 1 \quad (30)$$

(2) The probability of observing the probability $b_j(k) = \Pr[u_k(s_j)]$, that is, to produce the observation of u_k under the state s_j . If there is a possible M observation, the $b_j(k)$ consists of a $M * N$ matrix B .

$$B = [b_j(k)] \text{ and } \sum_{k=1}^M b_j(k) = 1 \quad (31)$$

(3) Initial state probability: the first state q_1 takes which one in $[S_1, S_2, \dots, S_N]$. It consists of $1 * N$ vector π , $\pi = [\pi_1, \pi_2, \pi_3, \dots, \pi_N]$..

$$\pi_i = \Pr[q_1 = S_i] \quad (32)$$

3.3 DDoS attack detection based on abnormal network flow with sliding window trending prediction

Distributed denial-of-service (DDoS) is a rapidly growing problem with the fast development of the Internet. There're multitude DDoS detection approaches, however, three major problems about DDoS attack detection appear in the big data environment. Firstly, to shorten the respond time of the DDoS attack detector, secondly, to reduce the required compute resources, and lastly, to achieve a high detection rate with low false alarm rate.

The author Jieren Cheng in this work proposed an abnormal network flow feature sequence prediction approach which could fit to be used as a DDoS attack detector in the big data environment [20] and solve aforementioned problems. As illustrated in Fig.5, it is the basic workflow of the DDoS detection method.

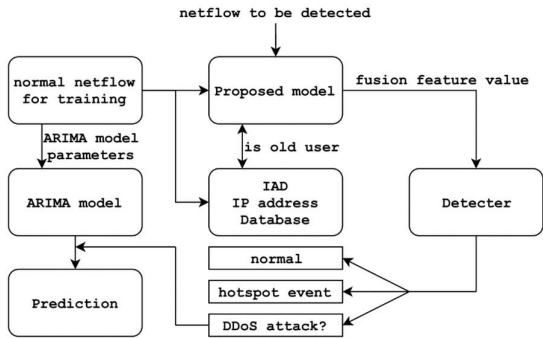


Figure 5. Basic workflow of the DDoS detection method

4 Experiment

According to the experiment scheme above. This work carried out experiments in real conditions. The processes including normal condition where there is no DDoS attack and the condition when the DDoS attack occurs.

Table 2. Gesture recognition results when there is no DDoS attack

Image of the video	Mask image	Binary image	Recognition result

4.1 Gesture recognition with master server in no DDoS attack condition in the experiment

When there is no DDoS attack, the cloud robot runs the gesture recognition in the master server. We built deep CNN model in the master server. There are five gestures in the preinstall: nothing, OK, peace, punch and stop. The recognition results are very accuracy, as illustrated in Table2.

As shown in the Table 2 above. The first line in the table shows the real image from the camera. The second column in the table shows mask images of the real image based on image segmentation. The third column shows binary images and the last line is the recognition result present by bar. From bars we can see that the recognition result of the deep CNN in the master server is accurate and positive. On the last row, I put a pen on the recognition area. From the mask image, the recognition is true and the result has high certainty.

4.2 Gesture recognition in robot itself when no DDoS attack occurs in the experiment

When there is DDoS attack, as illustrated in Fig.2, the master server sends warning signal to the standby server. Then the standby server will build communication with the robot. In this period, the robot will run the HMM model to recognize gesture. This work builds HMM model in the cloud robot. There are six gestures in the preinstall of HMM model: OK, peace, punch and stop and one. There is one more gesture type which need to be recognized in HMM model. This reflect the advantage of the non-deep learning: The non-deep learning method requires much few sample images, so there are more types of gesture can be recognized by this way. However, there are also shortcomings of these models. The accuracy is not as high as the deep learning method. And the certainty is not enough. As illustrated in Table 3, it is the probabilities of each gesture in recognition results of HMM. From the Table 3, we can see that the certainty is not so enough as deep CNN model. Despite the certainty is not so enough, we are still satisfied with this method. Because even the certainty is not enough, most of gesture recognition results are right. The correct rate is up to the level we can accept.

Table 3. Probabilities of each gesture in recognition results of HMM

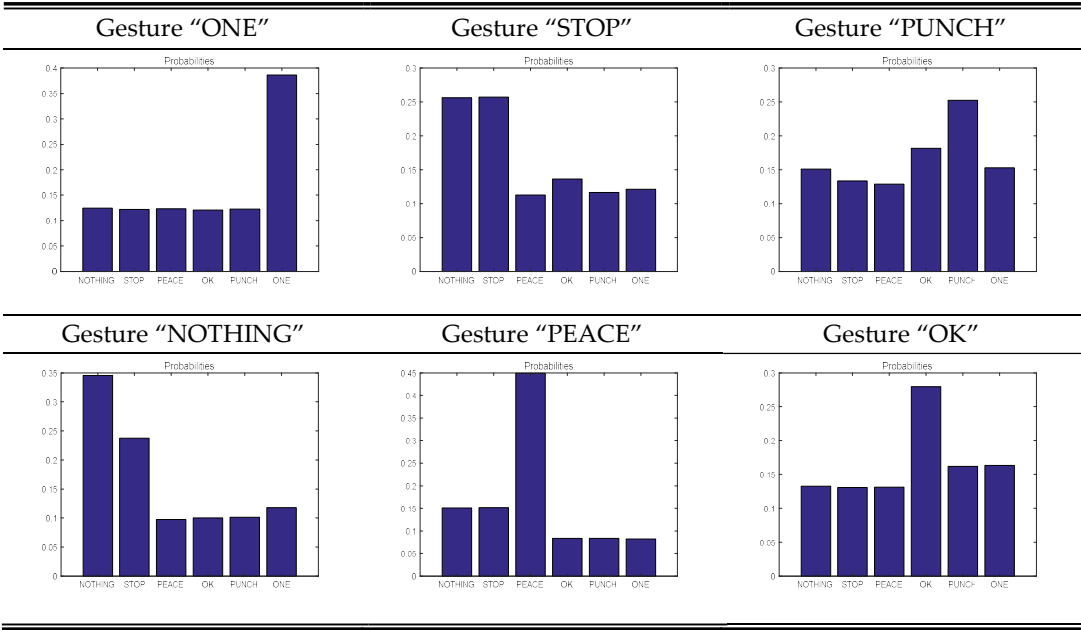


Table 3 shows the data of gesture recognition results. The results are right although the certainties of these results are not very high. So, when there is DDoS attack the HMM method can guarantee the right operation of robots.

4.3 DDoS detection experiment in the master server

To the DDoS detection method in the master server. We deploy the DDoS detection program proposed in 3.3 of the paper. in the master server. This work conducted experimental tests using network data with DDoS. Fig.6(a)- Fig.6(c) show the DDoS detection results in master server.

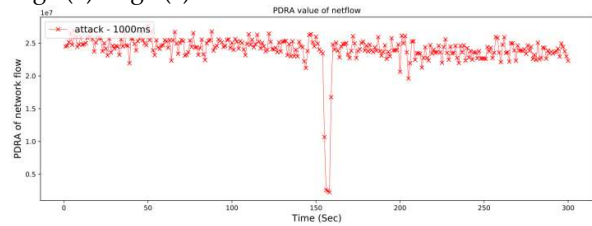


Figure 6. (a) PDRA values of the DDoS attack netflow

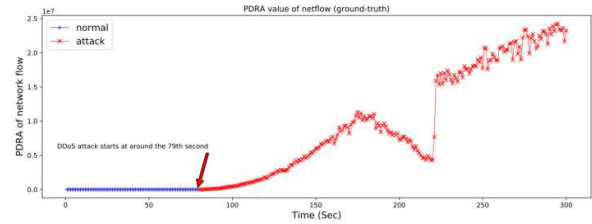


Figure 6. (b) PDRA values of the netflow combined with normal netflow and DDoS netflow

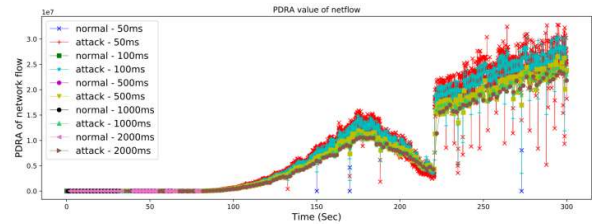


Figure 6. (c) PDRA value of DDoS Attack 2007 dataset with different Δt

Fig.5(a) shows PDRA values of the DDoS attack netflow. From the Fig.3(a) we can see that the PDRA value is enough high when the netflow is DDoS attack. Fig.5(b) shows PDRA values of the netflow combined with normal netflow and DDoS netflow. From the Fig.5(c) we can see that there is a sudden increase of the PDRA value when the DDoS attack occur. So, this method is effective. Fig3(c) is the detection effect of the proposed method compared with other popular methods. From this figure, we can see that the proposed method is excellent.

4.4 Realistic test scenario

As illustrated in Fig.7(a)-(c), we took some pictures of realistic test scenario. The gesture of “Peace” makes the robot steer, the gesture of “Punch” makes the robot grab things, the gesture of “Stop” makes the robot stop grabbing etc..

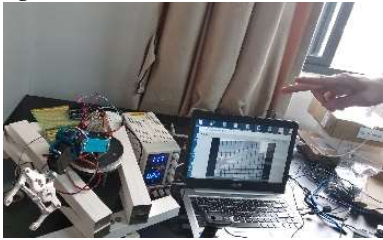


Figure 7. (a) Realistic test scenario of gesture “Peace”



Figure 7. (b) Realistic test scenario of gesture "Punch"

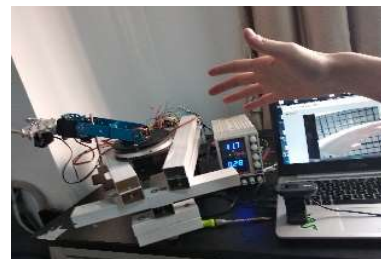


Figure 7. (c) Realistic test scenario of gesture "Stop"

From the results of the experiment. We can see that the method in this work is effective when there is DDoS attack. The gesture recognition robot gets accurate results from the master server. Even there is DDoS attack to the master server, the robot can work normally by running HMM in robot itself. This work also verified the DDoS detection method based on PDRA. It works pretty well in detect DDoS attack. So, the feasibility and effectiveness of the design of cloud robots against DDoS attack.

5. Conclusions

This work designed and implemented a cloud robot against DDoS attack. In particular, we proved DDoS attack obey Poisson distribution. The important parameter of backup interval is proposed in this work. Based on the scheme in this work, we used the deep CNN model, the HMM model and the PDRA feature of netflow to implement a safe gesture recognition robot. This robot proved that the security policy to defense DDoS attack for cloud robot.

Funding: This research was funded by [National Natural Science Foundation of China] grant number [61762033, 61702539], [The National Natural Science Foundation of Hainan] grant number [617048, 2018CXTD333], [Hainan University Doctor Start Fund Project] grant number [kyqd1328], [Hainan University Youth Fund Project] grant number [qnjj1444].

Author Contributions: Conceptualization, Boyi Liu; Data curation, Yijie Li, Xiangyan Tang and Jiquan Chen; Formal analysis, Boyi Liu; Funding acquisition, Xiangyan Tang; Methodology, Boyi Liu; Project administration, Jieren Cheng; Resources, Jieren Cheng; Software, Jiquan Chen; Visualization, Yijie Li; Writing – original draft, Boyi Liu; Writing – review & editing, Jieren Cheng.

References

1. Kuffner J J, Lavalley S M. Space-filling trees: A new perspective on incremental search for motion planning. In: International Conference on Intelligent Robots and Systems. pp. 2199-2206 (2011).
2. Waibel M, Beetz M, Civera J, et al. RoboEarth - A World Wide Web for Robots, IEEE Robotics and Automation Magazine 18(2), 69-82 (2011).
3. Ravi N, Mala T, Srinivasan M K, et al. Design and Implementation of VOD (Video on Demand) SaaS Framework for Android Platform on Cloud Environment. In: IEEE International Conference MDM 2, pp. 171-176 (2013).
4. Turnbull L, Samanta B. Cloud robotics: Formation control of a multi robot system utilizing cloud infrastructure, Southeastcon. In: 2013 Proceedings of IEEE. IEEE, pp. 1-4 (2013).

5. Kehoe B, Matsukawa A, Candido S, et al. Cloud-based robot grasping with the google object recognition engine. In: IEEE International Conference on Robotics and Automation. pp. 4263-4270 (2013).
6. Guizzo E. Robots with their heads in the clouds. IEEE Transactions on Spectrum 48(3), 16-18 (2011).
7. Furler L, Nagrath V, Malik A S, et al. An Auto-Operated Telepresence System for the Nao Humanoid Robot, Communication Systems and Network Technologies (CSNT). pp. 262-267 (2013).
8. Kamei K, Nishio S, Hagita N, et al. Cloud networked robotics. IEEE Network 26(3), 28-34 (2012).
9. Greene J. Our driverless dilemma. Science 352(6293), 1514-1515 (2016)
10. Singh S, Okun A, Jackson A. Artificial intelligence: Learning to play Go from scratch. Nature 550(7676), 336 (2017).
11. Ranjan N, Ghouse Z, Hiwrale N. A Multi-function Robot for Military Application. Imperial Journal of Interdisciplinary Research 3(3), (2017).
12. Sahi A, Lai D, Li Y, et al. An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment 5(), 6036-6048 (2017).
13. Rukavitsyn A, Borisenko K, Shorov A. Self-learning method for DDoS detection model in cloud computing. IEEE Young Researchers in Electrical and Electronic Engineering. pp. 544-547. (2017).
14. H. Luo, Z. Chen, J. Li, and A. V. Vasilakos. Preventing distributed denial-of-service flooding attacks with dynamic path identifiers. IEEE Transactions on Information Forensics and Security 12(8), 1801-1815 (2017).
15. D. Han, K. Bi, H. Liu, and J. Jia. A ddos attack detection system based on spark framework. Computer Science and Information Systems 14(3), 769-788 (2017).
16. Q. Yan, Q. Gong, and F. Yu. Effective software-defined networking controller scheduling method to mitigate DDoS attacks. Electronics Letters 53(7), 469-471 (2017).
17. Kehoe B, Patil S, Abbeel P, et al. A Survey of Research on Cloud Robotics and Automation. IEEE Transactions on Automation Science & Engineering 12(2), 398-409 (2015).
18. Haight F A. Handbook of Poisson Distribution[J]. Journal of the Royal Statistical Society 18(4), (1967).
19. Sheldon M R. Introduction to Probability Models (Eleventh Edition). New York-London: Academic Press, (2014).
20. Jieren Cheng, Ruomeng Xu, Xiangyan Tang etc. An Abnormal Network Flow Feature Sequence Prediction Approach for DDoS Attacks Detection in Big Data Environment. Computers, Materials & Continua 55(1), (2018).