

Security Threats to Artificial Intelligence-Driven Wireless Communication Systems 2021

Lead Guest Editor: Huaming Wu

Guest Editors: Xiaolong Xu, Kaitai Liang, Junqing Zhang, and Yuan Yuan





**Security Threats to Artificial Intelligence-
Driven Wireless Communication Systems 2021**

Security and Communication Networks

**Security Threats to Artificial
Intelligence-Driven Wireless
Communication Systems 2021**

Lead Guest Editor: Huaming Wu

Guest Editors: Xiaolong Xu, Kaitai Liang, Junqing
Zhang, and Yuan Yuan







Copyright © 2021 Hindawi Limited. All rights reserved.

This is a special issue published in "Security and Communication Networks." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Roberto Di Pietro, Saudi Arabia

Associate Editors

Jiankun Hu , Australia
Emanuele Maiorana , Italy
David Megias , Spain
Zheng Yan , China

Academic Editors

Saed Saleh Al Rabae , United Arab Emirates
Shadab Alam, Saudi Arabia
Goutham Reddy Alavalapati , USA
Jehad Ali , Republic of Korea
Jehad Ali, Saint Vincent and the Grenadines
Benjamin Aziz , United Kingdom
Taimur Bakhshi , United Kingdom
Spiridon Bakiras , Qatar
Musa Balta, Turkey
Jin Wook Byun , Republic of Korea
Bruno Carpentieri , Italy
Luigi Catuogno , Italy
Ricardo Chaves , Portugal
Chien-Ming Chen , China
Tom Chen , United Kingdom
Stelvio Cimato , Italy
Vincenzo Conti , Italy
Luigi Coppolino , Italy
Salvatore D'Antonio , Italy
Juhriyansyah Dalle, Indonesia
Alfredo De Santis, Italy
Angel M. Del Rey , Spain
Roberto Di Pietro , France
Wenxiu Ding , China
Nicola Dragoni , Denmark
Wei Feng , China
Carmen Fernandez-Gago, Spain
AnMin Fu , China
Clemente Galdi , Italy
Dimitrios Geneiatakis , Italy
Muhammad A. Gondal , Oman
Francesco Gringoli , Italy
Biao Han , China
Jinguang Han , China
Khizar Hayat, Oman
Azeem Irshad, Pakistan

M.A. Jabbar , India
Minho Jo , Republic of Korea
Arijit Karati , Taiwan
ASM Kayes , Australia
Farrukh Aslam Khan , Saudi Arabia
Fazlullah Khan , Pakistan
Kiseon Kim , Republic of Korea
Mehmet Zeki Konyar, Turkey
Sanjeev Kumar, USA
Hyun Kwon, Republic of Korea
Maryline Laurent , France
Jegatha Deborah Lazarus , India
Huaizhi Li , USA
Jiguo Li , China
Xueqin Liang, Finland
Zhe Liu, Canada
Guangchi Liu , USA
Flavio Lombardi , Italy
Yang Lu, China
Vincente Martin, Spain
Weizhi Meng , Denmark
Andrea Michienzi , Italy
Laura Mongioi , Italy
Raul Monroy , Mexico
Naghme Moradpoor , United Kingdom
Leonardo Mostarda , Italy
Mohamed Nassar , Lebanon
Qiang Ni, United Kingdom
Mahmood Niazi , Saudi Arabia
Vincent O. Nyangaresi, Kenya
Lu Ou , China
Hyun-A Park, Republic of Korea
A. Peinado , Spain
Gerardo Pelosi , Italy
Gregorio Martinez Perez , Spain
Pedro Peris-Lopez , Spain
Carla Ràfols, Germany
Francesco Regazzoni, Switzerland
Abdalhossein Rezai , Iran
Helena Rifà-Pous , Spain
Arun Kumar Sangaiah, India
Nadeem Sarwar, Pakistan
Neetesh Saxena, United Kingdom
Savio Sciancalepore , The Netherlands



De Rosal Ignatius Moses Setiadi ,
Indonesia
Wenbo Shi, China
Ghanshyam Singh , South Africa
Vasco Soares, Portugal
Salvatore Sorce , Italy
Abdulhamit Subasi, Saudi Arabia
Zhiyuan Tan , United Kingdom
Keke Tang , China
Je Sen Teh , Australia
Bohui Wang, China
Guojun Wang, China
Jinwei Wang , China
Qichun Wang , China
Hu Xiong , China
Chang Xu , China
Xuehu Yan , China
Anjia Yang , China
Jiachen Yang , China
Yu Yao , China
Yinghui Ye, China
Kuo-Hui Yeh , Taiwan
Yong Yu , China
Xiaohui Yuan , USA
Sherali Zeadally, USA
Leo Y. Zhang, Australia
Tao Zhang, China
Youwen Zhu , China
Zhengyu Zhu , China

Contents





The Optimal Carrier-Secret Ratio for Wireless Covert Channels Based on Constellation Shaping Modulation

Sen Qiao , Guangjie Liu , Xiaopeng Ji, and Weiwei Liu
Research Article (15 pages), Article ID 6919530, Volume 2021 (2021)

FNet: A Two-Stream Model for Detecting Adversarial Attacks against 5G-Based Deep Learning Services

Guangquan Xu, Guofeng Feng , Litao Jiao, Meiqi Feng, Xi Zheng, and Jian Liu 
Research Article (10 pages), Article ID 5395705, Volume 2021 (2021)

An Enhanced Visual Attention Siamese Network That Updates Template Features Online

Wenqiu Zhu , Guang Zou , Qiang Liu , and Zhigao Zeng 
Research Article (19 pages), Article ID 9719745, Volume 2021 (2021)

Reinforcement Learning for Security-Aware Workflow Application Scheduling in Mobile Edge Computing

Binbin Huang , Yuanyuan Xiang, Dongjin Yu , Jiaojiao Wang, Zhongjin Li, and Shangguang Wang
Research Article (13 pages), Article ID 5532410, Volume 2021 (2021)

Research Article

The Optimal Carrier-Secret Ratio for Wireless Covert Channels Based on Constellation Shaping Modulation

Sen Qiao ¹, Guangjie Liu ¹, Xiaopeng Ji,¹ and Weiwei Liu²

¹School of Electrical and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

²School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Guangjie Liu; gjieliu@gmail.com

Received 26 April 2021; Revised 13 July 2021; Accepted 2 November 2021; Published 3 December 2021

Academic Editor: Vijayakumar Pandi

Copyright © 2021 Sen Qiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless covert communication is an emerging communication technique that prevents eavesdropping. This paper considers the bit error ratio (BER) problem of covert communication based on constellation shaping modulation (CSM). The impact of carrier-secret ratio (CSR) on BER is studied and the approximate solution of optimal CSR is obtained. Then, we extended the conclusion to typical communication scenarios with one and more relays where the undetectability and reliability were analyzed and inspected. It is proved that there also exists the optimal CSR in scenarios with relays. Additionally, it is found that the undetectability under the constraints of constant total power depends on the eavesdropper's position, and we found an undetectability deterioration area (UDA) in the scenario of relays. Simulation results show the existence of optimal CSR and its impact on transmission performance.

1. Introduction

Due to the openness of wireless channels, wireless communication systems are extremely vulnerable to attacks, counterfeiting, and eavesdropping. With the advent of the Internet of Things (IoT) era, a large number of smart devices are connected and controlled to meet various requirements. Hence, it is very important to safeguard the information against security breaches and to ensure the privacy of communication.

To ensure the security of personal information, some efficient anonymous authentication schemes have been proposed to adapt to different scenarios [1–3]. Multiple technologies are integrated to promote the realization of the Internet of Things (IoT), including wireless sensor networks (WSNs), radio frequency identification (RFID), machine to machine (M2M), and low-power personal area networks (PANs) [4]. To ensure the high efficiency of information channels from the clients to the cloud server, Ahmad proposed a new variant of the optimistic concurrency control protocol to avoid using the upstream

communication channel all the time [5]. The most important precondition of secure and reliable group communication is an efficient group key distribution. Azees and Vijayakumar proposed a computationally efficient group key distribution scheme for secure group communication based on bilinear pairing [6]. As the number of devices connected to supporting platforms continues to increase, some proper means for access control are demanding, such as authentication and authorization method [7, 8], image watermarking [9], and cloud computing [10].

However, the challenges of information security and privacy are not limited to the above. Count on the rapid growth of telecommunication field new challenges arises [11]. Eavesdroppers can intercept the wireless communication signals and try to get the communication contents, which poses a great threat to the security and privacy of the communication. In order to ameliorate the undetectability of private information, information hiding technology gradually becomes necessary. As a branch of modern information hiding technology in the field of wireless communication, wireless covert channels hide the transmission

process of information that needs to be kept secret in the process of normal wireless communication. Even if an eavesdropper intercepts the communication signal, it cannot be distinguished from normal wireless communication.

Based on the ubiquitous channel noise phenomenon, modulation-type wireless covert communication modulates the secret information into an artificial noise signal, which is superimposed on the normal communication signal. It is the most widely used physical-layer wireless covert communication at present. The basic theory and performance limit of the covert communication in AWGN channels are discussed in Reference [12]. It is indicated that at most $O(\sqrt{n})$ bits can be transmitted to the receiver reliably without being detected by the detector.

1.1. Motivations. In modulation-type wireless covert communication, the bit error ratio (BER) of covert information is usually much greater than that of the carrier signal. The problems of BER are always solved by means of coding or increasing the power of covert signals. Yet, the difficulty of decryption and the transmission rate of the covert messages will deteriorate with encoding. By means of increasing the transmission power, undetectability will deteriorate [13]. In Reference [14], relays are proposed to increase the power of covert signal received. But it has not been simulated with specific modulation methods, and no one has considered whether the optimal power ratio of covert signal exists. We plan to research the undetectability and reliability of wireless covert communication in the scenario of relays based on a specific modulation method. And consider whether there exists an optimal carrier-secret ratio (CSR), which can ameliorate the reliability of covert communication under the premise of meeting the requirements of undetectability.

1.2. Contributions. The contributions of our work are as follows:

- (1) We investigated the relationship between BER, CSR, and SNR in wireless covert channels with constellation shaping modulation. We obtained the approximate solution of optimal CSR and extended it to several scenarios with relays. With the approximate solution of optimal CSR, the process of searching for an actual optimal CSR can be accelerated when some optimization algorithms are adopted such as gradient descent and conjugate gradient.
- (2) We found an undetectability deterioration area (UDA) in the scenario of one relay and two relays, and the undetectability deteriorates when an eavesdropper is in it. The UDA can be used to avoid the deterioration of undetectability with an improper set of relays. Otherwise, eavesdroppers can detect in the UDA to improve detection efficiency.

The remainder of this paper is organized as follows: in the next section, some background including wireless covert channel with dirty constellation and wireless covert channel

with constellation shaping modulation is introduced; in Section 3, we introduced the basis of our scheme including the classic system model and binary hypothesis testing; in Section 4, the relationship between BER, CSR, and SNR in wireless covert channels with constellation shaping modulation is investigated. The approximate solution of optimal CSR is obtained and extended to several scenarios of relays, in which undetectability deterioration areas (UDAs) were found and analyzed; Section 5 gives the experimental results on undetectability and reliability; and finally, Section 6 concludes the whole paper.

2. Related Works

2.1. Background. Wireless covert communication mainly involves three factors of inspection: undetectability, reliability, and communication rate. At present, there is no special detection work to measure undetectability for noisy wireless covert communication. References [13, 15, 16] take KL divergence between residual and ambient noise as parameters to inspect undetectability. Reference [17] inspects the undetectability with KS distance between residual and ambient noise.

Reliability refers to the ability of wireless covert communication to resist channel interference. Channel interference may come from the natural fading of the channel, or from the jammer. To resist channel interference, multihop relaying is a frequently used method [18]. Reference [19] evaluated and optimized the covert communications by designing the parameters of the multihop network, including the coding rates, transmit power, and required number of hops.

The researchers further analyzed the covert communication capacity of multiple scenarios with multiple unfavorable factors to the eavesdropper, including three aspects of the transmitter [20, 21], receiver [22–24], and additional nodes [25–28]. The methods of covert communication include artificial additional signal noise, artificial coding domain error, insertion of additional signal band, etc. The research results in this field have also been further extended to other communication scenarios such as relay communication [14, 29], multiantenna [30, 31], and broadcast communication [32–34].

2.2. Wireless Covert Channel with Dirty Constellation. In the wireless covert channel with dirty constellation (WCC-DC), the secret message bits can be transmitted as the constellation error of the normal signal in order to reduce the suspicion by all uninformed detectors.

The framework of a wireless covert channel with dirty constellation is shown in Figure 1(a). The wireless covert channel is implemented on the wireless communication physical layer with OFDM structure. The transmitter divides all OFDM subcarriers into secret subcarriers and normal subcarriers. On the secret subcarrier, the carrier information is modulated in QPSK to obtain the carrier signal, and then the covert signal modulated by QPSK is superposed on the carrier signal. The covert constellation points are rotated at a

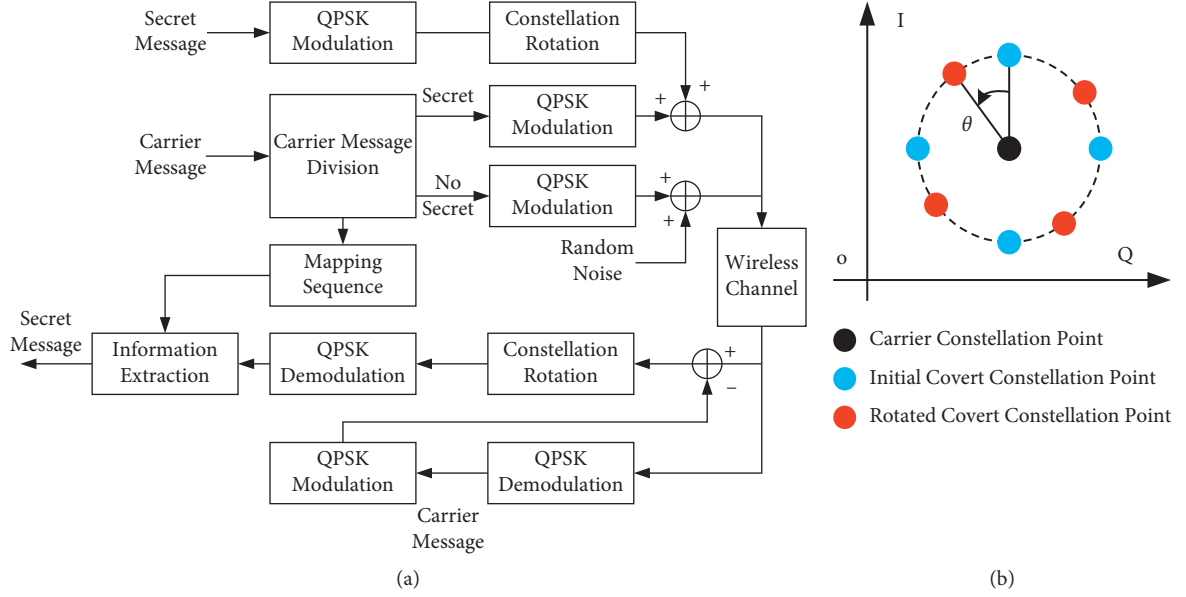


FIGURE 1: The schematic diagram of WCC-DC: (a) the framework of WCC-DC and (b) rotation of covert constellation.

certain angle around normal constellation points, as shown in Figure 1(b). On the normal subcarrier, carrier information is modulated in QPSK to obtain the carrier signal, and then the random noise is superimposed. The purpose of the rotation of signal-loaded signals and the superimposition of random noise is to remove the regularity of secret signals in the constellation. The subcarrier partition results of the wireless covert channel should be shared between the transmitter and the receiver.

However, the wireless covert channel with dirty constellation has a high BER when the power of covert signal is low. When we increase the power of covert signal, the undetectability of covert communication deteriorates. Therefore, Cao et al. [35] proposed a covert communication method based on constellation shaping modulation (WCC-CSM).

2.3. Wireless Covert Channel with Constellation Shaping Modulation. The general framework for the wireless communication system with constellation shaping modulation is demonstrated in Figure 2. We suppose that each subcarrier m_c of the OFDM wireless communication is modulated by QPSK. In the proposed scheme, we can use all subcarriers to establish the wireless covert communication. With constellation shaping modulation, the secret information m_s is modulated into an artificial noise signal S_s . Then, the artificial noise signal S_s is superimposed on the carrier signal S_c to generate the secret subcarrier S_{ct} .

To generate the secret artificial noise signal S_s , the cumulative distribution function (CDF) F_{CDF} of noise is estimated with the reference channel noise data S_0 .

The secret information is denoted by $m_s = (m_{s,1}, m_{s,2}, \dots, m_{s,N})$, and the artificial noise signals S_s are divided into I/Q vectors, which are denoted by $x_s^I + j \cdot x_s^Q$. Here, x_s^I is the I vector of artificial noise signals

denoted by $x_s^I = [x_{s,1}^I, x_{s,2}^I, x_{s,3}^I, \dots, x_{s,N}^I]$, and x_s^Q is the Q vector denoted by $x_s^Q = [x_{s,1}^Q, x_{s,2}^Q, x_{s,3}^Q, \dots, x_{s,N}^Q]$. The constellation shaping modulation function is defined as

$$F_{\text{SMF}}(m_s) = S_s = x_s^I + j \cdot x_s^Q. \quad (1)$$

For shaping modulation, the transmitter firstly transforms the secret information m_s into continuous variables d_i , and then d_i are mapping to artificial noise signal S_s with CDF of the reference channel noise S_{normal} .

The transform function of d_i is defined as follows:

$$d_i = \frac{m_{s,i} + r}{2}. \quad (2)$$

We denote r as a random number distributed in the interval $(0, 1)$. And the mapping function which transforms m_s into S_s is defined as

$$S_s = F_{\text{CDF}}^{-1}(d_i). \quad (3)$$

The mapping function F_{CDF}^{-1} is the inverse function of cumulative distribution function of S_{normal} . Then, the artificial noise signal S_s is superimposed on carrier signal S_c to generate secret subcarrier S_{ct} .

The received secret subcarrier is denoted by \hat{S}_{ct} . The I/Q vectors of secret subcarrier \hat{S}_{ct} are denoted by $\hat{x}_{ct}^I + j \cdot \hat{x}_{ct}^Q$, the subcarrier \hat{m}_c can be demodulated with QPSK:

$$\hat{m}_c = F_{\text{de-QPSK}}(\hat{x}_{ct}^I + j \cdot \hat{x}_{ct}^Q), \quad (4)$$

and the subcarrier \hat{m}_c will be modulated by QPSK again to acquire the ideal the subcarrier \hat{S}_c :

$$\hat{S}_c = F_{\text{QPSK}}(\hat{m}_c) = \hat{x}_c^I + j \cdot \hat{x}_c^Q. \quad (5)$$

We denote $\hat{x}_c^I + j \cdot \hat{x}_c^Q$ as the I/Q vectors of \hat{S}_c . The receiver can obtain the ideal subcarrier $\hat{x}_c^I + j \cdot \hat{x}_c^Q$, and then the residual signal (i.e., artificial noise \hat{S}_s) can be extracted with

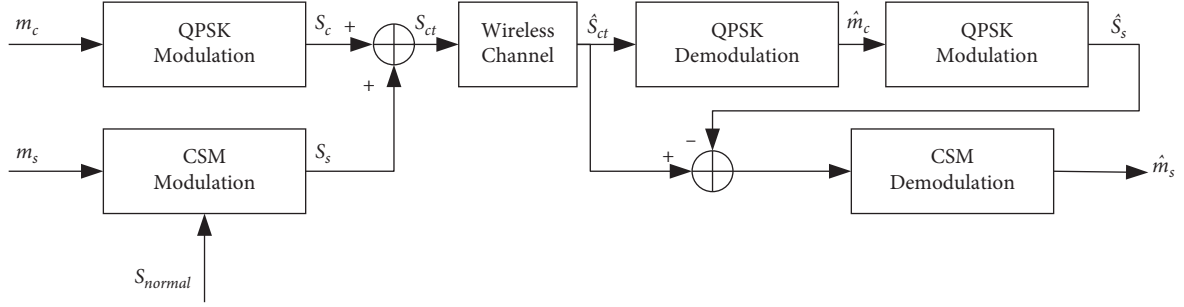


FIGURE 2: The framework of WCC-CSM wireless covert channel.

$$\widehat{S}_s = \widehat{S}_{ct} - \widehat{S}_c = (\widehat{x}_{ct}^I - \widehat{x}_c^I) + j \cdot (\widehat{x}_{ct}^Q - \widehat{x}_c^Q). \quad (6)$$

We denote the I/Q vectors of artificial noise \widehat{S}_s as $\widehat{x}_s^I + j \cdot \widehat{x}_s^Q$. The artificial noise \widehat{S}_s can be transformed into \widehat{d}_i by the cumulative distribution function F_{CDF} with

$$\widehat{d}_i = F_{\text{CDF}}(\widehat{S}_s). \quad (7)$$

The receiver can demodulate the secret information \widehat{m}_s by the covert demodulation constellation (CDC), which is illustrated in Figure 3.

The four black points are the ideal constellation points; the red regions are the distribution areas of secret subcarrier with artificial noise. The function of covert demodulation constellation is denoted by $F_{\text{CDC}}(\cdot)$:

$$\widehat{m}_s = F_{\text{CDC}}(\widehat{d}_i). \quad (8)$$

3. Basis of Our Scheme

3.1. System Model. Similar to the famous Alice–Bob model [36], the standard wireless covert channel system model includes the transmitter (i.e., Alice), the receiver (i.e., Bob), and the detector (i.e., Willie).

Willie observes the channel to detect whether Alice transmits or not. Willie’s probability of detection error consists of two components: the probability of missed detection and the probability of false alarm.

The literature as seen in the aforementioned works only mentioned the impact of finite samples (i.e., finite $m[i]$) on the detection performance at Willie. It is numerically shown that with noise uncertainty at Willie, there may exist an optimal number of samples that maximize the communication rate subject to $\xi \geq 1 - \varepsilon$, where ξ is the sum of P_F (i.e., false alarm rate) and P_M (i.e., miss detection rate) at Willie and ε is an arbitrarily small number. We define $0 < \varepsilon \leq 1$ as the maximum acceptable detection rate of Willie.

3.2. Binary Hypothesis Testing at Willie. According to the system model shown in Figure 4, the performance elements of the wireless covert channel mainly include two aspects: undetectability and reliability.

In communication, Alice totally transmits n symbols to Bob. We denote the finite block as $m[i] (i \in [1, n])$, which consists of normal information $m_c[i]$ and secret

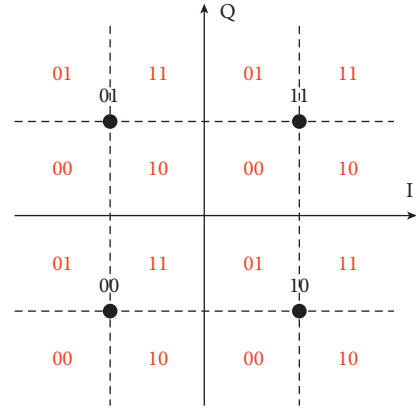


FIGURE 3: Covert demodulation constellation of WCC-CSM.

information $m_s[i]$, while Willie is passively collecting $m[i]$ observations on Alice’s transmission in order to detect the presence of her secret information (i.e., whether Alice is transmitting secret information). We denote the AWGN at Bob and Willie as $r[i] \sim \text{CN}(0, \sigma_w^2)$. The received signal at Willie for each signal symbol is given by

$$y_w[i] = m[i] + r[i]. \quad (9)$$

The main purpose of Willie is to confirm whether Alice transmits or not. We define two hypotheses, H_0 and H_1 , to distinguish these two cases:

$$\begin{cases} H_0: m[i] = m_c[i], \\ H_1: m[i] = m_c[i] + m_s[i]. \end{cases} \quad (10)$$

H_0 denotes the null hypothesis, where Alice is not transmitting secret information. H_1 denotes the alternative hypothesis, where Alice is transmitting secret information. In the covert communication, the ultimate goal of Willie is to minimize the total error rate (i.e., ξ). We denote T and F as binary decisions that infer whether Alice is transmitting or not. The false alarm rate and miss detection rate are given by

$$\begin{cases} P_F = P_r(T|H_0), \\ P_M = P_r(F|H_1). \end{cases} \quad (11)$$

Suppose Willie performs the optimal detect. Following Pinsker’s inequality [37, 38]

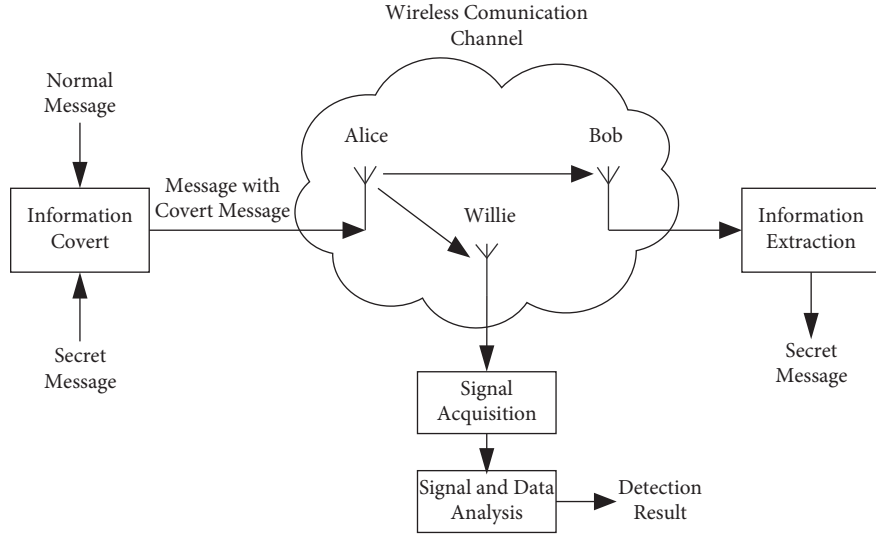


FIGURE 4: The framework of wireless covert communication.

$$P_F + P_M \geq 1 - \sqrt{\frac{1}{2} D(P_0 \| P_1)}, \quad (12)$$

where relative entropy $D(P_0 \| P_1)$ (also called KL divergence) is defined as follows:

$$D(P_0 \| P_1) = \int_n p_0(x) \ln \frac{p_0(x)}{p_1(x)} dx, \quad (13)$$

where n is the value range of x . We denote $p_1(x)$ as the distribution of a sequence $m[i]$, which detected by Willie, and $p_0(x)$ denotes the distribution of a sequence $m_c[i]$, which detected when Alice is not transmitting.

The KL divergence is always used to calculate the correlation between distributions. Except KL divergence, we can also use KS distance (also called Kolmogorov–Smirnov statistic) to calculate the distance between distributions. The KS distance is defined as follows:

$$D_{KS} = \max |F_1(x) - F_0(x)|. \quad (14)$$

$m_1[i]$ and $m_0[i]$ are both divided into K bins. The number of the elements in $m_1[i]$ and $m_0[i]$ are denoted by $h_1(j)$ and $h_0(j)$, $j \in (1, 2, \dots, K)$. The cumulative distribution functions of $m_1[i]$ and $m_0[i]$ are defined as follows:

$$\begin{aligned} F_1(x) &= \frac{\sum_{j=1}^x h_1(j)}{n}, \\ F_0(x) &= \frac{\sum_{j=1}^x h_0(j)}{n}. \end{aligned} \quad (15)$$

Willie always set threshold Γ of KL divergence and KS distance to judge whether there is communication.

$$\begin{cases} \Pr(T|D(P_0 \| P_1) > \Gamma \| D_{KS} > \Gamma), \\ \Pr(F|D(P_0 \| P_1) < \Gamma \| D_{KS} < \Gamma). \end{cases} \quad (16)$$

To measure the reliability of the communication, we denote BER as follows:

$$P_e = \frac{n_{\text{error}}}{n}, \quad (17)$$

where n_{error} is the number of error symbols.

4. Optimal Carrier-Secret Ratio for WCC-CSM

4.1. Classic Scenario. After the secret subcarrier \hat{S}_{ct} transmitting through the wireless channel, the receiver can get the output of the slow-fading channel as $\hat{S} = [\hat{S}_{ct}(1), \hat{S}_{ct}(2), \dots, \hat{S}_{ct}(N)]$ with

$$\hat{S}_{ct}[i] = \sqrt{P_t} \cdot \frac{\sqrt{\lambda_0}}{(\sqrt{d_{tr}})^\alpha} \cdot h_A \cdot S_{ct}[i] + n_A[i], \quad (18)$$

where P_t denotes the transmission energy of transmitter; λ_0 denotes the wavelength of the signal; h_A denotes the complex baseband equivalent channel coefficient of the main channel between the transmitter (i.e., Alice) and receiver (i.e., Bob); and n_A denotes zero-mean circularly symmetric complex Gaussian noise. And we also have two real variables d_{tr} and $\alpha \in R$ that denote the distance and path-loss exponent of the channel between Alice and Bob, respectively. The path-loss exponent α takes a value between 2 and 4. In free space, microwave transmission has path-loss exponent $\alpha = 2$.

The detector (i.e., Willie) can receive the output as $\hat{y}_{\text{willie}} = [\hat{y}_{\text{willie}}(1), \hat{y}_{\text{willie}}(2), \dots, \hat{y}_{\text{willie}}(N)]$ with

$$\hat{y}_{\text{willie}}[i] = \sqrt{P_t} \cdot \frac{\sqrt{\lambda_0}}{(\sqrt{d_{\text{willie}}})^2} \cdot h_W \cdot S_{ct}[i] + n_W[i]. \quad (19)$$

h_W denotes the complex baseband equivalent channel coefficient of the main channel between the transmitter (i.e., Alice) and detector (i.e., Willie); d_{willie} denotes the distance exponent between Alice and Willie. The noise n_W also follows a zero-mean circularly symmetric complex Gaussian distribution.

Theorem 1. *The undetectability of wireless covert communication deteriorates with the increase of CSR.*

Proof. As is mentioned above, the probability of detection error must satisfy a lower boundary of

$$P_F + P_M \geq 1 - \sqrt{\frac{1}{2}D(P_0\|P_1)}. \quad (20)$$

Considering equations (13) and (20) jointly, we can obtain the lowest boundary is at the lowest ‘‘KL divergence.’’ We denote $p_1(x)$ as the distribution of residual signal \widehat{S}_s , which detected by Willie, and $p_0(x)$ denotes the distribution of the reference channel noise S_0 . The artificial noise S_s is the mapping of $S_0 \sim N(0, \sigma_\omega^2)$; thus, we can obtain the residual signal $\widehat{S}_s \sim N(0, \sigma_\omega^2) \sim (0, P + \sigma_\omega^2)$. P denotes the energy of the signal, which received by Bob or Willie. The ‘‘KL divergence’’ of S_0 and \widehat{S}_s can be expressed as

$$\begin{aligned} D(S_0\|\widehat{S}_s) &= \int_n p_0(x) \ln \frac{p_0(x)}{p_1(x)} dx, \\ &= \int_n \frac{1}{\sqrt{2\pi}\sigma_\omega} e^{-x^2/\sigma_\omega^2} \cdot \ln\left(\frac{\sigma_s}{\sigma_\omega} \cdot e^{x^2/\sigma_\omega^2 - x^2/\sigma_s^2}\right) dx \\ &= \int_n \frac{1}{\sqrt{2\pi}\sigma_\omega} e^{-x^2/\sigma_\omega^2} \cdot \left(\ln\left(\frac{\sqrt{P + \sigma_\omega^2}}{\sigma_\omega}\right) + \frac{x^2 \cdot P}{(P + \sigma_\omega^2) \cdot \sigma_\omega^2}\right) dx \\ &= \frac{1}{2} \left(\ln \frac{P + \sigma_\omega^2}{\sigma_\omega^2} - \frac{P}{P + \sigma_\omega^2}\right) \end{aligned} \quad (21)$$

As is illustrated in equation (21), the ‘‘KL divergence’’ increases with the increase of P . Even the zero-mean circularly symmetric complex Gaussian noise with the same variance has different distributions. In order to calculate the ‘‘KL divergence’’, we divide both S_0 and \widehat{S}_s into K bins, the probability in j bins are denoted by $P_{S_0}(j)$ and $P_{S_{res}}(j)$, $j \in (1, 2, \dots, K)$. The expression (21) can be expressed as

$$D(S_0\|\widehat{S}_s) = \int_n P_{S_0}(x) \ln \frac{P_{S_0}(x)}{P_{S_{res}}(x)} dx. \quad (22)$$

With the increase of K , $P_{S_0}(j)$ and $P_{S_{res}}(j)$ will approach $p_0(x)$ and $p_1(x)$. However, if the K is too great, the P_F will be great. If the K is too small, the P_M will be great. We need to choose a suitable value of K . In this paper, we set $K=100$.

P_{Willie} denotes the signal energy detected by Willie. P_{ideal} denotes the ideal signal energy. The ‘‘KL divergence’’ can be expressed as

$$\begin{aligned} D(S_0\|\widehat{S}_s) &= \frac{1}{2} \left(\ln \frac{P + \sigma_\omega^2}{\sigma_\omega^2} - \frac{P}{P + \sigma_\omega^2}\right) \\ &= \frac{1}{2} \left(\ln \frac{P_{\text{Willie}} - P_{\text{ideal}}}{\sigma_\omega^2} - \frac{P_{\text{Willie}} - P_{\text{ideal}} - \sigma_\omega^2}{P_{\text{Willie}} - P_{\text{ideal}}}\right). \end{aligned} \quad (23)$$

ΔP denotes $P_{\text{Willie}} - P_{\text{ideal}}$. Taking the partial derivative with respect to ΔP , equation (23) can be expressed as

$$\frac{\partial D(S_0\|\widehat{S}_s)}{\partial \Delta P} = \frac{1}{2} \left(\frac{\Delta P - \sigma_\omega^2}{\Delta P^2}\right). \quad (24)$$

ΔP can be expressed as

$$\Delta P \approx \left(1 - \frac{1}{\text{CSR}}\right) \cdot P_{\text{Willie}} + \sigma_\omega^2. \quad (25)$$

Analysis: Considering expression equations (24) and (25) jointly, we can get $\partial D(S_0\|\widehat{S}_s)/\partial \Delta P > 0$. The KL divergence increases with the increase of ΔP . With the P_{Willie} unchanged, ΔP increases with the increase of CSR. Hence, the KL divergence increases with the increase of CSR.

The KL divergence increases with the increase of P_{Willie} . When the KL divergence is greater than Γ , Willie judges there is covert communication. When the KL divergence equals to Γ , the corresponding SNR_0 can be expressed as

$$\text{SNR}_0 = P_0 \cdot \frac{\lambda_0 \cdot h_0^2}{d_0^2}. \quad (26)$$

If $\text{SNR} < \text{SNR}_0$, the covert communication will not be detected. When the transmission power is constant, the threshold detection distance d_0 can be illustrated in Figure 5. The purple dotted circle is the equipower line in which the covert communication will not be detected with ‘‘KL divergence.’’

The probability of undetected P_{ud} can be expressed as

$$P_{ud} = \Pr \left\{ P_t \cdot \frac{\lambda_0 \cdot h_W^2}{d_W^2} < \text{SNR}_0 \right\}. \quad (27)$$

□

Theorem 2. *There exists an optimal ratio between the carrier signal and secret signal no matter what the value of SNR is. The BER minimizes at the optimal ratio.*

Proof. The reliability of the system is inspected by the BER. The BER of QPSK is

$$P_{\text{eQPSK}} = \frac{1}{2} \text{erfc}(\sqrt{r}). \quad (28)$$

$\text{erfc}(\cdot)$ denotes the Gauss error function, r denotes the signal-noise ratio. The BER of covert communication is denoted as P_{ecov} , which can be expressed as

$$\begin{aligned} P_{\text{ecov}} &= 1 - [1 - P_{e,mc}] \cdot [1 - P_{e,sc}] \\ &= 1 - \left[1 - \text{erfc}\left(\sqrt{\frac{\text{SNR} \cdot \text{CSR}}{\text{SNR} + \text{CSR} + 1}}\right)\right] \left[1 - \text{erfc}\left(\sqrt{\frac{\text{SNR}}{\text{CSR} + 1}}\right)\right]. \end{aligned} \quad (29)$$

SNR denotes the signal-noise ratio of \widehat{S}_{ct} to S_0 , CSR denotes the carrier-secret ratio of S_c to S_s . Considering the range of SNR and CSR, the expression (29) can be approximated as

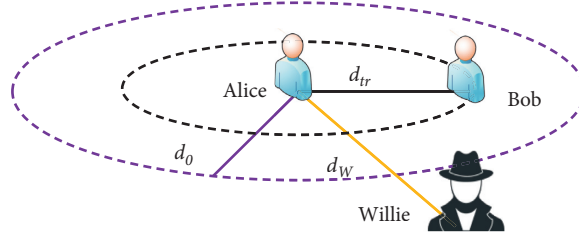


FIGURE 5: Location diagram of Alice, Bob, and Willie.

$$P_{\text{ecov}} = 1 - [1 - P_{e,mc}] \cdot [1 - P_{e,sc}] \approx 1 - \text{erf}(\sqrt{0.75 \times \text{CSR}}) \cdot \text{erf}\left(\sqrt{\frac{\text{SNR}}{\text{CSR}}}\right). \quad (30)$$

Taking the partial derivative with respect to CSR, the equation (30) can be expressed as

$$\frac{\partial P_{\text{ecov}}}{\partial \text{CSR}} = -\left[\left(\frac{1}{\sqrt{\pi}} \cdot e^{-0.75\text{CSR}} \cdot \frac{\sqrt{0.75}}{\sqrt{\text{CSR}}} \cdot \text{erf}\left(\sqrt{\frac{\text{SNR}}{0.75\text{CSR}}}\right)\right) - \left(\text{erf}(\sqrt{\text{CSR}}) \cdot \frac{1}{\sqrt{\pi}} \cdot e^{-\text{SNR}/\text{CSR}} \cdot \sqrt{\text{SNR}} \cdot \text{CSR}^{-3/2}\right)\right]. \quad (31)$$

Analysis: Let the $\partial P_{\text{ecov}}/\partial \text{CSR}$ equals zeros. We can obtain

$$\begin{aligned} \frac{1}{\sqrt{\pi}} \cdot e^{-0.75\text{CSR}} \cdot \frac{\sqrt{0.75}}{\sqrt{\text{CSR}}} \cdot \text{erf}\left(\sqrt{\frac{\text{SNR}}{0.75\text{CSR}}}\right) &= \text{erf}(\sqrt{\text{CSR}}) \cdot \frac{1}{\sqrt{\pi}} \cdot e^{-\text{SNR}/\text{CSR}} \cdot \sqrt{\text{SNR}} \cdot \text{CSR}^{-3/2} \text{erf}\left(\sqrt{\frac{\text{SNR}}{0.75\text{CSR}}}\right) \cdot e^{-0.75\text{CSR}} \\ &\cdot \sqrt{0.75\text{CSR}} = \text{erf}(\sqrt{\text{CSR}}) \cdot e^{-\text{SNR}/\text{CSR}} \cdot \sqrt{\frac{\text{SNR}}{\text{CSR}}}. \end{aligned} \quad (32)$$

The equality can be established when $\text{CSR} = \sqrt{4/3\text{SNR}}$, which is the optimal CSR. We can obtain the lowest BER at the optimal CSR. The expression (29) can be expressed as Figure 6.

As can be seen from Figure 6, P_{ecov} minimizes at the optimal CSR. And the approximate solution we obtained is consistent with the theoretical P_{ecov} in Figure 6. With the approximate solution of optimal CSR, the process of searching for an actual optimal CSR can be accelerated when some optimization algorithms are adopted such as gradient descent and conjugate gradient.

As wireless communication is affected by channel fading, it is often necessary to set one or more relays to extend the communication distance. Therefore, the relay communication scenarios are described in details in the following subsections. \square

4.2. One-Relay Scenario. Covert information is always transmitted with low power; we can set relays to extend the transmission distance. Each relay employs the amplify-and-forward (AF) protocol and has two phases. Alice transmits

signal in one phase; the relay amplifies the signal and forwards to Bob in another phase. We can set the positions of Alice, Bob, Willie, and relay as illustrated in Figure 7(a). d_{tr} denotes the distance between Alice and Bob; d_{trr} denotes the distance between Alice and relay; and d_{rb} denotes the distance between relay and Bob. If we keep the total transmission energy constant, the transmission energy of Alice and relay are both $0.5P_b$, and we can obtain the output of the covert channel:

$$\begin{cases} \hat{y}_{\text{relay}}[i] = \sqrt{\frac{P_t}{2}} \cdot \frac{\sqrt{\lambda_0}}{d_{trr}} \cdot h_A \cdot S_{ct}[i] + n_A[i], \\ \hat{y}_{\text{Bob}}[i] = \sqrt{\frac{P_t}{2}} \cdot \frac{\sqrt{\lambda_0}}{d_{rb}} \cdot h_B \cdot \hat{y}_{\text{relay}}[i] + n_B[i]. \end{cases} \quad (33)$$

As is illustrated in Figure 7(a),

$$d_{rb} + d_{trr} = d_{tr}. \quad (34)$$

We can obtain the signal which is received by Bob:

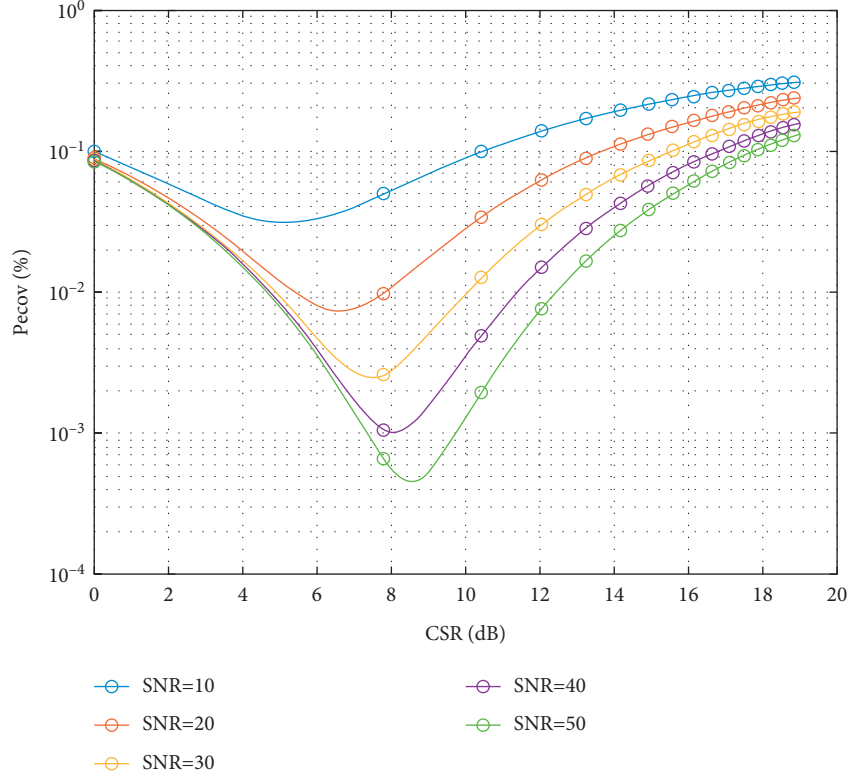


FIGURE 6: BER curve in AWGN channel.

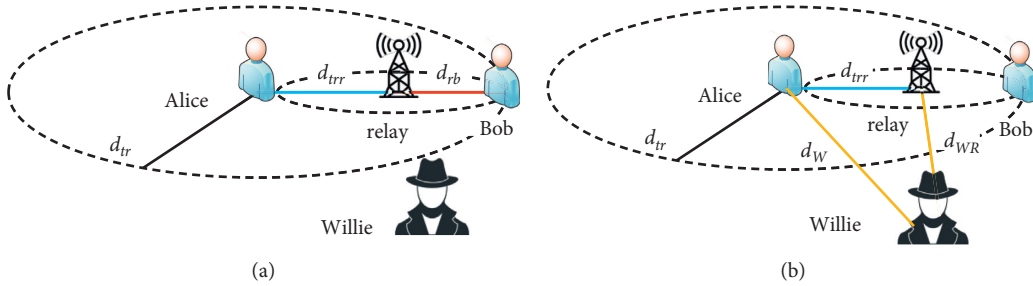


FIGURE 7: Diagram of relay location: (a) relay at random position and (b) relay in middle.

$$\begin{aligned} \hat{y}_{\text{Bob}}[i] &= \sqrt{\frac{P_t}{2}} \cdot \frac{\sqrt{\lambda_0}}{d_{rb}} \cdot h_B \cdot \left(\sqrt{\frac{P_t}{2}} \cdot \frac{\sqrt{\lambda_0}}{d_{trr}} \cdot h_A \cdot S_{ct}[i] + n_A[i] \right) + n_B[i], \\ &= \frac{P_t}{2} \cdot \frac{\lambda_0}{d_{rb} \cdot d_{trr}} \cdot h_B \cdot h_A \cdot S_{ct}[i] + n_C[i] \end{aligned} \quad (35)$$

In free space, the signal which is received by Bob is only about the energy and distance. So the optimization position of the relay is in the middle of Alice and Bob (i.e., $d_{rb} = d_{trr}$), as can be seen in Figure 7(b).

The distance between Willie, Alice, and relay are denoted by d_W and d_{WR} . The probability of undetectability P_{ud} can be expressed as

$$P_{ud} = \Pr \left\{ \frac{P_t}{2} \cdot \frac{\lambda_0 \cdot h_A^2}{d_W^2} < \text{SNR}_0 \right\} \cdot \Pr \left\{ \frac{P_t}{2} \cdot \frac{\lambda_0 \cdot h_B^2}{d_{WR}^2} < \text{SNR}_0 \right\}. \quad (36)$$

Construct a coordinate system with Alice as the origin of the coordinate axis. We can obtain the coordinates of Alice (0, 0), Bob ($2d_{trr}$, 0), relay (d_{trr} , 0), and Willie (X_{Willie} , Y_{Willie}).

In the AWGN channel, the power of signal received by Willie is just related to distance and transmit power. We can get “equipower lines” in the scenario of no relay and one relay.

Then, the power detected by Willie can be expressed as

$$2[(X_{\text{Willie}} - d_{\text{trr}})^2 + Y_{\text{Willie}}^2] = (X_{\text{Willie}})^2 + (Y_{\text{Willie}})^2. \quad (37)$$

Equation (37) can be expressed as

$$(X_{\text{Willie}} - 2d_{\text{trr}})^2 + Y_{\text{Willie}}^2 = (\sqrt{2}d_{\text{trr}})^2. \quad (38)$$

We denoted the circle expressed in equation (38) as the undetectability deterioration area (UDA). The undetectability deteriorates with setting relay when Willie is in the UDA. And the eavesdropper can detect in the UDA to improve detection efficiency.

It is illustrated in Figure 8, the green dotted line is UDA. If Willie is in the UDA, the P_{ud} will decrease in the scenario

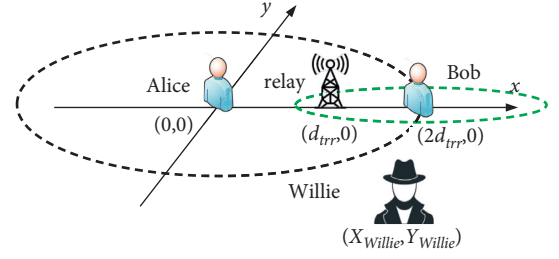


FIGURE 8: Power comparison of classic scenario and one relay scenario.

of one relay and the consequent deterioration of undetectability. Correspondingly, the undetectability will ameliorate if Willie is outside the green dotted circle. When William is on the green dotted line, the P_{ud} will be constant.

The BER of covert communication $P_{\text{ecov},r1}$ can be expressed as

$$P_{\text{ecov},r1} = 1 - [1 - P_{e,mc1}] \cdot [1 - P_{e,sc1}] [1 - P_{e,mc2}] \cdot [1 - P_{e,sc2}] + o(\Delta) \approx 1 - \left\{ \left[1 - \text{erfc} \left(\sqrt{\frac{2\text{SNR} \cdot \text{CSR}}{2\text{SNR} + \text{CSR} + 1}} \right) \right] \left[1 - \text{erfc} \left(\sqrt{\frac{2\text{SNR}}{\text{CSR} + 1}} \right) \right] \right\}^2. \quad (39)$$

Referring expression (32), we can obtain the local minimum of $P_{\text{ecov},r1}$ at $\text{CSR} = \sqrt{4/3\text{SNR}}$, which is the optimal CSR.

4.3. Two-Relay Scenario. Based on the above, we discuss the two-relay scenario. If we keep the total transmission energy constant, the transmission energy of Alice, relay1, and relay2 are all $P_t/3$. We can obtain the output of the covert signal:

$$\begin{cases} \hat{y}_{\text{relay1}}[i] = \sqrt{\frac{P_t}{3}} \cdot \frac{\sqrt{\lambda_0}}{d_{\text{trr}}} \cdot h_A \cdot S_{ct}[i] + n_A[i], \\ \hat{y}_{\text{relay2}}[i] = \sqrt{\frac{P_t}{3}} \cdot \frac{\sqrt{\lambda_0}}{d_{\text{trr}}} \cdot h_B \cdot \hat{y}_{\text{relay1}}[i] + n_B[i], \\ \hat{y}_{\text{Bob}}[i] = \sqrt{\frac{P_t}{3}} \cdot \frac{\sqrt{\lambda_0}}{d_{\text{trr}}} \cdot h_C \cdot \hat{y}_{\text{relay2}}[i] + n_C[i]. \end{cases} \quad (40)$$

As can be seen in Figure 9, the distances between Willie and Alice, relay1, and relay2 are denoted by d_{WR} , d_{WR1} , and d_{WR2} , respectively.

The probability of undetected P_{ud} can be expressed as

$$P_{ud} = \Pr \left\{ \frac{P_t}{3} \cdot \frac{\lambda_0 \cdot h_A^2}{d_W^2} < \text{SNR}_0 \right\} \cdot \Pr \left\{ \frac{P_t}{3} \cdot \frac{\lambda_0 \cdot h_B^2}{d_{WR1}^2} < \text{SNR}_0 \right\} \cdot \Pr \left\{ \frac{P_t}{3} \cdot \frac{\lambda_0 \cdot h_C^2}{d_{WR2}^2} < \text{SNR}_0 \right\}. \quad (41)$$

Further extension, the probability of undetectability for n hops can be expressed as

$$P_{ud} = \prod_n \Pr \left\{ \frac{P_t}{n} \cdot \frac{\lambda_0 \cdot h_n^2}{d_{Wn}^2} < \text{SNR}_0 \right\}. \quad (42)$$

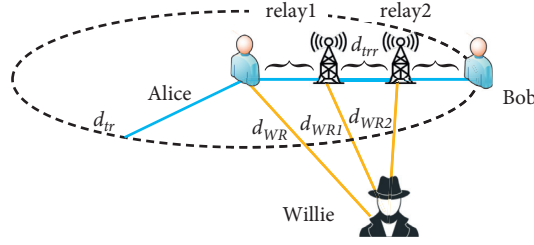


FIGURE 9: Diagram of two relays location.

The UDA of relay1 can be expressed as

$$3[(X_{\text{Willie}} - d_{\text{trr}})^2 + Y_{\text{Willie}}^2] = X_{\text{Willie}}^2 + Y_{\text{Willie}}^2. \quad (43)$$

Equation (43) can be converted to

$$\left(X_{\text{Willie}} - \frac{3}{2}d_{\text{trr}}\right)^2 + Y^2 = \left(\frac{\sqrt{3}}{2}d_{\text{trr}}\right)^2. \quad (44)$$

The UDA of relay2 can be expressed as

$$3[(X_{\text{Willie}} - 2d_{\text{trr}})^2 + Y_{\text{Willie}}^2] = X_{\text{Willie}}^2 + Y_{\text{Willie}}^2. \quad (45)$$

Equation (45) can be converted to

$$(X_{\text{Willie}} - 3d_{\text{trr}})^2 + Y^2 = (\sqrt{3}d_{\text{trr}})^2. \quad (46)$$

As can be seen in Figure 10, the circles expressed in equations (44) and (46) are UDAs. If Willie is in the blue or green dotted circle, the undetectability will deteriorate.

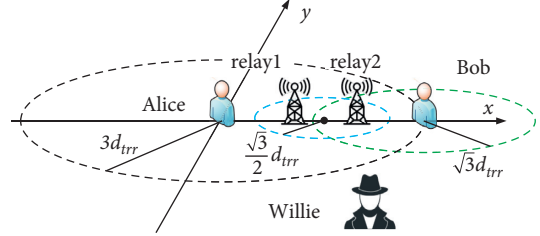


FIGURE 10: Power comparison of the classic scenario and two-relay scenario.

Correspondingly, the undetectability will ameliorate if Willie is outside the blue and green dotted circles.

The BER of covert communication $P_{\text{ecov},r2}$ can be expressed as

$$P_{\text{ecov},r2} = 1 - [1 - P_{e,mc1}] \cdot [1 - P_{e,sc1}] [1 - P_{e,mc2}] \cdot [1 - P_{e,sc2}] [1 - P_{e,mc3}] \cdot [1 - P_{e,sc3}] + o(\Delta) \approx 1 - \left\{ \left[1 - \text{erfc} \left(\sqrt{\frac{3\text{SNR} \cdot \text{CSR}}{3\text{SNR} + \text{CSR} + 1}} \right) \right] \left[1 - \text{erfc} \left(\sqrt{\frac{3\text{SNR}}{\text{CSR} + 1}} \right) \right] \right\}^3. \quad (47)$$

It has been proved that there exists an optimal CSR, and we can obtain the minimum of $P_{\text{ecov},r2}$ at $\text{CSR} = \sqrt{4/3\text{SNR}}$.

5. Experimental Result

5.1. Experimental Setup. In this section, we inspect the undetectability and reliability to benchmark the proposed scheme. We set the wireless communication on an 802.11a/g PHY layer. The wireless covert channel is performed on all 100000 symbols. In transmissions, there are 48 subcarriers in a symbol. Simulation experiments are carried out in wireless channel models of AWGN channel models [39]. In some simulations, the wireless covert channel with dirty constellation (WCC-DC) is chosen for comparison. The undetectability is inspected by “KL divergence” and “KS distance”. The undetectability measures of I vectors, Q vectors, magnitudes, and phases of constellation errors are presented in the range of transmission power $\text{SNR} = 10, \dots, 40$ dB. The reliability is measured by BERs.

5.2. Undetectability. Willie (i.e., detector) observes the channel to judge whether Alice (i.e., transmitter) is transmitting in the covert channel or not. There must be a threshold Γ to compare the value of “KL divergence” and “KS distance”. If the Γ is great, the P_F will be too great. If the Γ is small, the P_M will be too great. In this paper, we set the Γ of four measures of “KL divergence” as [0.04, 0.04, 0.055, 0.055], and the Γ of “KS distance” as [0.025, 0.025, 0.025, 0.025].

In this section, we set the number of bins $K = 100$. Four samples were chosen for comparison, and the samples are as follows: WCC-DC with $\text{CSR} = 5$ dB and $\text{CSR} = 10$ dB, WCC-CSM with $\text{CSR} = 5$ dB, and $\text{CSR} = 10$ dB.

As can be seen in Figure 11, the “KL divergence” of WCC-CSM with $\text{CSR} = 5$ dB and $\text{CSR} = 10$ dB meet the threshold Γ [0.04, 0.04, 0.055, 0.055] in the range of $\text{SNR} = 10, \dots, 40$ dB. The resulting KL divergence is lower than the KL divergence achieved with WCC-DC.

In Figure 11(a), WCC-DC with $\text{CSR} = 10$ dB meets the threshold Γ of I vectors in the range of $\text{SNR} = 10, \dots, 30$ dB. The “KL divergence” of WCC-DC with $\text{CSR} = 10$ dB exceeds the

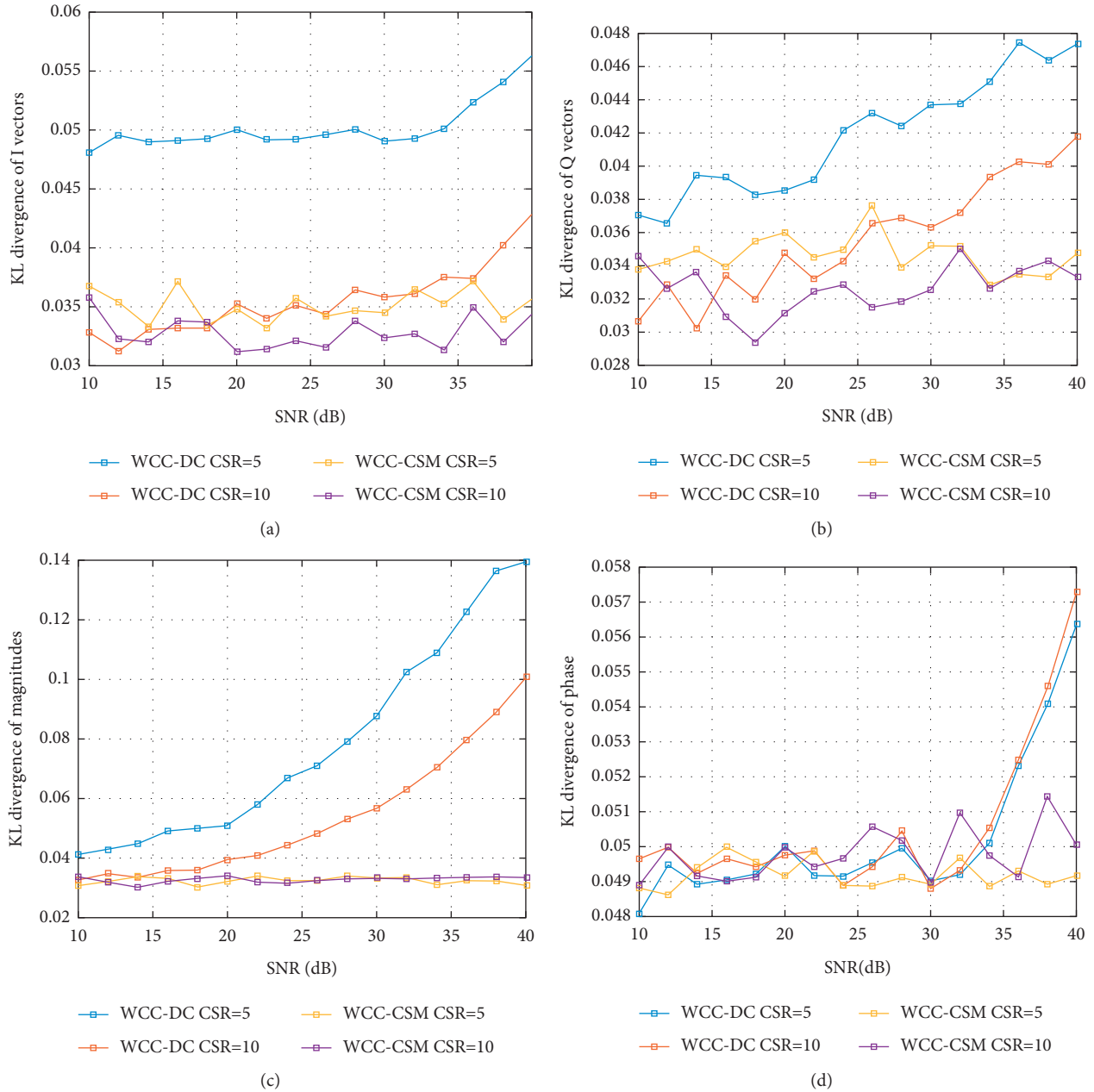


FIGURE 11: KL divergence of constellation errors with different CSRs: (a) KL divergence of I vectors, (b) KL divergence of Q vectors, (c) KL divergence of magnitudes, and (d) KL divergence of phase.

threshold Γ when SNR = 40 dB, and WCC-DC with CSR = 5 dB exceeds the threshold in the range of SNR = 10, ... 40 dB. In Figure 11(b), the “KL divergence” of WCC-DC with CSR = 5 dB exceeds the threshold Γ at SNR = 23 dB. The “KL divergence” of WCC-DC with CSR = 10 dB exceeds the threshold Γ at SNR = 33 dB. In Figures 11(c) and 11(d), WCC-DC with CSR = 5 dB exceeds the threshold Γ at SNR = 23 dB and SNR = 38 dB, and WCC-DC with CSR = 10 dB exceeds the threshold Γ at SNR = 28 dB and SNR = 37 dB.

As can be seen in Figure 12, WCC-CSM meets the threshold Γ and WCC-CSM has a smaller “KS distance” than WCC-DC in the range of SNR = 10, ..., 40 dB. The “KS

distance” of WCC-CSM changes little with different CSRs. And we can regulate the CSR without exceeding the threshold Γ of “KS distance.” In Figures 12(a)–12(c), the “KS distance” of WCC-DC with CSR = 5 dB exceeds the threshold at SNR = 10 dB. In Figure 12(c), the “KS distance” of WCC-DC with CSR = 10 dB exceeds the threshold at SNR = 13 dB.

We can come to a stage conclusion, WCC-CSM meets the threshold of “KL divergence” and “KS distance” in the range of SNR = 10, ..., 40 dB. Hence, we can regulate the CSR to reduce the BER without exceeding the threshold of “KL divergence” or “KS distance.”

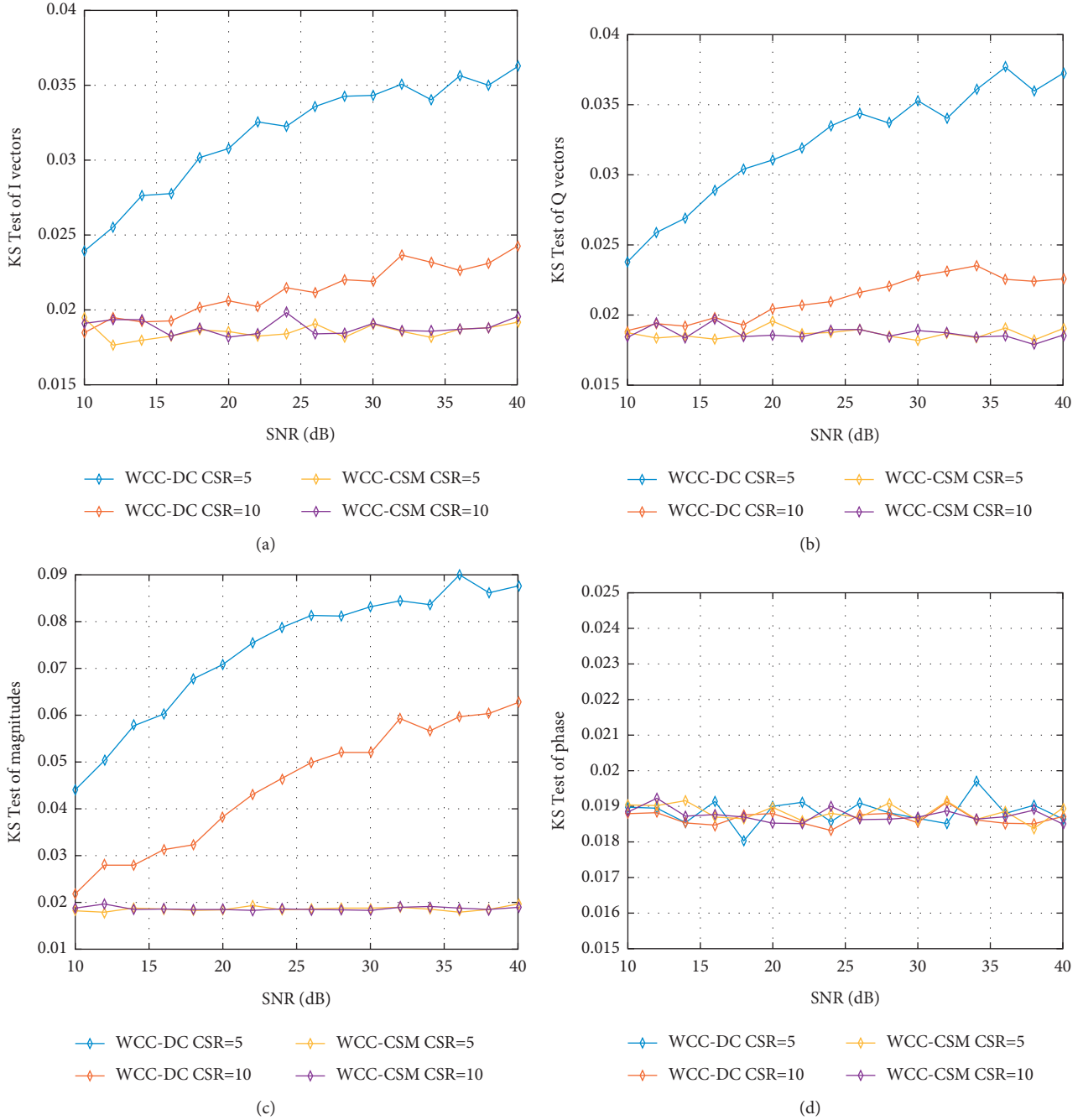


FIGURE 12: KS distances of constellation errors with different CSRs: (a) KS distances of I vectors, (b) KS distances of Q vectors, (c) KS distances of magnitudes, and (d) KS distances of phase.

5.3. Reliability. The reliability of the system is measured by the BER. As is illustrated in expression (28), the BER decreases with the increase of SNR. In Section 4.1, the optimal CSR was proposed. And the approximate solution of optimal CSR is obtained and extended to several scenarios (one relay, two relays). The BER curves of wireless covert channels in several scenarios are shown in Figure 13. The position of Alice, Bob, relay1, and relay 2 was shown in Section 4.

In Figure 13(a), the BER of the classic scenario with different SNRs was presented in the range of CSR = 1, 2, ..., 20 dB. The BER of WCC-CSM with SNR = 15 dB minimizes

at CSR = 7 dB, which is 10% lower than the BER at CSR = 15 dB. The approximate solution of optimal CSR is 5 dB. The minimizing BER of WCC-CSM with SNR = 30 dB is 4% at CSR = 8 dB, and the approximate solution of optimal CSR is 7 dB. The minimizing BER of WCC-CSM with SNR = 40 dB is 0.05% at CSR = 9 dB. The theoretical approximation of the optimal CSR-with SNR = 40 dB is 7.5 dB.

It is proved that an optimal CSR exists and the BER minimizes at the optimal CSR. With the increase of SNR, the optimal CSR gradually increases. But the theoretical approximate value of optimal CSR is slightly lower than the

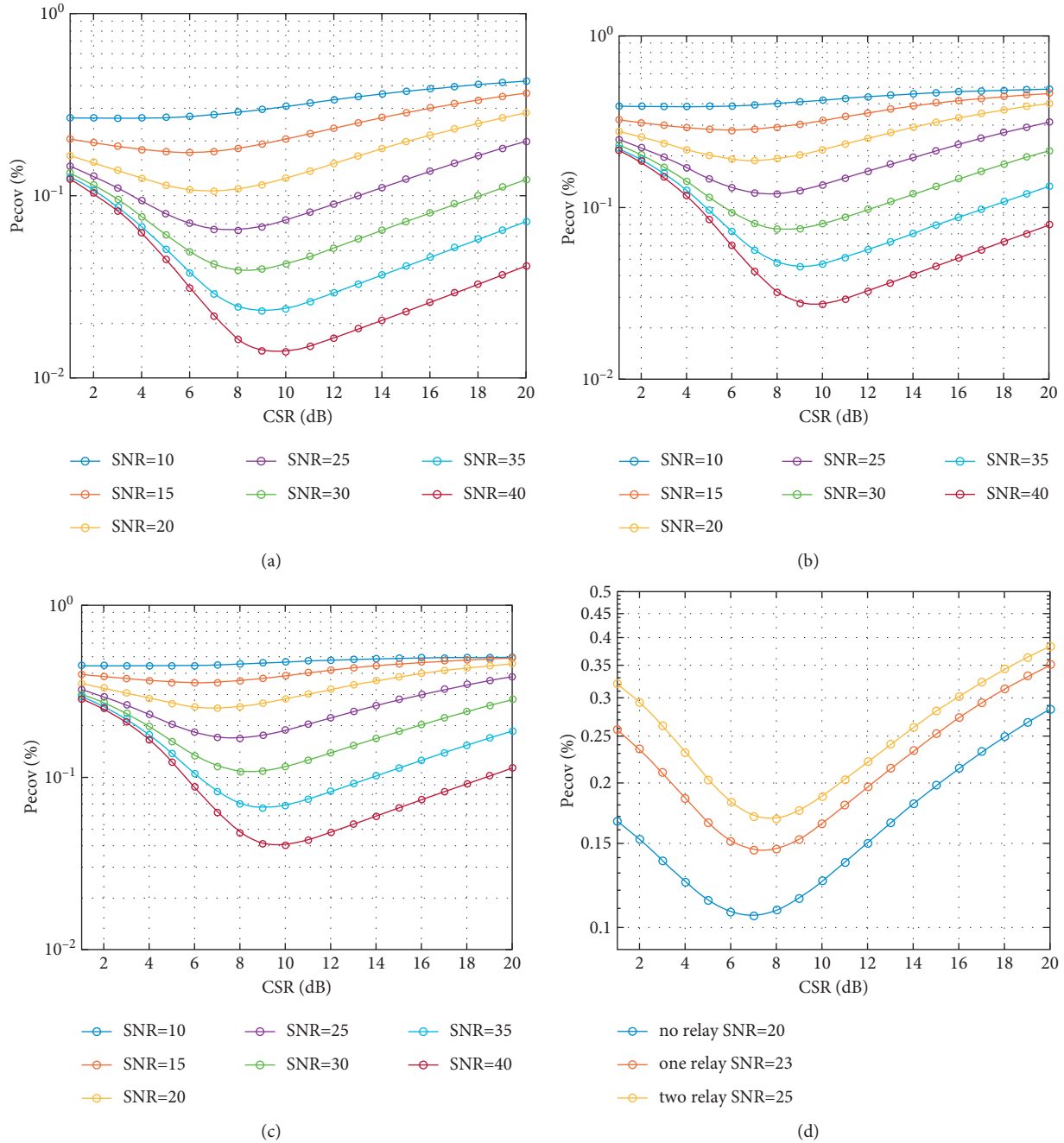


FIGURE 13: BER of wireless covert channel in typical scenarios. (a) BER of classic scenario with different SNR. (b) BER of one relay with different SNR. (c) BER of two relays with different SNR. (d) Comparison of BER of several scenarios.

simulation results. As can be seen in Figures 13(b) and 13(c), the optimal CSR exists and is slightly higher than the theoretical approximation in scenarios of relays. The BER minimizes at the optimal CSR.

Comparing the BER under the constraints of constant total power, suppose that the SNR in the classic scenario is 20 dB. The SNR in the scenario of one relay is 23 dB, and the SNR in the scenario of two relays is 25 dB. As can be seen in Figure 13(d), the reliability of classic scenario is optimal and the BER minimizes at the optimal CSR. No error correction

coding is used in the paper, and the reliability deteriorates in the scenarios of relays. It is proved that simply setting relays without increasing the total power, the BER of covert communication will deteriorate.

Simulation experiments are carried out in wireless channel models of AWGN channel. As can be seen in Figure 13, we can obtain minimum P_{ecov} , $P_{ecov,r1}$, and $P_{ecov,r2}$ at the optimal CSR in the AWGN channel. And the optimal CSR achieved in simulation is slightly higher than the theoretical approximation.

6. Conclusions

Reliability and undetectability are the main aspects of wireless covert communication. We considered the BER problem of covert communication based on WCC-CSM. We studied the impact of carrier-secret ratio (CSR) on the BER and investigated the relationship between SNR, CSR, and BER. We obtained the approximate solution of optimal CSR and extended it to the scenario of relays. With the approximate solution of optimal CSR, the process of searching for an actual optimal CSR can be accelerated. Furthermore, we found that the undetectability under the constraints of constant total power depends on the eavesdropper's position. And we found an undetectability deterioration area (UDA) in the scenario of relays, and undetectability deteriorates with setting relays when an eavesdropper is in the UDA.

The simulation proved that there exists an optimal CSR in the AWGN channel. The transmitter can obtain greater reliability with great undetectability at the optimal CSR. Additionally, the reliability deteriorates with setting relays under the constraints of constant total power. Some error correction coding or other methods must be adopted, avoiding the deterioration of BER.

To improve the detection capability of Willie, it is necessary to find a better way to detect the covert communication except "KL divergence" or "KS distance" in our future work.

Data Availability

The data used to support the findings of this study are included within the supplementary information files.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. U1836104, 61772281, 61702235, 61801073, 61931004, and 62072250).

Supplementary Materials

These files contain the KL divergence of four vectors, KS distance of four vectors, and BERs of wireless covert communication in typical scenarios. (i) KL divergence of constellation errors with different CSR.docx: (a) KL divergence of I vectors, (b) KL divergence of Q vectors, (c) KL divergence of magnitudes, and (d) KL divergence of phase. (ii) KS distances of constellation errors with different CSR.docx: (a) KS distances of I vectors, (b) KS distances of Q vectors, (c) KS distances of magnitudes, and (d) KS distances of phase. (iii) BERs of wireless covert communication in typical scenarios.docx: (a) BERs of classic scenario with different SNR, (b) BERs of one relay with different SNR, (c) BERs of two relays with different SNR, and (d) comparison of BERs of several scenarios. (*Supplementary Materials*)

References

- [1] P. Vijayakumar, M. S. Obaidat, M. Azees, S. H. Islam, and N. Kumar, "Efficient and secure anonymous authentication with location privacy for IoT-based WBANs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2603–2611, 2019.
- [2] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2013.
- [3] M. Xu, D. Wang, Q. Wang, and Q. Jia, "Understanding security failures of anonymous authentication schemes for cloud environments," *Journal of Systems Architecture*, vol. 118, Article ID 102206, 2021.
- [4] B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, p. e4946, 2020.
- [5] A. Al-Qerem, M. Alauthman, A. Almomani, and B. B. Gupta, "IoT transaction processing through cooperative concurrency control on fog-cloud computing environment," *Soft Computing*, vol. 24, no. 8, pp. 5695–5711, 2020.
- [6] M. Azees and P. Vijayakumar, "CEKD: computationally efficient key distribution scheme for vehicular ad-hoc networks," *Australian Journal of Basic and Applied Sciences*, vol. 10, no. 2, pp. 171–175, 2016.
- [7] H. Kim, E. Kang, D. Broman, and E. A. Lee, "Resilient authentication and authorization for the Internet of Things (IoT) using edge computing," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 1–27, 2020.
- [8] M. A. Christie, A. Bhandar, S. Nakandala et al., "Managing authentication and authorization in distributed science gateway middleware," *Future Generation Computer Systems*, vol. 111, pp. 780–785, 2020.
- [9] M. Yamni, A. Daoui, O. El ogri et al., "Fractional Charlier moments for image reconstruction and image watermarking," *Signal Processing*, vol. 171, Article ID 107509, 2020.
- [10] P. Sun, "Security and privacy protection in cloud computing: discussions and challenges," *Journal of Network and Computer Applications*, vol. 160, Article ID 102642, 2020.
- [11] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, "IoT-based big data secure management in the fog over a 6G wireless network," *IEEE Internet of Things Journal*, vol. 8, 2020.
- [12] B. A. Bash, D. Goeckel, and D. Towsley, "Limits of reliable communication with low probability of detection on AWGN channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1921–1930, 2013.
- [13] S. Yan, Y. Cong, S. V. Hanly, and X. Zhou, "Gaussian signalling for covert communications," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3542–3553, 2019.
- [14] J. Hu, S. Yan, F. Shu, and J. Wang, "Covert transmission with a self-sustained relay," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4089–4102, 2019.
- [15] S. Yan, X. Zhou, N. Yang, B. He, and T. D. Abhayapala, "Artificial-noise-aided secure transmission in wiretap channels with transmitter-side correlation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8286–8297, 2016.
- [16] A. Sheikholeslami, M. Ghaderi, D. Towsley, B. A. Bash, S. Guha, and D. Goeckel, "Multi-hop routing in covert wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3656–3669, 2018.

- [17] S. Lee, R. J. Baxley, J. B. McMahon, and R. S. Frazier, "Achieving positive rate with undetectable communication over MIMO Rayleigh channels," in *Proceedings of the 2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, IEEE, A Coruna, Spain, June 2014.
- [18] J. Yao, X. Zhou, Y. Liu, and S. Feng, "Secure transmission in linear multihop relaying networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 822–834, 2017.
- [19] Z. Liu, J. Liu, Y. Zeng, J. Ma, and Q. Huang, "On covert communication with interference uncertainty," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, IEEE, Kansas City, MO, USA, May 2018.
- [20] J. Hu, K. Shahzad, S. Yan, X. Zhou, F. Shu, and J. Li, "Covert communications with a full-duplex receiver over wireless fading channels," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, IEEE, Kansas City, MO, USA, May 2018.
- [21] H.-M. Wang, Y. Zhang, X. Zhang, and Z. Li, "Secrecy and covert communications against UAV surveillance via multihop networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 389–401, 2019.
- [22] K. Shahzad, X. Zhou, S. Yan, J. Hu, F. Shu, and J. Li, "Achieving covert wireless communications using a full-duplex receiver," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8517–8530, 2018.
- [23] F. Shu, T. Xu, J. Hu, and S. Yan, "Delay-constrained covert communications with a full-duplex receiver," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 813–816, 2019.
- [24] T. V. Sobers, B. A. Bash, D. Goeckel, S. Guha, and D. Towsley, "Covert communication with the help of an uninformed jammer achieves positive rate," in *Proceedings of the 2015 49th Asilomar Conference on Signals, Systems and Computers*, IEEE, Pacific Grove, CA, USA, November 2015.
- [25] R. Soltani, B. Bash, D. Goeckel, S. Guha, and D. Towsley, "Covert single-hop communication in a wireless network with distributed artificial noise generation," in *Proceedings of the 2014 52nd Annual Allerton Conference on communication, control, and computing (Allerton)*, IEEE, Monticello, IL, USA, October 2014.
- [26] K. Li, P. A. Kelly, and D. Goeckel, "Optimal power adaptation in covert communication with an uninformed jammer," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3463–3473, 2020.
- [27] R. Soltani, D. Goeckel, D. Towsley, B. A. Bash, and S. Guha, "Covert wireless communication with artificial noise generation," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7252–7267, 2018.
- [28] M. Forouzes, P. Azmi, A. Kuhestani, and P. L. Yeoh, "Covert communication and secure transmission over untrusted relaying networks in the presence of multiple wardens," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3737–3749, 2020.
- [29] K. Shahzad, "Relaying via cooperative jamming in covert wireless communications," in *Proceedings of the 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, IEEE, Cairns, Australia, December 2018.
- [30] K. S. K. Arumugam, M. R. Bloch, and L. Wang, "Covert communication over a physically degraded relay channel with non-colluding wardens," in *Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT)*, IEEE, Vail, CO, USA, June 2018.
- [31] A. Abdelaziz and C. E. Koks, "Fundamental limits of covert communication over MIMO AWGN channel," in *Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS)*, IEEE, Las Vegas, NV, USA, October 2017.
- [32] K. S. K. Arumugam and M. R. Bloch, "Covert communication over broadcast channels," in *Proceedings of the 2017 IEEE Information Theory Workshop (ITW)*, IEEE, Kaohsiung, Taiwan, November 2017.
- [33] K. S. Kumar Arumugam and M. R. Bloch, "Embedding covert information in broadcast communications," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2787–2801, 2019.
- [34] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, *Secret Agent Radio: Covert Communication through Dirty Constellations*, Springer Berlin Heidelberg, Berlin, Germany, 2013.
- [35] P. Cao, W. Liu, G. Liu, X. Ji, J. Zhai, and Y. Dai, "A wireless covert channel based on Constellation shaping modulation," *Security and Communication Networks*, vol. 2018, Article ID 1214681, 15 pages, 2018.
- [36] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology*/Springer, New York, NY, USA, 1984.
- [37] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, Springer Science & Business Media, New York, NY, USA, 2006.
- [38] T. M. Cover, *Elements of Information Theory*, John Wiley & Sons, Hoboken, NJ, USA, 1999.
- [39] I. L. M. S. Committee, *Wireless LAN media Access Control (MAC) and Physical Layer (PHY) Specifications*, Standards, London, UK, 2009.

Research Article

FNet: A Two-Stream Model for Detecting Adversarial Attacks against 5G-Based Deep Learning Services

Guangquan Xu,^{1,2} Guofeng Feng ,¹ Litao Jiao,² Meiqi Feng,¹ Xi Zheng,³ and Jian Liu ¹

¹College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

²School of Big Data, Qingdao Huanghai University, Qingdao 266555, China

³Department of Computing, Macquarie University, North Ryde 2113, Australia

Correspondence should be addressed to Jian Liu; jianliu@tju.edu.cn

Received 26 June 2021; Accepted 17 August 2021; Published 7 September 2021

Academic Editor: Benjamin Aziz

Copyright © 2021 Guangquan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the extensive application of artificial intelligence technology in 5G and Beyond Fifth Generation (B5G) networks, it has become a common trend for artificial intelligence to integrate into modern communication networks. Deep learning is a subset of machine learning and has recently led to significant improvements in many fields. In particular, many 5G-based services use deep learning technology to provide better services. Although deep learning is powerful, it is still vulnerable when faced with 5G-based deep learning services. Because of the nonlinearity of deep learning algorithms, slight perturbation input by the attacker will result in big changes in the output. Although many researchers have proposed methods against adversarial attacks, these methods are not always effective against powerful attacks such as CW. In this paper, we propose a new two-stream network which includes RGB stream and spatial rich model (SRM) noise stream to discover the difference between adversarial examples and clean examples. The RGB stream uses raw data to capture subtle differences in adversarial samples. The SRM noise stream uses the SRM filters to get noise features. We regard the noise features as additional evidence for adversarial detection. Then, we adopt bilinear pooling to fuse the RGB features and the SRM features. Finally, the final features are input into the decision network to decide whether the image is adversarial or not. Experimental results show that our proposed method can accurately detect adversarial examples. Even with powerful attacks, we can still achieve a detection rate of 91.3%. Moreover, our method has good transferability to generalize to other adversaries.

1. Introduction

Deep learning has recently led to significant improvements in many fields, such as computer vision [1–3], speech recognition [4, 5], and natural language processing [6, 7]. With the continuous development of 5G communication and artificial intelligence technology, the two have developed from mutual independence to deep integration. The artificial intelligence promotes the intelligent development of the communication network itself, and the industry widely believes that 5G and artificial intelligence are general-purpose technologies (GPTs) [8]. Many have explored the application of artificial intelligence in 5G communication in the form of investigation or empirical research [9, 10]. Under this trend, communication artificial intelligence has

developed rapidly and vigorously. For example, Chinese operators provide services to users and partners by AI center [11]. The AI center is based on artificial intelligence algorithms, encapsulates scene-oriented services, and provides applications and services in customer sales and customer service.

Although deep learning is powerful, there are security risks in deep learning services provided by neural network models. For example, in customer sales services deployed on 5G platforms, the main application of artificial intelligence technology is image classification. This service effectively recognizes the image data through the image classification model. Although the image classification model provides great convenience for users, the inherent fragile of deep learning is a weakness of the service. Recent studies [12, 13]

have shown that an attacker can create adversarial samples by adding small disturbances to the original data. The disturbances are very small, and they are almost invisible to the human eye. If the attacker inputs adversarial samples into the recognition model, the model will not correctly recognize the samples.

The deep learning model deployed on the 5G platform provides intelligent image recognition services. Hackers attack the deep learning model, causing errors in the deployed image recognition service. Specifically, the attacker adds a small disturbance to the original input to cause a huge change in the output of model. Most adversarial attacks currently existing are aimed at image classification. To ensure the security of 5G-based deep learning services, we mainly research the detection method of adversarial samples to protect against image classification models. Although methods [14–17] are proposed to detect adversarial examples, these methods always fail when faced with a powerful attack like the CW. To solve the adversarial attack on the image classification service based on the 5G platform, we mainly research the detection method of adversarial samples. In our method, we regard adversarial disturbances as special noise features that could provide additional evidence for adversarial sample detection. Since adversarial disturbance and image steganography both modify the picture directly, destroying the correlation between the original image pixels, we can apply the method of steganalysis to the field of adversarial sample detection. In fact, Goodfellow et al. also proposed that adversarial attacks can be regarded as accidental steganography [18].

In this paper, we use rich features extracted from the spatial rich model (SRM) [19] to help the deep learning model for detecting adversarial examples. The SRM is a traditional approach to extract noise features in steganalysis [20]. The emergence of steganography promoted the development of steganalysis. Steganography is to add secret information to the original carrier, making the information hidden in the carrier difficult to detect [21]. The steganography of the picture changes the pixel value of the picture, which will destroy the correlation between adjacent pixels of the original image. Therefore, steganalysis can be performed according to this. Steganalysis determines whether the image has steganalysis by modeling the correlation between adjacent pixels of the image. Traditional steganalysis algorithms are based on manual feature extraction. After continuous research by many scholars, SRM has been able to extract 30,000 multidimensional features from the data through improved high-pass filters (HPFs) [22]. The design of these high-pass filters is mostly based on experience. SRM uses 30 different pixel predictors. The pixel predictor is linear or nonlinear. Each linear predictor is a shift-invariant finite-impulse response filter which is described by a kernel matrix K^{pred} . The residual R is a matrix which has the same dimension as Y :

$$R = K^{\text{pred}} * Y - Y \triangleq K * Y, \quad (1)$$

where the symbol $*$ denotes the convolution with Y mirrored. Thus, R has the same dimension as Y .

There are six types of residuals: first-order, second-order, third-order, SQUARE, EDGE 3×3 , and EDGE 5×5 [19]. Table 1 shows the calculation methods of first-order, second-order, and third-order linear residuals. For example, one simple linear residual is $R_{ij}^h = y_{ij+1} - y_{ij}$, which is the difference between a pair of horizontally adjacent pixels. In this case, the residual kernel is $K = (-1, 1)$, which means that the pixel value is predicted as its horizontally neighboring pixel. We can use this method to extract the noise features of other directions.

SQUARE, EDGE 3×3 , and EDGE 5×5 linear residuals use more directional neighborhood pixels in their calculations. Tables 2 and 3 show SQUARE, EDGE 3×3 , and EDGE 5×5 SRM filter kernels.

Our model consists of a two-stream network and a decision network. The RGB stream is used to capture subtle differences, such as contrast differences of a RGB image. The SRM noise stream is used to capture the noise inconsistency between clean samples and adversarial samples. We use 30 typical SRM filters to extract noise features from adjacent pixels. The noise features are used as the SRM noise input of the two-stream model. Then, we use bilinear pooling [23] to fuse the features extracted from the two streams. Bilinear pooling used for it can fuse the features of the two streams while preserving spatial information. Finally, we use the fully connected layer as a decision network to detect adversarial samples.

Our contributions are summarized as follows:

- (1) To improve the security of 5G-based deep learning services, we propose a new two-stream adversarial example detection model and perform end-to-end training. This method can obtain rich feature information from noise features and provide additional evidence for adversarial example detection. Even with a powerful CW attack, we can still achieve a detection rate of 91.3%.
- (2) We choose the spatial rich model (SRM) to generate linear and nonlinear noise features. The 30 SRM filters could amplify the difference in the noise domain and get additional rich information to help detect adversarial samples.

The rest of this paper is organized as follows. Section 2 describes the related work about adversarial sample attacks and against deep learning models. Section 3 introduces the proposed two-stream network. Section 4 analyzes the experimental results. Finally, Section 5 concludes the paper.

2. Related Work

We briefly review the related work in adversarial attack and adversarial defense in this section.

2.1. Security Issues of 5G-Based Deep Learning Services. Due to the inherent vulnerability of neural networks, 5G-based deep learning services face the same threat of adversarial attacks. Deep learning (including deep reinforcement learning) is vulnerable to adversarial examples

TABLE 1: The SRM linear residual filters, where R_{ij}^h denotes the linear residual of pixel at the position (i, j) in horizontal direction and y_{ij} denotes the pixel at the position (i, j) .

Residual type	HPF	Linear residual
First-order	(1, -1)	$R_{ij}^h = y_{ij+1} - y_{ij}$
Second-order	(1, -2, 1)	$R_{ij}^h = y_{ij-1} - 2y_{ij} + y_{ij+1}$
Third-order	(1, -3, 3, -1)	$R_{ij}^h = y_{ij-1} - 3y_{ij} + 3y_{ij+1} - y_{ij+2}$

TABLE 2: The SQUARE SRM filter kernels.

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$$

TABLE 3: The EDGE SRM filter kernels.

$$\begin{bmatrix} 2 & -1 \\ 4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} -2 & 2 & -1 \\ 8 & -6 & 2 \\ -12 & 8 & -2 \\ 8 & -6 & 2 \\ -2 & 2 & -1 \end{bmatrix}$$

[24]. These modified inputs can trick the model into making the wrong decision. Hackers may use this vulnerability to attack services based on deep learning [25]. Adversarial attacks can be divided into black-box attacks and white-box attacks. The black-box attack can only obtain the identification result of the inputs through the API interface of the platform but cannot get the internal parameters of the model. The white-box attack can get all the parameters of the neural network model deployed on the platform. Therefore, white-box attacks also represent the highest level of attack.

2.2. Deep Learning Attack. The deep learning algorithm is a weakness of deep learning services based on 5G. Because of the inherent vulnerability of neural networks [26], a small adversarial disturbance can cause a huge change in the output of the model. In addition, scholars have proposed attacks on the model optimization process and regularization process [27, 28].

Goodfellow et al. [18] proposed fast gradient notation (FGSM) to generate adversarial examples. The main idea of FGSM is to make the disturbance direction consistent with the gradient direction to maximize the change in the loss function value and then maximize the change in the classification result of the classifier.

Moosavi-Dezfooli et al. [28] proposed DeepFool to find the shortest distance from the clean images to the decision boundary of the adversarial images. DeepFool is a non-targeted attack method to generate an adversarial example by iteratively perturbing an image. Experiments show that the DeepFool method produces less interference than FGSM and has similar fool rates.

Carlini and Wagner [27] proposed the CW method to generate adversarial examples. CW is an adversarial example attack algorithm based on objective function optimization. It restricts the L_{∞} , L_2 , and L_0 norms to make the disturbance undetectable. Experiments show that defensive distillation is completely unable to defend against these three kinds of attacks. The CW method is a strong attack which is difficult to defend.

2.3. Deep Learning Defense. Recent research [29] has shown that adversarial examples not only mislead classifiers in electronic data but also have the same effect in the physical world. In view of the great harm of adversarial examples, many scholars have studied the defense of adversarial examples.

The adversarial training proposed by GoodFellow et al. [18] is an earlier measure to defend against samples. The main method is to train the adversarial samples and the normal samples simultaneously in the training stage to enhance the robustness of the model. However, confrontational training requires a large number of adversarial examples to ensure a high detection rate, which will lead to huge training costs.

Hinton et al. [15] first introduced the distillation defense method for small models to imitate large models. Later, Papernot et al. [30] adopted the distillation method to defend adversarial samples. This method makes the decision boundary of the model smoother and can effectively defend against the adversarial samples generated by FGSM, BIM, and other algorithms. Compared with adversarial training, its training cost is lower. However, this method is not effective against CW attacks.

Dziugaite et al. [16] studied the defensive effect of JPG feature compression against FGSM attacks. The limitation of this defense method is that large-scale compression which will lead to the decline of the accuracy of legal sample classification. For the sample with large interference, it is not enough to eliminate interference.

Due to the difficulty of reclassification of antagonistic samples, many researchers turn to the detection of adversarial samples.

Liang et al. [17] proposed an adversarial sample detection method based on adaptive denoising. In this method, antagonistic interference is regarded as artificial noise, and noise reduction technology is used to reduce its adversarial effect. Liu et al. [31] proposed to apply steganalysis to detect adversarial examples. This work found a relevant connection between computer vision and adversarial examples of steganalysis. Feinman et al. [32] proposed a method to detect adversarial examples using the Bayesian uncertainty of the network and the method of kernel density estimation.

2.4. Proposed Method. We designed a two-stream network to detect adversarial examples. As shown in Figure 1, the RGB stream directly uses the RGB image for input and the SRM stream uses the noise features extracted by the SRM filter as the input. Then, we use bilinear pooling to fuse

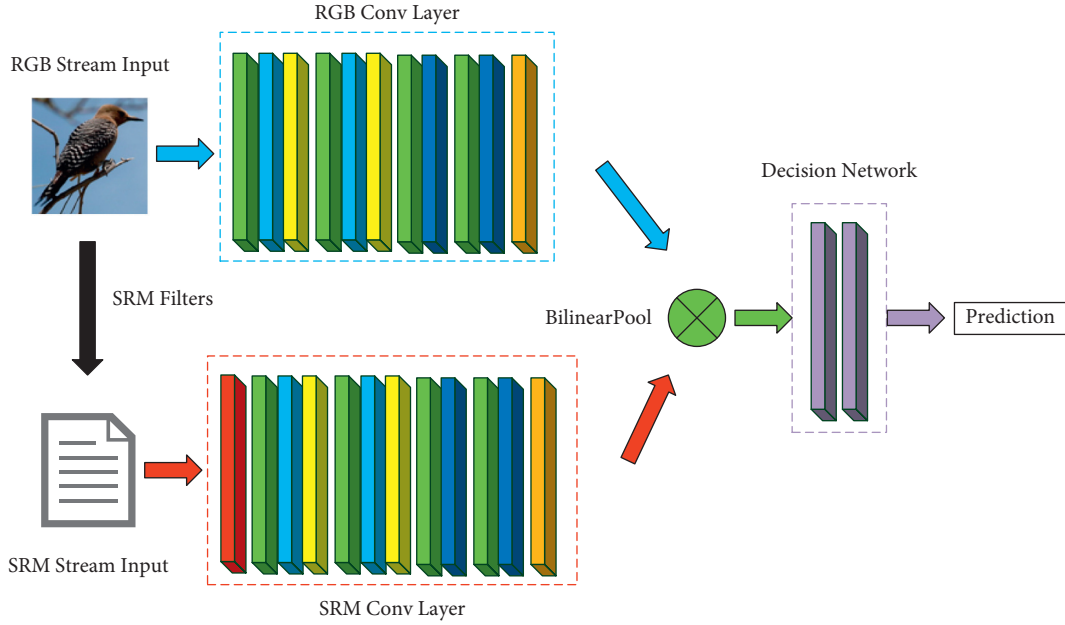


FIGURE 1: Illustration of our two-stream network (FNet). Color code used is as follows: light green = Conv, light blue = batchnorm + tanh, deep blue = batchnorm + ReLU, yellow = avg pooling, orange = max pooling, and purple = fully connected layers. The RGB stream uses original images as input and captures subtle difference like contrast difference and unnatural pixels from the RGB features. The noise stream first obtains noise feature maps through SRM filter layer and leverages the noise features to provide additional evidence for adversarial image detection. Bilinear pooling combines the features extracted by the two streams. Finally, passing the combined features through a decision network, the network generates the predicted label and determines whether the input image is adversarial or not.

the features extracted by the two CNNs. Finally, we input the final features into the decision network for classification. In addition, our trained model maintains a good detection rate against the adversarial examples generated by black-box models.

2.5. SRM Feature Extraction. Both adversarial attacks and steganography on images make perturbations on the pixel values, which alter the dependence between pixels. However, steganalysis can effectively detect modifications caused by steganography via modeling the dependence between adjacent pixels in natural images. So, we can also take advantage of steganalysis to identify deviations due to adversarial attacks. Inspired by the work of Liu [31] and Zhou [33], we decided to use the SRM filter to amplify the local noise disturbance of the image and use it as additional evidence to assist the decision-making network.

Although the work of Liu et al. [31] has used the SRM filter for adversarial sample identification and proved the effectiveness of the method, these works only consider the linear SRM noise features. In our work, we simulated the manual extraction method of SRM linear residuals. The linear residuals are obtained by convolving the image with a high-pass filter with a shift-invariant kernel. Specifically, we used 30 basic SRM filters and then convolved this convolution kernel with the original image to gather the basic noise features. The definition of residual filters is shown in Tables 1–3. Further, we divided the filters into five categories: first-order, second-order, third-order, SQUARE 3×3 +EDGE 3×3 , and SQUARE 5×5 +EDGE 5×5 . The

number of filters for each category is 8, 4, 8, 5, and 5, according to the different directions of pixel feature extraction. Specifically, for the first- and third-order, we used 8 filters to extract pixel features in eight directions $\{\uparrow, \downarrow, \leftarrow, \rightarrow, \nearrow, \swarrow, \nwarrow, \searrow\}$; for the second-order, EDGE 3×3 , and EDGE 5×5 , 4 filters were used to extract pixel features in four directions $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$; for SQUARE 5×5 and SQUARE 5×5 , we used 1 filter to extract pixel features. Based on the point view of more comprehensive characteristics of SRM's nonlinear residual statistical characteristics, we used the linear residual features obtained by SRM filters in the spatial domain to obtain nonlinear residual features by nonlinear processing.

We take the features extracted by the second-order SRM filter as an example to introduce how to obtain nonlinear residual features. First, we can predict pixel $Y_{i,j}$ from its horizontal or vertical neighboring pixels, thus obtaining 2 horizontal and 2 vertical residual. Then, we get 4 direction residuals: R_{ij}^{\rightarrow} , R_{ij}^{\leftarrow} , R_{ij}^{\uparrow} , and R_{ij}^{\downarrow} . Secondly, we can use these 4 residuals to compute 2 nonlinear “minmax” residuals as follows:

$$\begin{aligned} R_{ij}^{(\max)} &= \max(R_{ij}^{\rightarrow}, R_{ij}^{\leftarrow}, R_{ij}^{\uparrow}, R_{ij}^{\downarrow}), \\ R_{ij}^{(\min)} &= \min(R_{ij}^{\rightarrow}, R_{ij}^{\leftarrow}, R_{ij}^{\uparrow}, R_{ij}^{\downarrow}). \end{aligned} \quad (2)$$

The other four types of nonlinear residual features are calculated in the same way. In this way, the nonlinear residual features of the 10 channels are obtained. Finally, we get the linear features of 30 channels and the nonlinear

features of 10 channels. Then, we combine the nonlinear and linear features to get noise features of 40-channel as the input of the noise stream.

2.6. Two-Stream Network. We adopt a two-stream network with RGB stream and spatial rich model (SRM) noise stream to detect adversarial images. The model structure designed is shown in Figure 1. The role of the RGB input in our network is to catch significant disturbances. However, for tampered images that have been carefully processed to hide stitching boundaries and reduce contrast differences, it will be difficult to accomplish the task using RGB streams alone. In addition, when generating adversarial samples, we often use the L_p norm to restrict adversarial disturbances. Therefore, the generated adversarial sample disturbances are often invisible to the human eye. In this case, it is difficult to complete the detection of antagonistic samples only by relying on the RGB stream. Inspired by the work of Liu [31] and Zhou [33], we adopt SRM noise features as additional evidence to determine whether the input image is adversarial or not.

In our model, we used RGB stream to simulate visual tampering and detect image disturbance with large disturbance; SRM noise stream is used to extract and amplify noise features by SRM filters, which is used as the additional evidence for adversarial image detection (see Table 4 for the specific network structure). In the structure of the SRM noise stream network, we add an ABS layer to reduce the influence of symbols on the model decision. Then, we use bilinear pooling [23] to fuse the SRM noise stream and the RGB stream. Bilinear pooling is proposed for the fine-grained classifier. It has two CNN network structures, and the features extracted by the two CNN networks are fused through bilinear pooling. After the fused features pass through a decision network consisting of two fully connected layers, the final predicted result is obtained (see Figure 1). We use cross-entropy loss that leads to the following objective function:

$$L = L_{\text{cross}}(f_D(BP(f_{\text{RGB}}(x), f_N(f_{\text{SRM}}(x)))), y), \quad (3)$$

where x is the input image, y is x 's label, f_{SRM} denotes the SRM network with fixed weights, f_{RGB} and f_N are the RGB stream network and the noise stream network, BP denotes the bilinear pooling, f_D denotes the decision network, and L_{cross} denotes the cross-entropy loss.

The model accepts $32 * 32 * 3$ (width * height * channels) images and inputs two types of labels. This model is trained by the SGD optimizer. The parameters of the SGD optimizer are set as follows: momentum = 0.9 and weight_decay = 0. The hyperparameters of the network are set as follows: lr = 0.1 and batch_size = 64, and the training is designed to be 100 epoch, and lr is automatically changed to 0.1 times the original value every 30 epoch.

3. Experiments

In this section, we present an experimental evaluation of our method and compare it with several detection methods.

3.1. Experimental Setting

3.1.1. Dataset. We evaluate the detection method on the CIFAR-10 dataset. The CIFAR-10 dataset contains 60,000 32×32 color pictures, of which include 50,000 images for training and 10,000 images for testing.

For adversarial example datasets, we adopted three attack methods of FGSM, CW, and DeepFool. We used three methods to attack VGG16 [34], Resnet50 [35], and LeNet [36] and finally got 9 adversarial example datasets. For convenience, we named the adversarial example dataset generated by the VGG16 as "Adv-VGG16-Set." We used the Adv-VGG16-Set to train the two-stream network model (FNet) and baseline models. The datasets generated by attacking ResNet50 and LeNet are used for black-box testing to test the models trained on Adv-VGG16-Set. The parameter settings of the three attack methods are shown in Table 5. Previous work showed that nontargeted attack is easier to succeed, results in smaller perturbations, and transfers better to different models. So, we tested our method by nontargeted adversarial examples.

3.1.2. Classifier. For the CIFAR-10 dataset, we trained three models: VGG16 [34], ResNet50 [35], and LeNet [36]. These models were trained by the SGD optimizer (momentum = 0.9; weight_decay = 0), and the hyperparameters are set as follows: lr = 0.01, batch_size = 64, epoch = 30, and set lr to be multiplied by 0.1 times for every 10 epochs.

3.1.3. Baseline Models. We compared our method (FNet) with other detection methods including RGB-Net, SRM-Net, and KD + BU [32]. RGB-Net is a single-stream network that only inputs RGB images for judgment. It has the same network structure as the RGB stream part of the two-stream network we designed; similarly, SRM-Net only has the SRM part of our network. KD + BU [32] uses Bayesian uncertainty and model kernel density estimation to determine whether the sample is adversarial. For convenience, we use FNet to refer to our method.

3.1.4. Attack Methods. We adopted three attack methods from the Adversarial Robustness Toolbox designed by Microsoft [37]: FGSM [18], CW [27], and DeepFool [28]. For RGB-NET, SRM-NET, KD + BU, and FNet, we all used Adv-VGG16-Set for training. In training, we train a new detector using only one attack method each time.

3.1.5. Evaluation Metric. We use precision score, recall score, and area under ROC curve (AUC) score as the evaluation metric of the model [38]. The closer the AUC score is to 1, the larger the area under ROC curve and the better the model.

The calculation formula of the metric is as follows:

TABLE 4: The detailed two-stream network architecture for CIFAR-10. Conv (d, k, s) denotes the convolutional layer with d as dimension, k as kernel size, and s as stride.

RGB stream	SRM stream
Conv (64, 3, 1)	Conv (64, 3, 1)
BatchNorm layer, Tanh	ABSLayer
Avg pooling	BatchNorm layer, Tanh
Conv (128, 3, 1)	Avg pooling
BatchNorm layer, Tanh	Conv (128, 3, 1)
Avg pooling	BatchNorm layer, Tanh
Conv (256, 3, 1)	Avg pooling
BatchNorm layer, ReLU	Conv (256, 3, 1)
Conv (256, 3, 1)	BatchNorm layer, ReLU
BatchNorm layer, ReLU	Conv (256, 3, 1)
MAX pooling	BatchNorm layer, ReLU
	MAX pooling
	Full connected 4096, ReLU, dropout (0.5)
	Full connected 4096, ReLU, dropout (0.5)
	Softmax 2

TABLE 5: Parameter setting of three attack methods in adversarial robustness toolbox.

Attack method	Parameter
FGSM	Norm = 2, eps = 2.0, eps_step = 0.1
DeepFool	Eps = 0.1
CW	Lr = 0.2, confidence = 0.1

$$\text{precision} = \frac{TP}{TP + FP},$$

$$\text{recall} = \frac{TP}{TP + FN},$$
(4)

where TP denotes true positive rate, FP denotes false positive rate, and FN denotes false negative rate.

4. Experimental Results

On the CIFAR-10 dataset, Tables 6–8 show the precision and recall scores of different detection methods on the normal images and adversarial images. The bold values in tables represent the results of experiments conducted by our method (FNet). We can see that RGBNet and our method (FNet) have excellent effects in defending against both white-box attacks and black-box attacks. The precision score of FNet reaches 93.2% on adversarial images generated by DeepFool on VGG16. Experimental results show that it is difficult to detect adversarial examples generated by the CW method. SRM-Net is almost invalid against CW. KD + BU and RGB-Net achieve low scores when detecting CW. However, FNet improves KD + BU by more than 30%, and the precision score of FNet reaches 90.8% on detecting adversarial examples generated by CW. As we mention above, all detectors are trained on Adv-VGG16-Set. In addition, we can see that

our method with good transferability can well detect adversarial samples generated by black models. SRM-Net and KD + BU are almost invalid against adversarial examples generated on black models, while the precision score of FNet reaches 90.1% against CW.

Figure 2 shows ROC curves of detection methods on CIFAR-10. The figures in the first row show the detector performance of the three attack methods of different detection methods on the adversarial samples generated by the white-box model (VGG16). We can see that the AUC score of FNet is from 0.963 to 0.976, and the AUC score of KD + BU is 0.795 to 0.837. The figures show that it is difficult to detect CW attack. The AUC score of SRM-Net is 0.525, and the AUC score of KD + BU is 0.795. Our method achieves the best performance when detecting the adversarial samples generated by the white-box model. The figures in the second and third rows show the detector performance of different methods on the adversarial samples generated by the black-box model. Experimental results show that the AUC score of FNet reaches 0.954. KD + BU and SRM-Net achieve low AUC score when detecting samples generated by the black-box model. The best AUC score of KD + BU is 0.715 when detecting DeepFool attack on ResNet.

Combining precision, recall, and AUC scores, RGB-Net performs second to our method (FNet) and KD + BU ranks third. In Figure 2, we can clearly see that the performance of FNet is better than that of other detection

TABLE 6: Performance of normal images and their adversarial examples generated by FGSM on CIFAR-10.

Model	Method	Normal images		Adv images		
		Precision	Recall	Precision	Recall	
White model	VGG16	RGB-Net	0.896	0.928	0.864	0.807
		SRM-Net	0.748	0.773	0.571	0.538
		KDBU [32]	0.902	0.643	0.580	0.876
		FNet	0.926	0.926	0.868	0.868
Black model	ResNet	RGB-Net	0.888	0.928	0.874	0.809
		SRM-Net	0.692	0.773	0.543	0.440
		KDBU [32]	0.648	0.643	0.426	0.431
		FNet	0.912	0.926	0.879	0.854
	LeNet	RGB-Net	0.919	0.928	0.821	0.801
		SRM-Net	0.731	0.773	0.359	0.309
		KDBU [32]	0.697	0.643	0.269	0.319
		FNet	0.927	0.926	0.819	0.822

The bold values represent the results of experiments conducted by our method (FNet).

TABLE 7: Performance of normal images and their adversarial examples generated by CW on CIFAR-10.

Model	Method	Normal images		Adv images		
		Precision	Recall	Precision	Recall	
White model	VGG16	RGB-Net	0.912	0.856	0.843	0.903
		SRM-Net	0.539	1.000	0.000	0.000
		KDBU [32]	0.852	0.525	0.617	0.893
		FNet	0.913	0.922	0.908	0.898
Black model	ResNet	RGB-Net	0.916	0.856	0.840	0.906
		SRM-Net	0.544	1.000	0.000	0.000
		KDBU [32]	0.545	0.525	0.457	0.478
		FNet	0.883	0.922	0.901	0.855
	LeNet	RGB-Net	0.943	0.856	0.792	0.914
		SRM-Net	0.625	1.000	0.000	0.000
		KDBU [32]	0.589	0.525	0.330	0.390
		FNet	0.903	0.922	0.865	0.835

TABLE 8: Performance of normal images and their adversarial examples generated by DeepFool on CIFAR-10.

Model	Method	Normal images		Adv images		
		Precision	Recall	Precision	Recall	
White model	VGG16	RGB-Net	0.913	0.861	0.848	0.905
		SRM-Net	0.766	0.776	0.734	0.724
		KDBU [32]	0.857	0.526	0.619	0.898
		FNet	0.908	0.944	0.932	0.888
Black model	ResNet	RGB-Net	0.917	0.861	0.845	0.907
		SRM-Net	0.649	0.776	0.650	0.498
		KDBU [32]	0.554	0.526	0.466	0.495
		FNet	0.868	0.944	0.926	0.828
	LeNet	RGB-Net	0.940	0.861	0.806	0.913
		SRM-Net	0.661	0.776	0.510	0.370
		KDBU [32]	0.577	0.526	0.341	0.390
		FNet	0.881	0.944	0.900	0.798

The bold values represent the results of experiments conducted by our method (FNet).

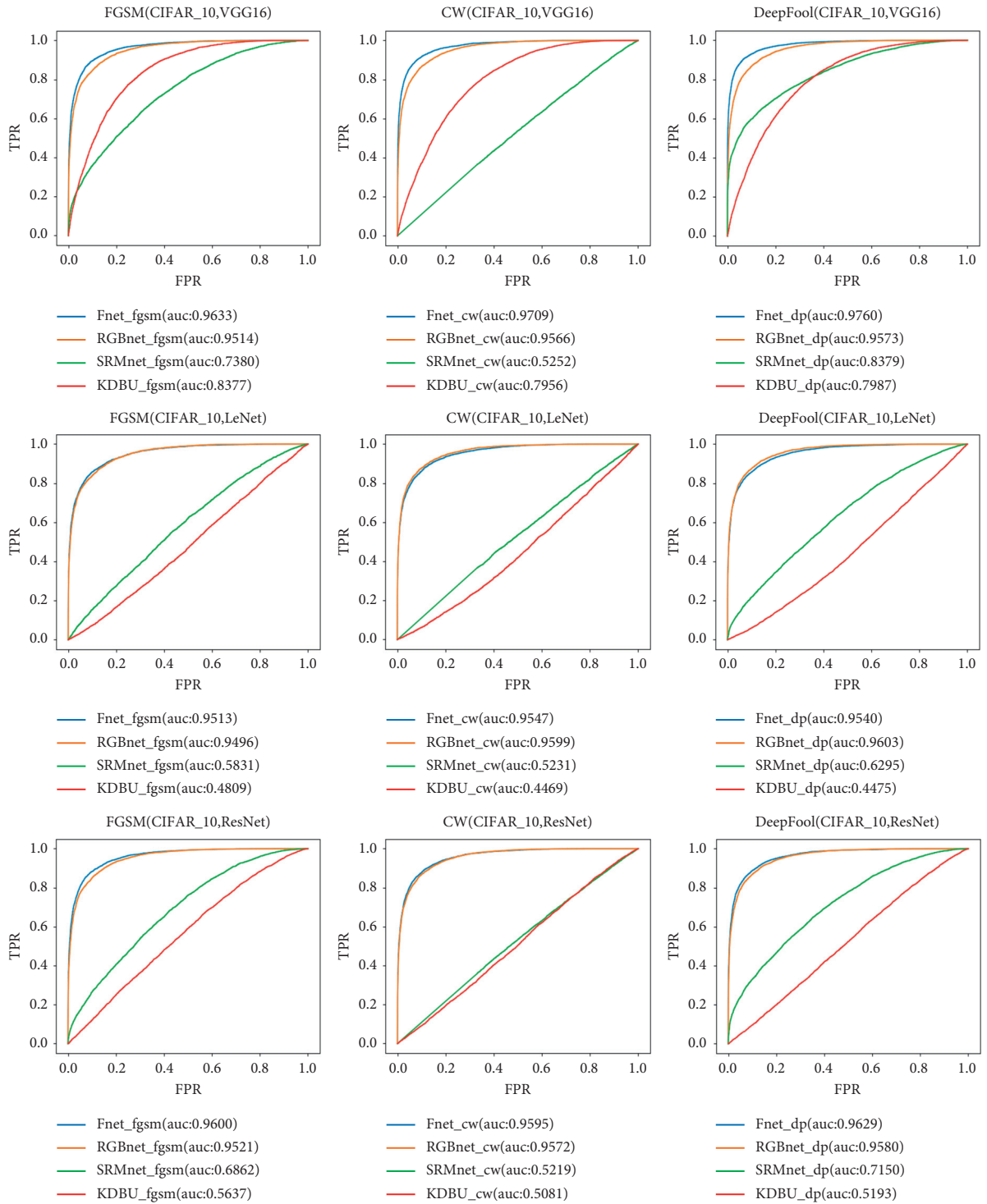


FIGURE 2: ROC curves of detection methods on CIFAR-10. We choose to display the ROC curves of detection methods on three different attacks. We can intuitively see that our method (FNet) is better than other methods in all cases.

methods in all cases. RGB-Net and KD + BU perform well when defending against white-box attacks, but they do not perform well against black-box attacks.

5. Conclusion

In this paper, we propose a two-stream model including RGB stream and SRM residual stream to improve the security of deep learning services based on 5G. We use 30 SRM filters to extract linear noise features and perform nonlinear processing to obtain rich features of 40 channels. We input these noise features into the network as additional evidence for detection. Experiments show that our method performs well against both black-box and white-box attacks. Moreover, our method has good transferability.

Although our method is effective to detect adversarial examples on images, the method is not effective on all types of data. The spatial rich model (SRM) feature extraction method is suitable for image data. Our two-stream network adopts SRM steganalysis to obtain features as additional evidence for adversarial detection, which is only effective for images. Therefore, our method is only effective for deep learning services which provide services such as image recognition. In the future work, we will explore more effective methods of adversarial sample detection and recovery.

Data Availability

The experiment data used to support the findings of this study are included within the article. The experiment data are described in Section 4 in detail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially sponsored by the National Key R&D Program of China (No. 2019YFB2101700), the National Science Foundation of China (Nos. 62172297 and 61902276), the Key Research and Development Project of Sichuan Province (No. 21SYSX0082), Tianjin Intelligent Manufacturing Special Fund Project (No. 20201159), Natural Science Foundation of Tianjin City grant (No. 19JCQNJC00200), and the Australian Research Council Linkage Project (No. LP190100676).





References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [2] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [3] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, Article ID 113240, 2020.
- [4] Z. Zhang, J. Geiger, J. Pohjalainen, A. Mousa, and B. Schuller, "Deep learning for environmentally robust speech recognition: an overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 1–28, 2018.
- [5] D. Amodei, S. Ananthanarayanan, and R. Anubhai, "Deep speech 2: end-to-end speech recognition in English and Mandarin," in *Proceedings of the International Conference on Machine Learning*, pp. 173–182, PMLR, New York, NY, USA, June 2016.
- [6] J. Guo, H. He, H. Tong et al., "GluonCV and GluonNLP: deep learning in computer vision and natural language processing," *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.
- [7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [8] R. G. Lipsey, K. I. Carlaw, and C. T. Bekar, *Economic Transformations: General Purpose Technologies and Long-Term Economic Growth*, OUP Oxford, Oxford, UK, 2005.
- [9] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks," 2017, <https://arxiv.org/abs/1710.02913>.
- [10] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE access*, vol. 6, pp. 32328–32338, 2018.
- [11] K. Jia, M. Kenney, J. Mattila, and T. Seppala, "The application of artificial intelligence at Chinese digital platform giants: baidu, alibaba and tencent," *ETLA Reports*, vol. 81, 2018.
- [12] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q. L. Han, "Deep learning-based autonomous driving systems: a survey of attacks and defenses," *IEEE Transactions on Industrial Informatics*, vol. 17, 2021.
- [13] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, Pisa, Italy, May 2020.
- [14] G. Xu, G. H. Xin, L. Jiao et al., "A semi-black-box android adversarial sample attack framework against DLAAS," 2021, <https://arxiv.org/abs/2105.11593>.
- [15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, <https://arxiv.org/abs/1503.02531>.
- [16] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," 2016, <https://arxiv.org/abs/1608.00853>.
- [17] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, 2018.
- [18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, <https://arxiv.org/abs/1412.6572>.
- [19] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

- [20] J. Fridrich and M. Goljan, "Practical steganalysis of digital images: state of the art," *International Society for Optics and Photonics*, vol. 4675, pp. 1–13, 2002.
- [21] N. F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, 1998.
- [22] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2018.
- [23] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457, Santiago, Chile, December 2015.
- [24] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp. 262–275, Springer, New York, NY, USA, July 2017.
- [25] P. Madani and N. Vljajic, "Robustness of deep autoencoder in intrusion detection under adversarial contamination," in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security*, pp. 1–8, Raleigh, NC, USA, April 2018.
- [26] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," 2013, <https://arxiv.org/abs/1312.6199>.
- [27] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, San Jose, CA, USA, May 2017.
- [28] S.-M. Moosavi-Dezfooli, A. Fawzi, and F. Pascal, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.
- [29] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, <https://arxiv.org/abs/1607.02533>.
- [30] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, San Jose, CA, USA, May 2016.
- [31] J. Liu, W. Zhang, Y. Zhang et al., "Detection based defense against adversarial examples from the steganalysis point of view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4825–4834, Long Beach, CA, USA, June 2019.
- [32] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, <https://arxiv.org/abs/1703.00410>.
- [33] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1053–1061, Salt Lake City, UT, USA, 2018.
- [34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [37] M.-I. Nicolae, M. Sinn, M. N. Tran et al., "Adversarial robustness toolbox v1. 0.0," 2018, <https://arxiv.org/abs/1807.01069>.
- [38] N. Carlini, A. Athalye, N. Papernot et al., "On evaluating adversarial robustness," 2019, <https://arxiv.org/abs/1902.06705>.

Research Article

An Enhanced Visual Attention Siamese Network That Updates Template Features Online

Wenqiu Zhu ^{1,2}, Guang Zou ^{1,2}, Qiang Liu ^{1,2} and Zhigao Zeng ^{1,2}

¹College of Computer Science, Hunan University of Technology, Zhuzhou, Hunan 412007, China

²Intelligent Information Perception and Processing Technology, Hunan Province Key Laboratory, Zhuzhou, China

Correspondence should be addressed to Guang Zou; 591891549@qq.com and Qiang Liu; liuqiang@hut.edu.cn

Received 1 May 2021; Revised 26 July 2021; Accepted 15 August 2021; Published 31 August 2021

Academic Editor: Yuan Yuan

Copyright © 2021 Wenqiu Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, Siamese trackers have attracted extensive attention because of their simplicity and low computational cost. However, for most Siamese trackers, only a frame of the video sequence is used as the template, and the template is not updated in inference process, which makes the tracking success rate inferior to the trackers that can update the template online. In the current study, we introduce an enhanced visual attention Siamese network (ESA-Siam). The method is based on a deep convolutional neural network, which integrates channel attention and spatial self-attention to improve the discriminative ability of the tracker for positive and negative samples. Channel attention reflects different targets according to the response value of different channels to achieve better target representation. Spatial self-attention captures the correlation between two arbitrary positions to help locate the target. At the same time, a template search attention module is designed to implicitly update the template features online, which can effectively improve the success rate of the tracker when the target is interfered by the background. The proposed ESA-Siam tracker shows superior performance compared with 18 existing state-of-the-art trackers on five benchmark datasets including OTB50, OTB100, VOT2016, VOT2018, and LaSOT.

1. Introduction

Visual object tracking is a process of identifying the region of interest in the video, which can track the target in a given video. At present, visual object tracking is widely used in video surveillance [1], automatic driving [2], UAV tracking [3], and other fields. Although various researchers have done a lot of work on tracker to improve its performance, target tracking still faces such practical problems as fast motion, similar background interference, target scale transformation, low image resolution, and so on [4, 5].

The naive correlation filter tracker uses hand-crafted features, such as KCF [6], SRDCF [7], CACF [8], DSST [9], and SAMF [10]. Compared with the method of end-to-end learning using deep convolutional neural networks (CNNs), it is much inferior.

Recently, deep learning has been widely used in visual tracking. Trackers use CNNs to extract target features, and the tracking success rate and robustness are significantly

improved. SiamFC [11] extracts template features and search features through AlexNet [12], uses the similarity measure method to perform cross-correlation operation on the extracted features to obtain the final response graph, and then predicts the target location according to the score of the response graph. Because the network model is simple and uses the offline pretraining network model, there is no online update and no complex calculation. Compared with the traditional online update method of correlation filter, SiamFC is faster and can meet the real-time requirements. Based on the Siamese network system, lots of trackers with state-of-the-art performance were proposed, such as SiamRPN [13], SiamMask [14], SiamRPN++ [15], and DaSiamRPN [16]. There are also some additional methods to build tracker based on different angles, such as thermal infrared [17], self-supervised [18], and focusing target regression model [19].

Siamese architecture has been applied in various fields of artificial intelligence, such as one-shot image

recognition [20], human reidentification [21], sentence similarity [22], and visual object tracking [23]. For visual object tracking, Siamese-based trackers train offline based on a large amount of data but do not update the target template online. Therefore, in the face of severe target deformation, scaling, occlusion, and other scenes, the target will be lost, resulting in the performance and robustness degradation of the tracker. In addition, the features extracted by CNN do not distinguish the weight in channel and space, and we know that different channel features correspond to different target information. The response channels that represent the tracking target should be given more weight, and not all of them should be given the same weight. The visual attention mechanism [24–26] can pay attention to the channel and location of interest and screen out the feature information that can represent the tracking target better. Based on this, we design a novel visual tracking method which can update the target online by enhancing the hybrid visual attention.

Inspired by the application of visual attention mechanism in RASNet [27] and EFCTA [28], we propose an enhanced visual attention Siamese network referred to as ESA-Siam. Considering that the information of search branch and template branch is mutually compensated, the context information of the search branch is also important. Combining the target information of the search branch can help the tracker identify positive and negative samples better. Therefore, we design a template search collaborative attention module, called T-SCAttn, which can update the template features online. It can improve the robustness and the positive and negative sample discrimination of the tracker and better deal with the problems of low image resolution and target occlusion. The main contributions of our work are as follows:

- (1) We introduce a new twin network visual tracking algorithm based on the enhanced visual attention mechanism (including channel attention, spatial self-attention, and template search collaborative attention). Channel attention distinguishes background and targets according to different target response values, and spatial self-attention aggregates nonlocal context information to help target location better.
- (2) We design a template search collaborative attention module which can update the template features online by recalculating the template images and search images.
- (3) We change the traditional pooling layer. Based on this, we propose golden threshold stochastic pooling to activate the target features with a higher probability and ignore other background features.
- (4) Our approach in the benchmark datasets OTB50 [29], OTB100 [30], VOT2016 [31], VOT2018 [32], and LaSOT [33] has excellent tracking performance, the tracking of which can reach speeds of up to 60 fps.

2. Related Work

Since MOSSE [34], trackers based on correlation filtering have been widely used due to the convenience and simplicity of the hand-crafted features. Such methods can update targets online and have high accuracy. However, due to its simple feature, the robustness is poor when the target is blocked and the appearance is deformed. The depth features based on CNNs can more fully express the target features. As a result, a number of tracking methods that combine related filtering and depth features have emerged, such as C-COT [35], CFNet [36], MDNet [37], DeepSRDCF [38], and ECO [39], to achieve better tracking performance. CFNet combines correlation filtering with SiamFC to win the VOT2017 real-time challenge and introduces a cyclic displacement matrix in SiamFC to improve performance. MDNet proposes a multidomain learning model based on CNN to distinguish multiple different independent targets.

In recent years, the current branch of building trackers is based on the Siamese network system. Since the SiamFC was proposed, more tracking methods based on this Siamese network have been proposed. Li et al. introduced candidate regions for target detection and proposed SiamRPN to treat the tracking task as a two-stage task: one is target classification and the other is target regression. Wang et al. combined target tracking with image segmentation; Siam-Mask segmented the target through a mask and completed the image segmentation while completing the target extraction. Zhu et al. proposed DaSiamRPN to effectively control the sampling strategy on the basis of SiamRPN, balanced the distribution of positive and negative samples, and improved the tracking performance. SA-Siam [40] uses two Siamese networks: one is to extract the semantic branch of high-level features of the target and the other is to extract the appearance branch of low-level features of the target; the network branches are trained separately and feature fusion is performed to improve the robustness of the tracker. Recently, Li et al. proposed SiamRPN++, using the deeper network ResNet50 [41] as the backbone network, analyzed the reason the Siamese network system cannot use the deep network, and further improved the tracking performance. However, since there is no online update, Siamese-based trackers are easily interfered by target occlusion and complex background.

Recently, with the widespread application of the visual attention mechanism in the field of computer vision, Hu et al. proposed SENet [42], which gives weights to different channels by squeeze and excitation channels and statistically the global information of the image at the characteristic channel level, selecting features in a targeted manner. Shaw et al. proposed novel self-attention, which pays more attention to the correlation between internal feature elements and obtains the global dependence of any two positions in the feature map. Wang et al. proposed a generalized and simple nonlocal block [43] that can be directly embedded in the network, which can capture time and space information for integration. Particularly, Wang et al. combined the visual attention mechanism to propose RASNet, which separated feature learning and discriminant analysis and used

cross-correlation to update the target to enhance the ability to distinguish between target and background. However, only using the feature information of the target, the distinction between the target and the background is not enough, and it is impossible to face complex scenes.

3. Proposed Method

The overall framework of the enhanced visual attention Siamese network is shown in Figure 1. Compared with other Siamese-based trackers, ESA-Siam uses a template and search area to coordinate the attention block to update the target online implicitly to adapt to changes in the target's appearance. Second, our network uses a golden threshold stochastic pooling layer to activate target features with greater probability. Finally, we use the channel attention mechanism and spatial self-attention to filter feature maps and combine the correlation between global context information and local features to help locate targets and estimate target contours. The following sections describe in detail the various components of the proposed tracker.

3.1. Siamese-Based Trackers. The key point of the SiamFC tracking algorithm is the use of offline training and online fine-tuning of the network, which can effectively improve the speed of the algorithm. Its network structure is composed of a template branch and a search branch, and the two branches extract features through the same shared network (AlexNet). We perform cross-correlation of the extracted two branch features to calculate the feature similarity and locate the target according to the similarity value. The position with high similarity is the target position. When a full convolutional network is used, the size of the search image does not need to be consistent with the template image, which can provide a larger search area for the network and calculate the similarity of more subwindows. The cross-correlation function is shown in the following formula:

$$f(z, x) = \varphi(z) * \varphi(x) + b_1, \quad (1)$$

where x is the input search image, z is the input template image, $\varphi(\cdot)$ is the feature extraction network, $*$ represents the convolution operation, b represents the offset of each position in the score map, and $f(z, x)$ represents the similarity score map between the template feature and the search feature. The position with the highest score is the target position.

3.2. Golden Threshold Stochastic Pooling. Zeiler and Fergus combined the advantages and disadvantages of maximum pooling and average pooling to propose stochastic pooling [44]. Zeiler believes that the maximum pooling is always to select the largest activation from the pooling area every time, completely excluding other activations except the maximum value. Stochastic pooling applies multinomial distribution and calculates the score probability of each response position to randomly select activation. In this way, nonmaximum activations could also be selected. We calculate the

probability of each position i by normalizing the area activation, as shown in the following formula:

$$P_i = \frac{a_i}{\sum_{k \in R_j} a_k}, \quad (2)$$

where a_i is the activation value of position i and R_j represents the area j in the feature map. Multinomial distribution selects a location i within the region:

$$s_j = a_i, \quad \text{where } l \sim P\left(p_1, \dots, p_{|R_j|}\right), \quad (3)$$

where s_j represents the final activation of region j , which is randomly selected by the probability calculated through each position of region j . The activation with the greater probability is more likely to be selected. Although this can ensure that the information is not lost to a certain extent, because of its randomness, it is possible to select a value with a small activation probability and lose important information. In the target tracking task, we should try to avoid this uncertainty. Therefore, we improve on the basis of stochastic pooling, sort the calculated probabilities, and filter out some activations with too small probability values by setting a threshold T (e.g., $T = 0.002$).

$$s_j = a_i, \quad \text{where } l \sim P\left(p_1, \dots, p_{|R_j|}\right) > T. \quad (4)$$

The selection of T is set according to the ratio of the maximum activation part (e.g., $T = 0.618P_{\max}$). Meanwhile, to make reasonable use of the advantages of the maximum pooling layer to highlight important information, we pay more attention to the top ranked by the activation value, so that the random selection can fall in this range with a high probability, and weaker activations are inhibited. An example of golden threshold stochastic pooling is shown in Figure 2. The backpropagation process is similar to the maximum pooling backpropagation, and only the value of the position of the selected node that has been recorded by the forward propagation is retained, as shown in the following formula:

$$\frac{\partial L}{\partial x_i} = \begin{cases} 0, & \delta(i, j) = \text{false}, \\ \frac{\partial L}{\partial y_j}, & \delta(i, j) = \text{true}, \end{cases} \quad (5)$$

where x_i, y_j are the input node and output node and $\delta(i, j)$ is the decision function, which represents whether the input node i is selected as the maximum output by the output node j .

3.3. Channel Attention Module. According to the characteristics of the target tracking task, we designed an enhanced attention mechanism, as shown in Figure 3, which consists of a spatial self-attention module, a channel attention module, and a template search collaborative attention module. The spatial attention module is based on the correlation dependency structure of the pixels at the same

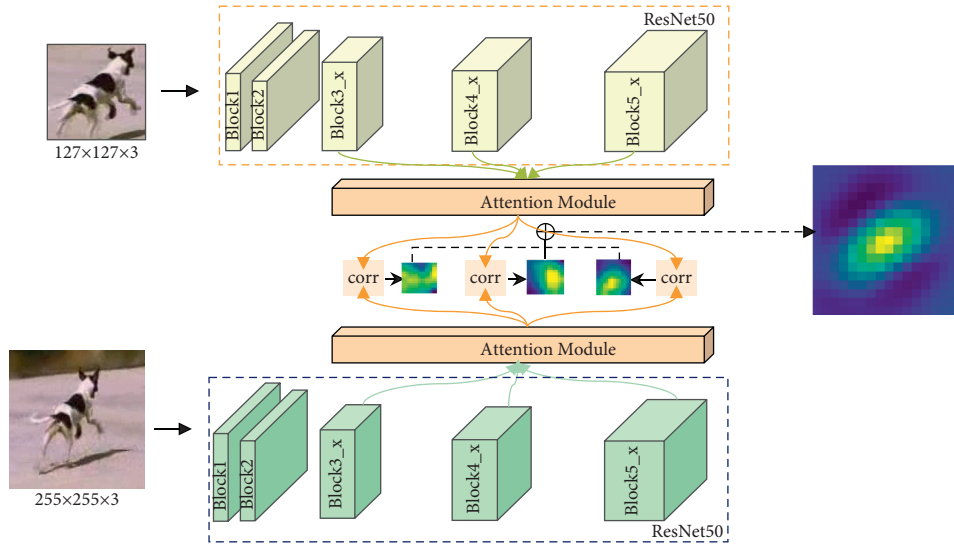


FIGURE 1: ESA-Siam network framework. Based on the Siamese benchmark network framework, ESA-Siam uses ResNet50 as the backbone network to do attention screening on the last three network blocks of the template branch and the search branch. After that, cross-correlation operations are performed on the template features and their respective search features, and then the fusion features are performed to obtain the final output feature map.

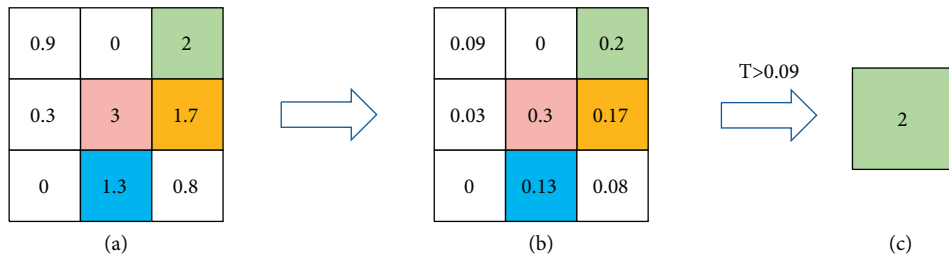


FIGURE 2: Example of golden threshold stochastic pooling. (a) Activation within a given pooling region. (b) Probability of activation. (c) Sampled activation. If the threshold $T > 0.09$ is set, the probability of selecting a nonmaximum value will increase.

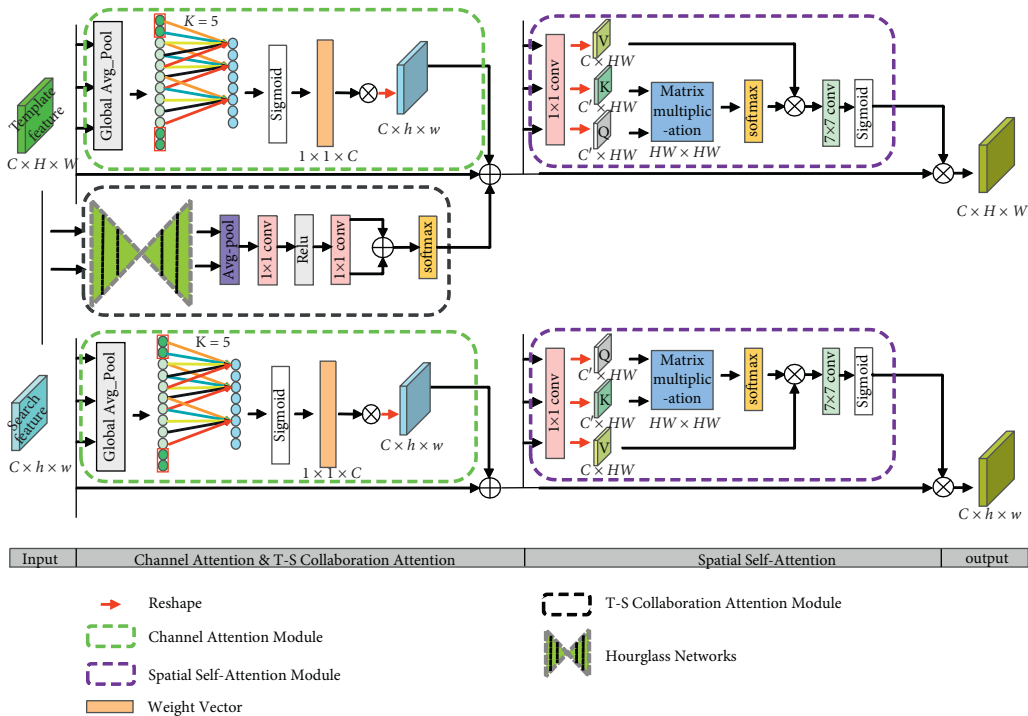


FIGURE 3: The details of the attention module. It consists of three parts, a channel attention module, a spatial self-attention module, and a template search collaborative attention module. The features extracted through the backbone network are input to the channel attention and T-SCAttn, and the features generated are finally input to the spatial self-attention module for optimization and improvement.

location in the feature map to characterize features. Spatial self-attention mechanism can capture the relationship between internal data and features to establish a correlation between any two locations. The feature of particular location can be weighted and summed through all location feature information. Channel attention can distinguish targets by the response of different channels to different targets. A channel with a high response value may represent the same target, and a higher response weight will be given, while a lower response weight will be given, so as to adjust the characteristic response adaptively. The template search collaborative attention module captures nonlocal semantic feature information globally and updates template features through the hourglass network [45].

The traditional channel attention mechanism uses a multilayer perceptron (MLP) method to calculate the weight of each channel. This method increases a great number of parameters due to the use of a large number of fully connected layers, reduces the calculation speed, and affects the real-time performance of the algorithm. We designed and introduced the ECA module in ECA-Net [46] to avoid the negative impact of dimensionality reduction using a fully connected layer, and at the same time, proper cross-channel interaction can significantly reduce the model parameters. This strategy adopts one-dimensional convolution to realize and uses the feature of convolution operation weight sharing. The size of the convolution kernel k in one-dimensional convolution is obtained through adaptive calculation. The specific calculation formula is as follows:

$$k = \psi(C) = \left\lfloor \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rfloor. \quad (6)$$

In general, the number of channels is always the power of 2. We set $r=2$, $b=1$. Through the adaptive convolution kernel size k to complete the cross-channel information interaction, so that the layer with a larger number of channels can interact more between channels.

Compared with using multilayer perceptrons to connect to each other, the parameter number is significantly reduced, to ensure the real-time nature of the algorithm. As shown in Figure 3, the channel attention module (C-Attn) squeezes the input feature map F , and after global average pooling, a feature vector $f = (f_1, f_2, \dots, f_c)$ is obtained as the input of the one-dimensional convolutional layer, where $f_i \in \mathbb{R}$. Then, we get the weight vector $P = (p_1, p_2, \dots, p_c)$ from the sigmoid function, where $p_i \in \mathbb{R}$. The input feature F is elementwise multiplied with the weight vector P . Finally, we get the features $F_A^C \in \mathbb{R}^{C \times h \times w}$ filtered by the channel attention.

3.4. Spatial Self-Attention Module. The self-attention space module is a supplement to channel attention, as shown in Figure 3. Channel attention and spatial self-attention work in series. The output of the channel attention is the input of the spatial self-attention module. $F_A^C \in \mathbb{R}^{C \times h \times w}$ is input to an independent 1×1 convolution and passes through three convolution functions to obtain three feature vectors $Q, K \in \mathbb{R}^{C \times HW}$, $V \in \mathbb{R}^{C \times HW}$. We transpose vector Q and

then perform matrix multiplication with vector K . We can generate a spatial self-attention feature map through the columnwise softmax operation as follows:

$$\beta_{i,j} = \exp \frac{(Q_i^T \cdot K_j)}{\sum_{i=1}^{WH} \exp(Q_i^T \cdot K_j)}, \quad (7)$$

where $\beta_{i,j}$ represents the weight between the i -th location region and the j -th location region. The result $\beta_{i,j}$ is elementwise multiplied with vector V . Then, we performed a 7×7 convolution operation. A sigmoid activation is performed for generating a feature vector with weights $\Omega = (\omega_1, \omega_2, \dots, \omega_c)$, where $\omega_i \in \mathbb{R}^{C \times HW}$. After that, the input feature is elementwise multiplied with Ω . Finally, we get the final output feature $F_A^S \in \mathbb{R}^{C \times h \times w}$ with high similarity to the target by the following formula:

$$X_A^S = \alpha \Omega F, \quad (8)$$

where α is a hyperparameter. We initialize it to 0.0001 and then gradually increase it to give more weight, which can adapt to simple tasks at the beginning and face more complex tasks later.

3.5. Template Search Collaborative Attention Module (T-SCAttn). We designed a template search collaborative attention (T-SCAttn) module to implicitly update template features. We combine the context information of the target in the search image with the template feature and use the context information to improve the accuracy of target positioning. Search branch and template branch are complementary. T-SCAttn is composed of two components, where one is used to perform multiscale information interaction between template features and search features extracted by the backbone network. We use the stacked hourglass network (as shown in Figure 4) to generate multiscale template information X_t^{T-S} and multiscale search information X_s^{T-S} . The hourglass network does not change the size of the feature map. Another component is used to perform attention filtering on features. Inspired by CBAM, we only use global average pooling and one-dimensional convolution to get the context information of the feature map. We first perform a 1×1 convolution to reduce the number of channels to one channel. Then, we apply the ReLU activation function and one-dimensional convolution to filter the context information feature map and apply the softmax layer. The result of the T-SCAttn is elementwise added with the output of channel attention module and input feature. We can get it according to the following formula:

$$X^{T-S} = \text{Softmax}(\text{AvgPool}(X_t^{T-S}) + \text{AvgPool}(X_s^{T-S})), \quad (9)$$

where X^{T-S} is the output of the T-SCAttn module.

3.6. Network Structure and Algorithm. The proposed network is based on the Siamese network and an enhanced attention mechanism. The proposed network framework can be described as

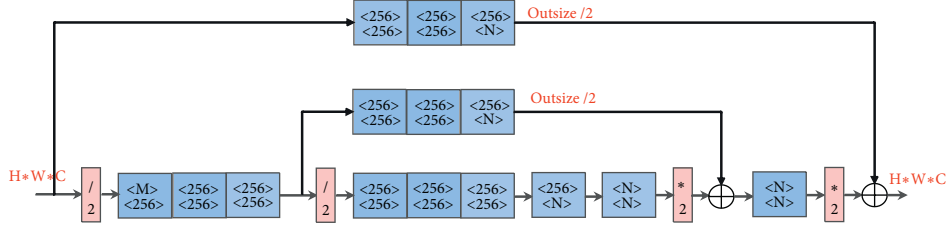


FIGURE 4: Structure of stacked hourglass networks. The size of input feature map is $H * W * C$. $/2$ represents downsampling, and $*2$ represents upsampling, using nearest neighbor upsampling. The upper $< >$ content in the blue box represents the number of input channels, and the lower $< >$ represents the number of output channels. \oplus represents elementwise addition. The size of the output feature map is $H * W * C$. Stacked hourglass network has nothing to do with the input size, and it only needs to provide the number of input and output channels. Also, it can gradually extract deeper features.

$$m(T, S) = [\eta(\mu(\psi(Z)) + \Delta(Z, X))] * [\eta(\mu(\psi(X)))] + b, \quad (10)$$

where Z is a template image, X is the search patch, $\psi(\cdot)$ represents the backbone network, $\mu(\cdot)$ denotes the channel attention module, $\eta(\cdot)$ denotes the spatial self-attention module, and $\Delta(\cdot)$ represents the template search collaborative attention module. The output of $\psi(\cdot)$ is fed to the channel attention module as $\mu(\psi(Z))$. Then, the output forwards input to the spatial self-attention module as $\eta(\mu(\psi(Z)))$.

The backbone network uses ResNet50, and the network structure and the corresponding operations of each layer are shown in Table 1. The network is divided into five blocks, and the number of residual blocks from the second block to the fifth block is (3, 4, 6, 3). To avoid the resolution of the feature map extracted by the network from being too small, the last three blocks are not downsampled but replaced by the dilated convolution, which increases the receptive field while not making the feature map resolution too small. We train on positive and negative samples to construct a loss function. We get the optimal model parameters by minimizing the loss function. A positive sample is expressed as a point that does not exceed a certain pixel distance from the center. If one sample point exceeds the distance range, it is treated as a negative sample. The single point loss function is defined as shown in the following formula:

$$l(y, v) = \log(1 + \exp(-yv)), \quad (11)$$

where $y \in (+1, -1)$ indicates the ground-truth label of the sample and v represents actual score of the template image and search image. We use the average loss value of all location points to represent the loss during training as shown in the following formula:

$$L(y, v) = \frac{1}{D} \sum_{u \in D} l(y[u], v[u]), \quad (12)$$

where D represents the score map, u is the search position, and $v[u]$ represents the score for each position. We use stochastic gradient descent (SGD) during training to find the global minimum of the loss function as shown in the following formula:

$$\arg \min_{\theta} E(L[y, f(z, x, \theta)]), \quad (13)$$

where θ refers to the network parameters and E represents the mathematical expectation.

We also describe the training algorithm and testing algorithm of the proposed network framework, as shown in Algorithms 1 and 2.

4. Experiments and Results

We evaluate the proposed tracker algorithm ESA-Siam on five benchmark datasets, including OTB50, OTB100, VOT2016, VOT2018, and LaSOT. We compared 18 state-of-the-art tracking methods, including SiamDW [47], DSiam, HCF [48], CSR-DCF [49], GradNet [50], Staple [51], fDSST [52], UpdateNet [53], RASNet, SAME, SiamRPN, DeepSRDCF, SRDCF, CFNet, MDNet, C-COT, ECO, and SiamFC.

4.1. Implementation Details. We use ResNet50 trained offline on GOT10K [54] as the backbone network. The GOT10K dataset contains more than 10,000 video clips of real moving objects and more than 1.5 million manually labeled bounding boxes, covering more than 560 categories. According to SiamFC, we set the search image size during training and testing to $127 \times 127 \times 3$ and the template image size to $255 \times 255 \times 3$. We use stochastic gradient descent (SGD) optimizer with momentum set to 0.9 to minimize equation (13). During training, the initial learning rate is set to 0.01, the L2 penalty item (weight_decay) is set to $5e-4$, and the learning rate is exponentially decayed until 10^{-5} . The batch_size is 8, and the training epochs are 50. We set the threshold $T = 0.618P_{\max}$ in equation (4) and use three scale ratios [0.9638, 1, 1.0375] to scale the search patch. We set the initial value of the hyperparameter in equation (8) to 0.0001 and then increase it to $10^{-1} \sim 10^{-2}$ exponentially. Our method is implemented based on Python3.8, Cuda10.2, and Pytorch1.6. The experiment was performed on a machine with a CPU model of Intel(R)Core(TM)i5-9400F CPU @2.90 GHz, a graphics card of NVIDIA GeForce RTX 2070s, and a memory of 32 GB DDR4 RAM. The average tracking speed of the proposed tracker was 60 frames per second (FPS). The loss change during training is shown in Figure 5. The Y-axis is the loss value, and the X-axis is the number of training batches.

TABLE 1: Network structure and operations corresponding to each network block (block represents network block, Gold-SPool represents golden stochastic pooling, dilation represents dilated convolution, ResNet in Figure 1 includes Block1, Block2, Block3, Block4, and Block5, and “—” represents no operation).

Block	Operation	Template size	Search size
—	—	$127 \times 127 \times 3$	$255 \times 255 \times 3$
Block1	$7 \times 7, 64, 3 \times 3$ Gold-SPool, $s=2$	$31 \times 31 \times 64$	$62 \times 62 \times 64$
Block2	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$15 \times 15 \times 256$	$31 \times 31 \times 256$
Block3 + dilation	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$15 \times 15 \times 512$	$31 \times 31 \times 512$
Attention	—	$15 \times 15 \times 256$	$31 \times 31 \times 256$
Block4 + dilation	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$15 \times 15 \times 1024$	$31 \times 31 \times 1024$
Attention	—	$15 \times 15 \times 512$	$31 \times 31 \times 512$
Block5 + dilation	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 024 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$15 \times 15 \times 2048$	$31 \times 31 \times 2048$
Attention	—	$15 \times 15 \times 1024$	$31 \times 31 \times 1024$

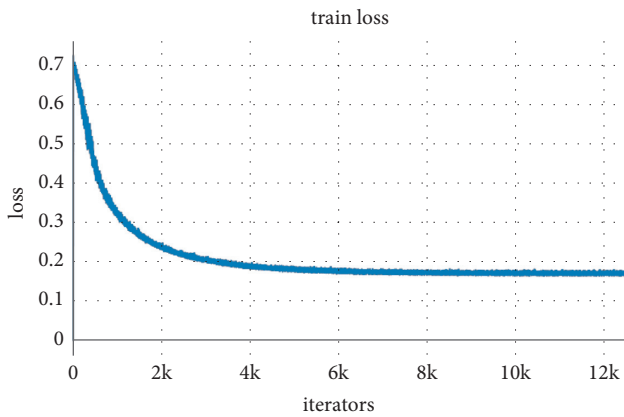


FIGURE 5: The loss changes during training.

4.2. *Experiments on OTB50 and OTB100.* The proposed method is evaluated on the OTB50 and OTB100 benchmark datasets. OTB50 has 50 video sequences, and OTB100 has 100 video sequences. The OTB dataset evaluation tool evaluates the tracking algorithm through two indicators: precision plot and success plot. The evaluation standard of tracking accuracy is the percentage of the number of frames with the center position error within T1 (the experiment is set to 20) pixels to the number of frames in the entire video sequence. The tracking success rate refers to the percentage of the frame number of the entire video sequence whose intersection

ratio IoU (Intersection over Union) is greater than the threshold T2 (experimentally set to 0.5) between the target frame predicted by the algorithm and the real target frame, as shown in the following equation:

$$\text{IoU} = \frac{\text{Box}_t \cap \text{Box}_g}{\text{Box}_t \cup \text{Box}_g}, \quad (14)$$

where Box_t represents the area of the area enclosed by the target prediction bounding box and Box_g represents the area enclosed by the target real bounding box.

As shown in Figure 6, the tracking accuracy and tracking success rate of ESA-Siam on the OTB50 dataset are 0.85 and 63.30, respectively, which are 4% and 4.77% higher than the state-of-the-art algorithm SiamRPN. Compared with the reliable channel-based method, CSR-DCF has increased by 11% and 10.04%.

It can be seen from Figure 7 that the tracking accuracy and tracking success rate of ESA-Siam on the OTB100 dataset are 0.863 and 65.04, respectively. Compared with the basic algorithm SiamFC, the tracking success rate has increased by 6.72%. It is 9% and 6.65% higher than that of the CFNet algorithm that combines correlation filtering and SiamFC, respectively, and 1.8% and 2.13% higher than that of SiamRPN. Meanwhile, the tracking success rate of ESA-Siam on the OTB100 dataset is 2.7% higher than that of SiamDW which also introduced the ResNet50 network. In addition, the performance of ESA-Siam is also better than that of CSR-DCF, a reliable channel-based method, and

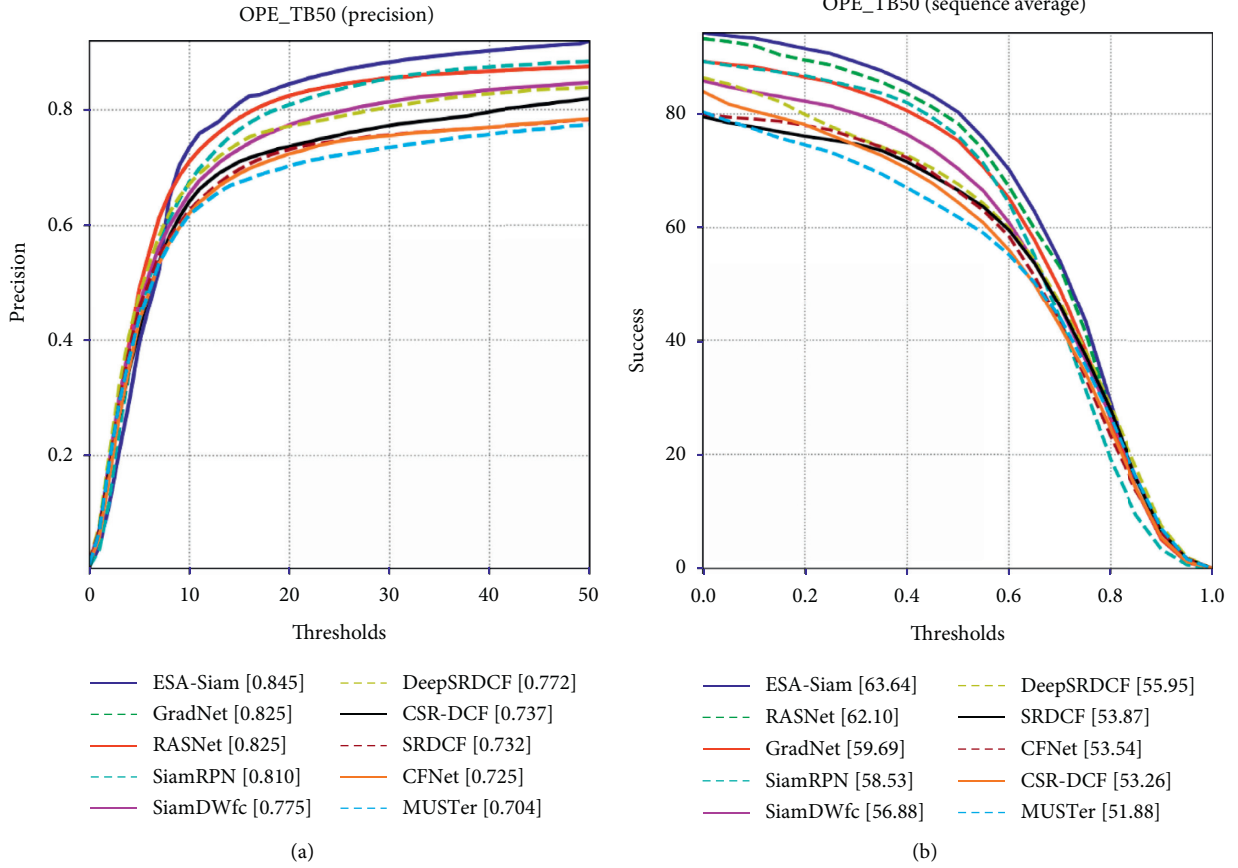


FIGURE 6: Test results of different trackers on the OTB50 dataset: (a) precision plot of OPE on OTB50; (b) success plot of OPE on OTB50.

RASNet, a method that integrates attention. ESA-Siam integrates the hybrid attention mechanism while improving the downsampling method of ResNet50 and uses the T-SCAttn module to implicitly update the template features, which can select features that are more discriminative to the target, so it can improve the robustness of the algorithm.

To verify the robustness of the ESA-Siam algorithm, we further carried out experiments on 11 tracking challenges on the OTB100 dataset, including Background Clutter (BC), Deformation (DEF), Fast Motion (FM), In-Plane Rotation (IPR), Illumination Variation (IV), Low Resolution (LR), Motion Blur (MB), Occlusion (OCC), Out-of-Plane Rotation (OPR), Out-of-View (OV), and Scale Variation (SV). As shown in Figure 8, we mainly evaluated the success of OPE on OTB100. We have observed that the ESA-Siam algorithm has won the championship in IV, IPR, LR, OCC, and so on. In other challenges such as SV, BC, and so on, ESA-Siam also has achieved great tracking performance.

Quantitative analysis of the algorithm was done as described in the previous section, in order to further verify the effectiveness of ESA-Siam. At the same time, a challenging sequence was selected from the OTB dataset for qualitative testing of the algorithm. Meanwhile, it was compared with CFNet and the related filtering algorithm DeepSRDCF combined with deep learning features, SiamFC and CSR-DCF. In the comparative experiment, six video sequences of Bird2, Human9, KiteSurf, Matrix, Singer2, and Dancer2

were selected. These six video sequences include IPR, OPR, LR, OCC, IV, DEF, FM, BC, and other challenges. Figure 9 shows the tracking effect comparison of the five algorithms including ESA-Siam. In these challenging sequences, the ESA-Siam algorithm has achieved better tracking results.

4.3. Experiments on VOT2016, VOT2018, and LaSOT. We also tested the methods on the VOT2016, VOT2018, and LaSOT datasets according to the three indicators expected average overlap (EAO), accuracy (A), and Robustness (R). Among them, EAO can be used as an index for comprehensive performance evaluation of the algorithm. The calculation of EAO is related to the accuracy and robustness. First, the average of per-frame overlaps Φ_{N_s} in the length N_s of the video sequence is defined as

$$\Phi_{N_s} = \frac{1}{N_s} \sum_{i=1}^{N_s} \Phi_i, \quad (15)$$

where Φ_i is the accuracy rate between the predicted target frame and the real target frame. EAO is defined as

$$\Phi = \frac{1}{N_{hi} - N_{lo}} \sum_{N_s=N_{lo}}^{N_{hi}} \Phi_{N_s}, \quad (16)$$

Table 2 shows the comparison of the test results of each method on VOT2016. We compared 9 algorithms including ESA-Siam, HCF, SAMF, SiamFC, SRDCF, MDNet,

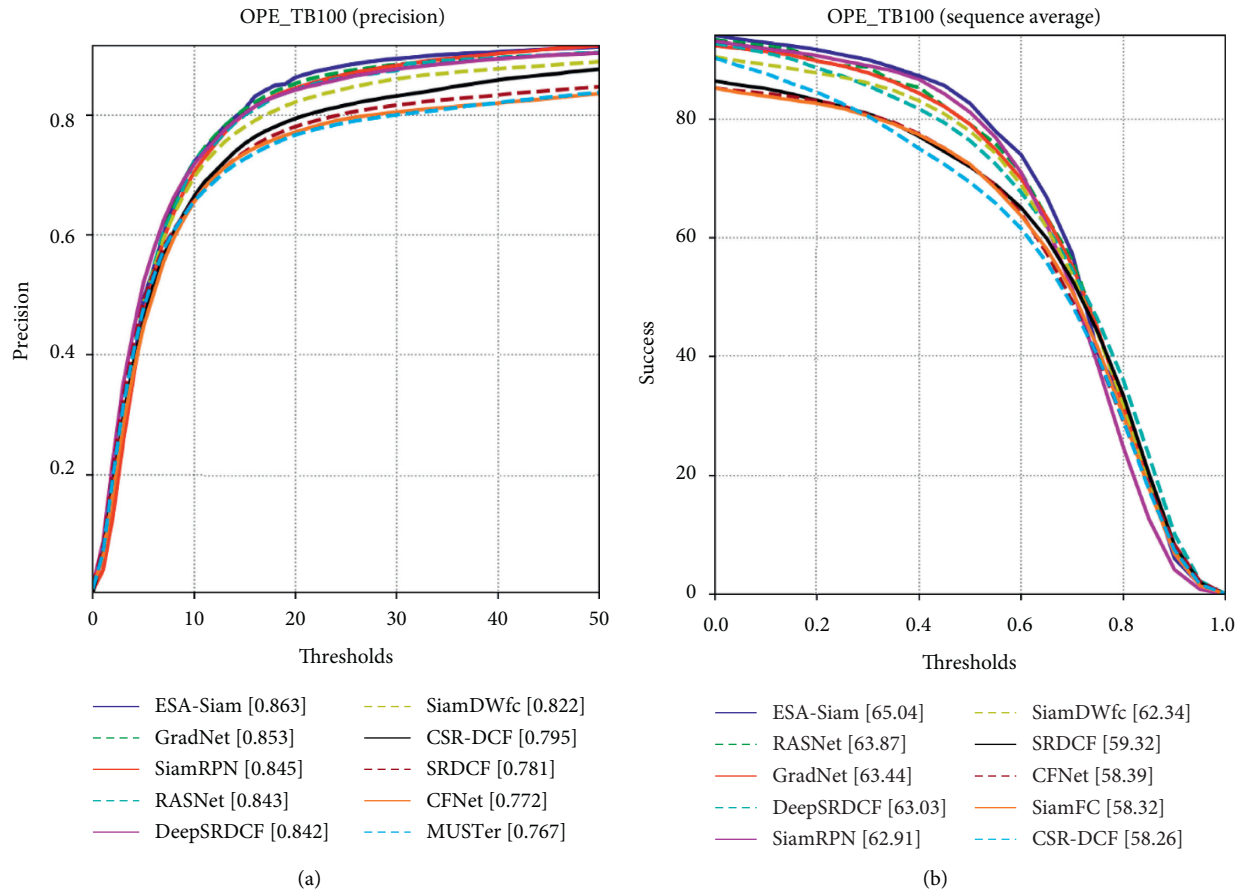


FIGURE 7: Test results of different trackers on the OTB100 dataset: (a) precision plot of OPE on OTB100; (b) success plot of OPE on OTB100.

DeepSRDCF, Staple, and C-COT. ESA-Siam has achieved good results on the indicators of accuracy and robustness.

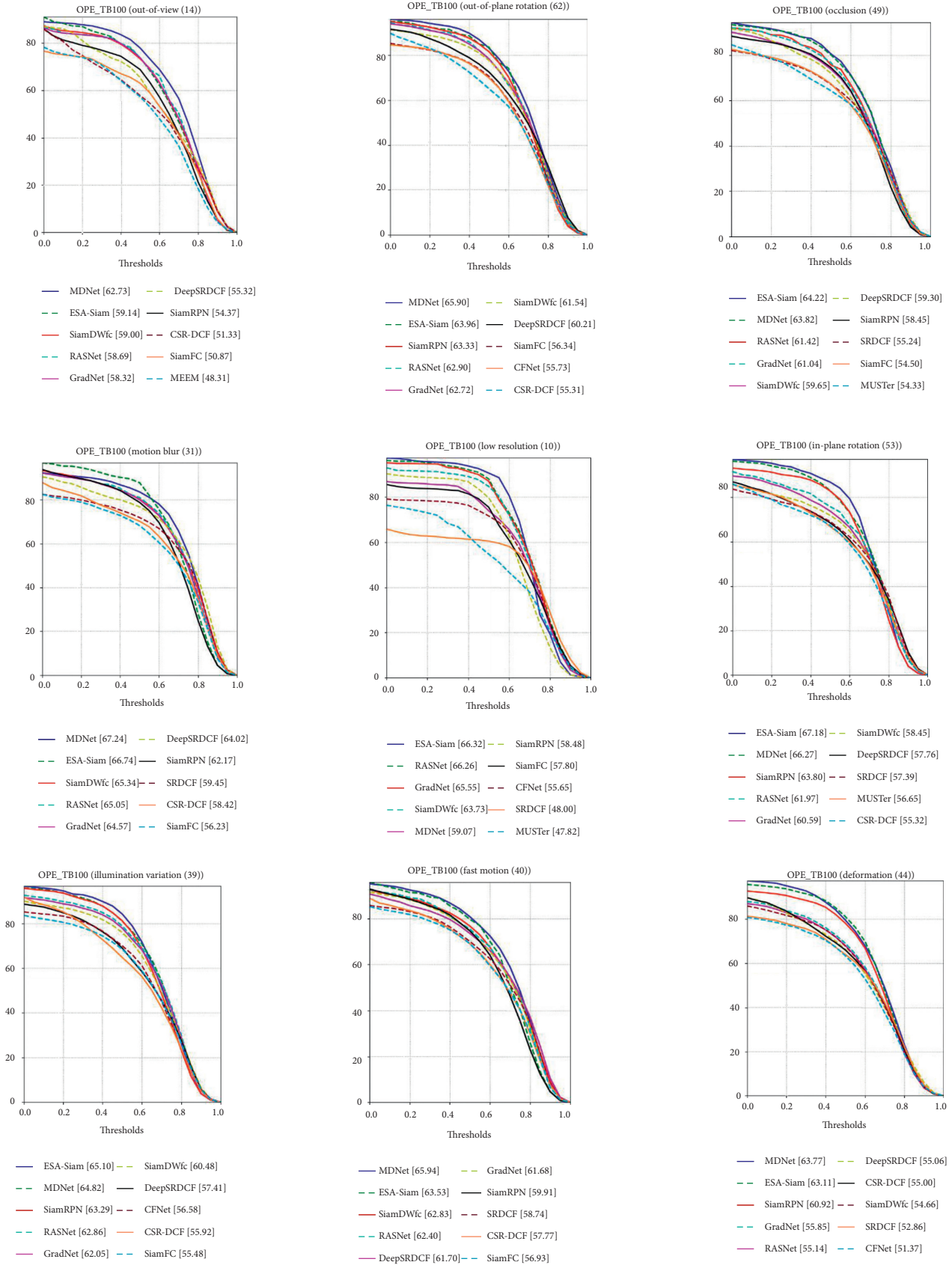
Table 3 shows the comparison of the tracking results of each method on the VOT2018 test set. Our proposed method achieves accuracy of 0.618, robustness of 0.223, and EAO of 0.411. On OVT2018, our method achieved the highest scores on the three evaluation indicators. Compared with the state-of-the-art SiamRPN, our method has a significant improvement of 7.5% and 5.6% in EAO and accuracy, respectively.

Table 4 shows the comparison of the test results of each algorithm on the LaSOT dataset. Our method achieved the highest scores in both success rate and standardization accuracy. Compared with MDNet, the standardization accuracy has increased from 0.461 to 0.515 (with 0.413 \rightarrow 0.450).

In addition, we selected three complex challenge scenarios (including Deformation, Motion Blur, and Partial Occlusion) on the LaSOT dataset to evaluate the proposed method with 9 existing state-of-the-art algorithms as shown in Figure 10. The experimental results show that the proposed method has achieved the champion results in tracking precision and success rate. In terms of success rate, the proposed method improved by nearly 6 percentage points over the second place. The proposed method can better deal with the challenging scenes in real life, such as Full Occlusion or Partial Occlusion, Target Deformation, and so on.

4.4. Ablation Study. We conducted extensive ablation studies with ESA-Siam on OTB100 to verify the effectiveness of its various components. Two indicators are used to evaluate the work of each component: one is tracking accuracy and the other is tracking success rate. We name the methods of using different components. The component using spatial self-attention is named S-Attn, the component using channel attention is named C-Attn, and the component using template search feature is named T-S-Attn. In addition, we compare the evaluation results of each component with the benchmark algorithm SiamFC and CSR-DCF, as shown in Figure 11. Experimental results verify the effectiveness of various components of ESA-Siam. Compared with SiamFC, the performance of the channel attention module C-Attn has increased by 7.6% on precision (with 58.32 \rightarrow 63.95 on success). Deleted on the other hand, the introduction of the template-search collaboration attention module T-S-Attn is 5.83% higher in success rate and 5.9% higher in accuracy than CSR-DCF, which is also based on channel weighting.

In addition, we also conducted experiments on the gold stochastic pooling method and other components in OVT2016. Table 5 shows the performance changes of the tracker with the integration of different components. As



(a)

FIGURE 8: Continued.

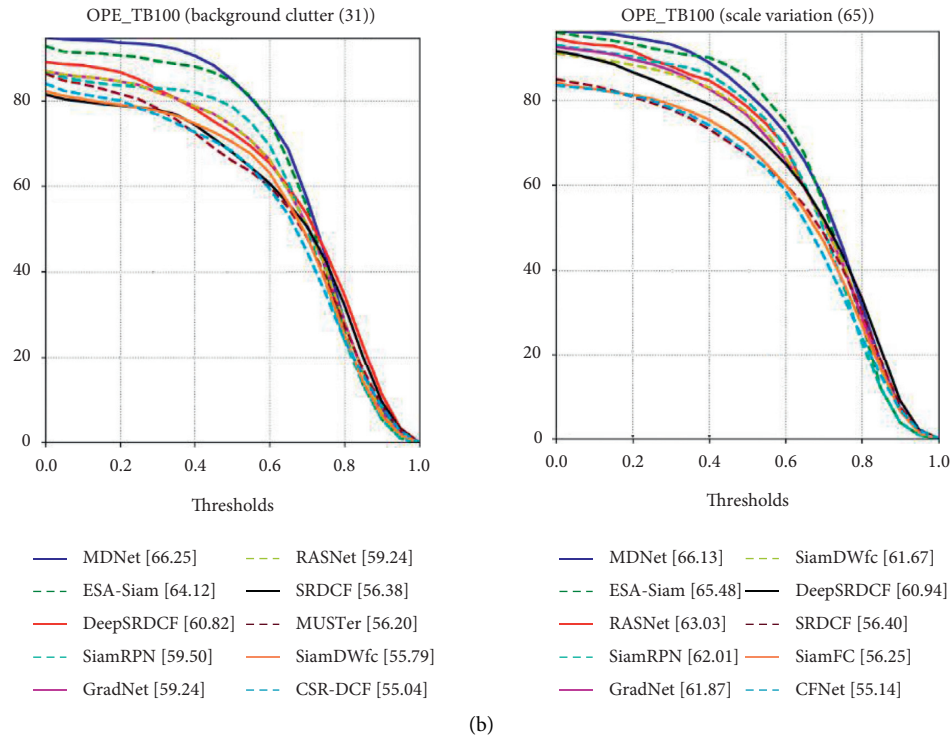


FIGURE 8: Comparison of the tracking success rate of 11 challenge sequences on the OTB100 dataset by 10 different algorithms.

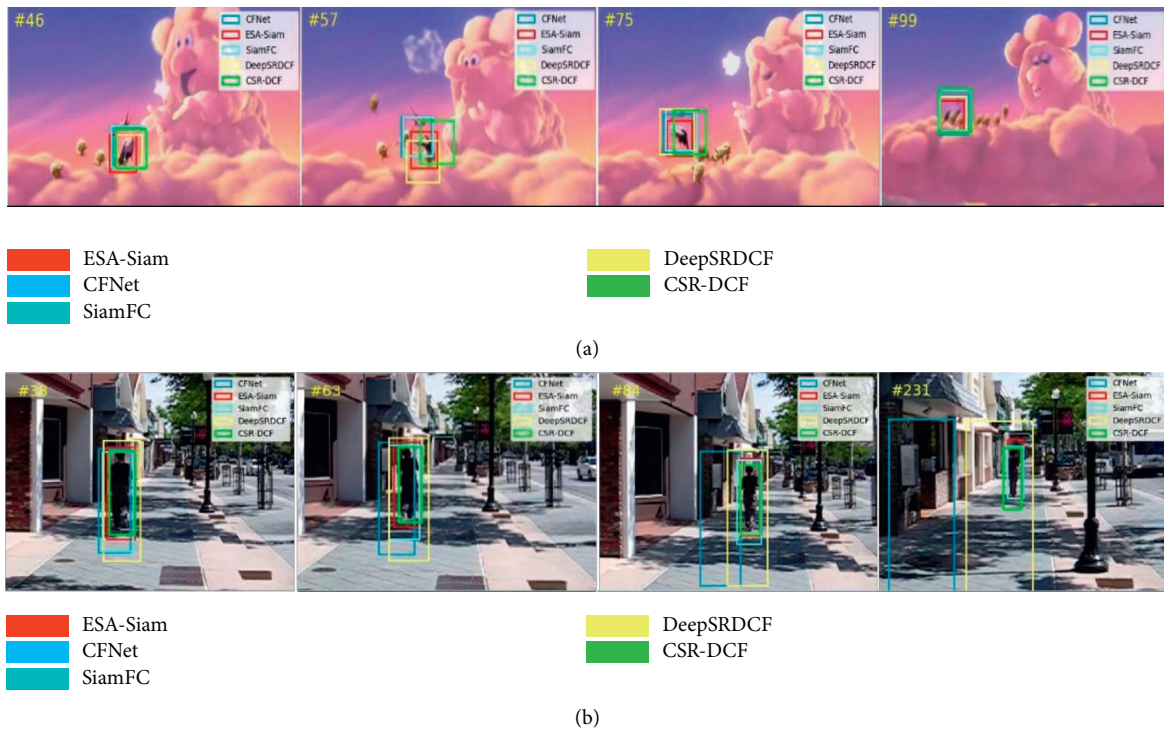


FIGURE 9: Continued.

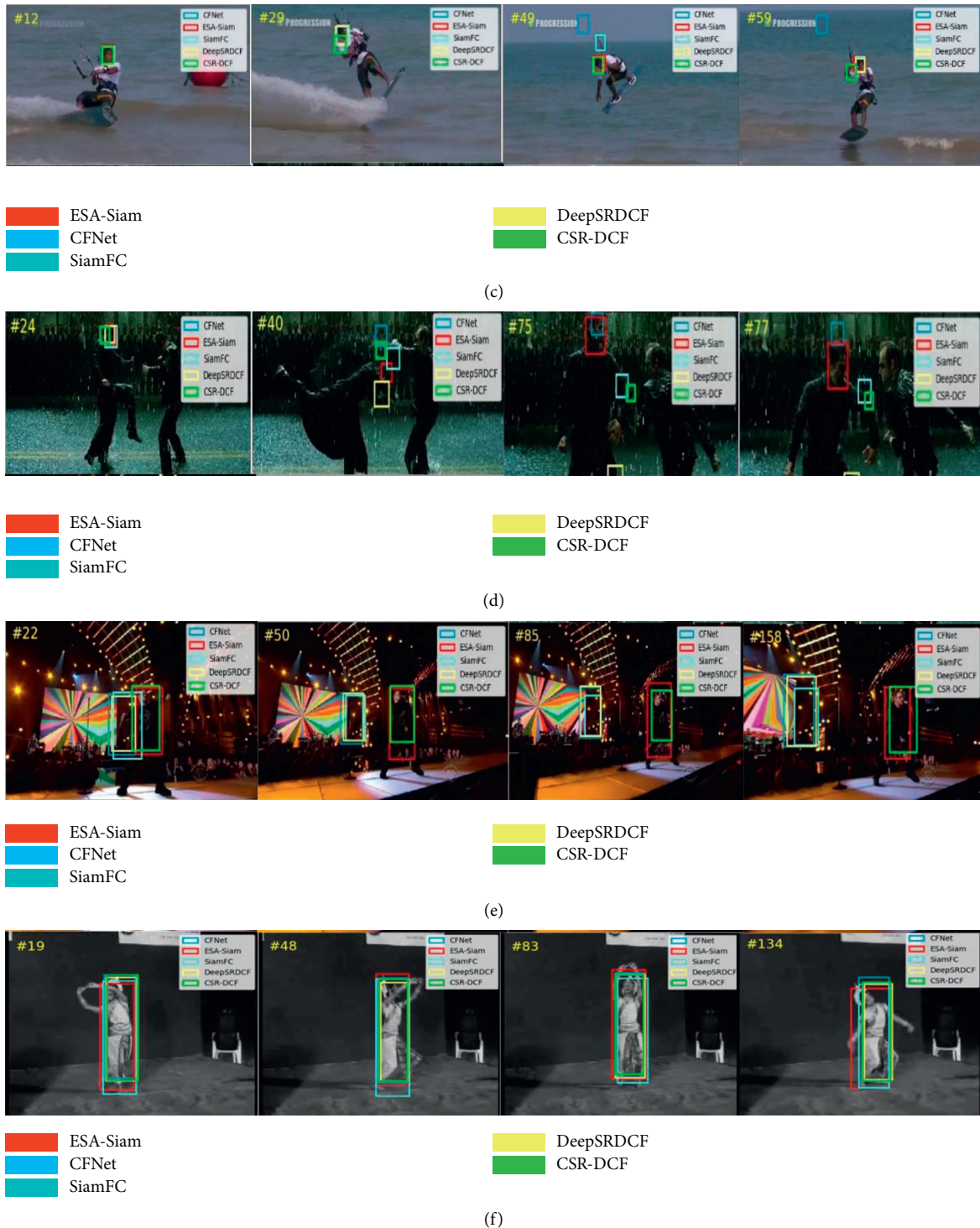


FIGURE 9: Qualitative comparison of tracking results of various algorithms on challenges, such as (a) Bird2, (b) Human9, (c) KiteSurf, (d) Matrix, (e) Singer2, and (f) Dancer2.

shown in Table 5, simply adding single channel attention and spatial attention to the benchmark SiamFC cannot effectively improve its tracking performance. Integrating channel attention and spatial attention can increase by 7.8% with EAO. If combined with the template search attention module, EAO can increase by 11.8%. In addition, gold

stochastic pooling components can also be effectively applied to the twin network to improve the tracking performance. As shown in Table 6, we conducted a series of experiments to discuss the impact of different gold stochastic pooling thresholds on the tracking performance. When $T=0$, it means that gold stochastic pooling degenerates to

Input: random initialization the network parameters θ , golden threshold stochastic pooling T , spatial self-attention parameters of α .
 Template Z and search patch X from GOT10K.
Preprocessing: crop and resize Z and X and set optimizer, loss function, and learning rate adjustment strategy.
While $epoch > 0$ and input video dataset is not empty **do**
 Get template Z and corresponding bounding box;
 Get search patch X and corresponding bounding box;
 Compute $\psi(Z)$, $\psi(X)$ by the backbone network;
 Compute $\mu(\psi(Z))$, $\mu(\psi(X))$ by the channel attention module;
 Compute $\eta(\mu(\psi(Z)))$, $\eta(\mu(\psi(X)))$ by the spatial self-attention module;
 Create sample positive and negative labels;
 Compute $\Delta(Z, X)$ and update template;
 Computer response map of Z nad X ;
 Computer loss and update parameters;
 Optimize loss to minimize.
end

ALGORITHM 1: Offline training of the proposed framework.

Input: test video; initial frame and bounding box of initial frame;
 Compute $\psi(Z)$ by the backbone network;
 Compute $\mu(\psi(Z))$ by the channel attention module;
 Compute $\eta(\mu(\psi(Z)))$ by the spatial self-attention module;
Preprocessing: crop and resize X and set three different scale patches X_1, X_2, X_3 .
While test video is not empty **do**
 Get search patch X and corresponding bounding box;
 Compute $\psi(X)$ by the backbone network;
 Compute $\mu(\psi(X))$ by the channel attention module;
 Compute $\eta(\mu(\psi(X)))$ by the spatial self-attention module;
 Upsampling feature map X to 272×272 ;
 Locate target center in feature map X by finding peak;
 Computer the offset of the upsampled map relative to the feature map;
 Computer the offset of the feature map relative to original image;
 Update target size and corresponding bounding box;
end

ALGORITHM 2: Inference of the proposed framework.

TABLE 2: Results on VOT2016 on expected average overlap (EAO), accuracy (A), and robustness (R).

Trackers	Accuracy \uparrow	Robustness \downarrow	EAO \uparrow
HCF	0.450	0.396	0.220
SAMF	0.503	0.443	0.226
SiamFC	0.532	0.461	0.235
SRDCF	0.535	0.419	0.247
MDNet	0.541	0.337	0.257
DeepSRDCF	0.528	0.326	0.276
Staple	0.544	0.378	0.295
C-COT	0.539	0.238	0.331
ESA-Siam	0.622	0.231	0.353

The values in bold highlights the algorithm with the first performance ranking, which can be seen intuitively.

normal stochastic pooling. When the value of t is greater, the probability of selecting the largest activation is higher. We observe that ESA-Siam can achieve the best performance

when $T = 0.618P_{\max}$. Due to the mathematical peculiarity of 0.618 in stochastic pooling, we named it the golden stochastic pooling.

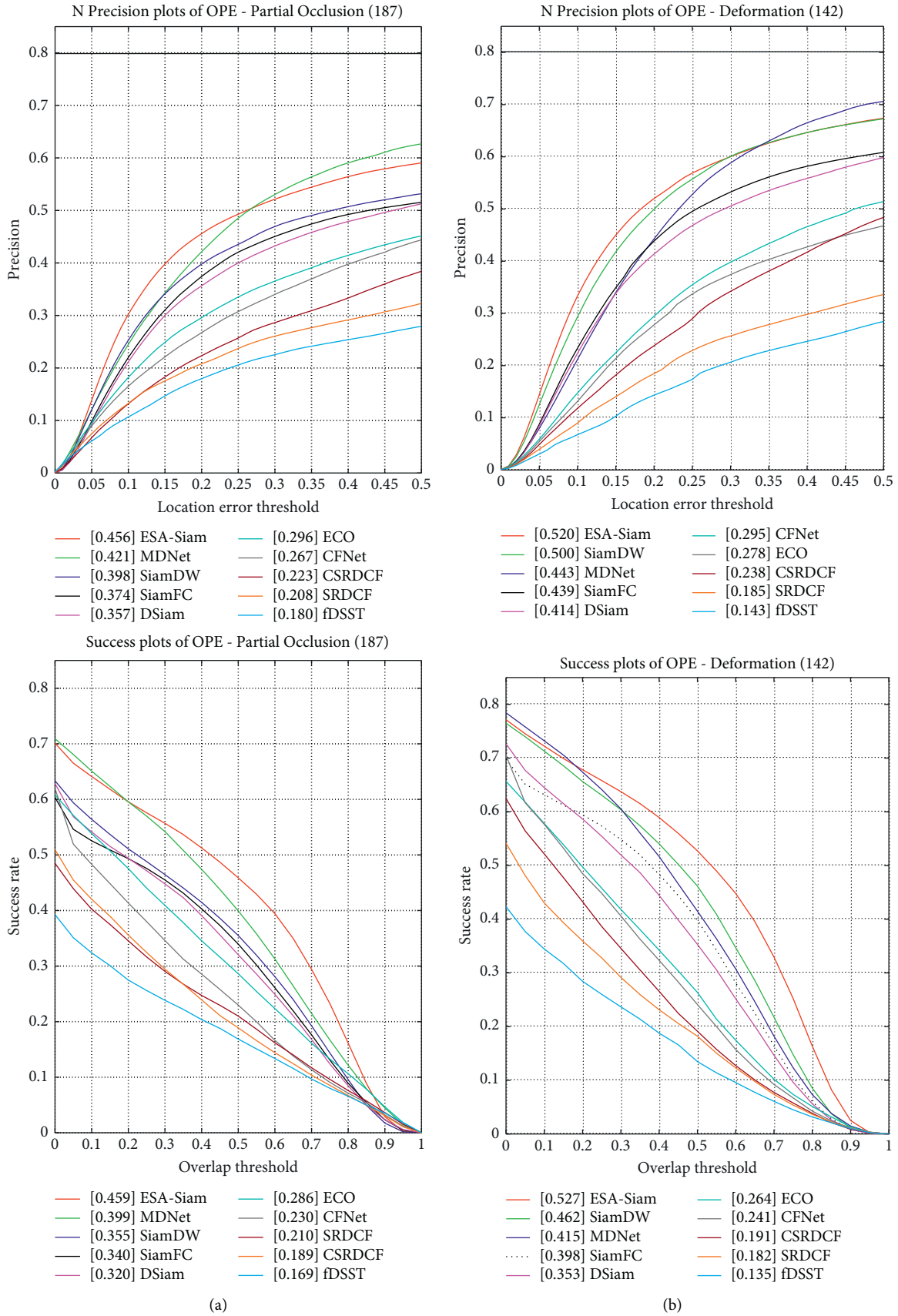
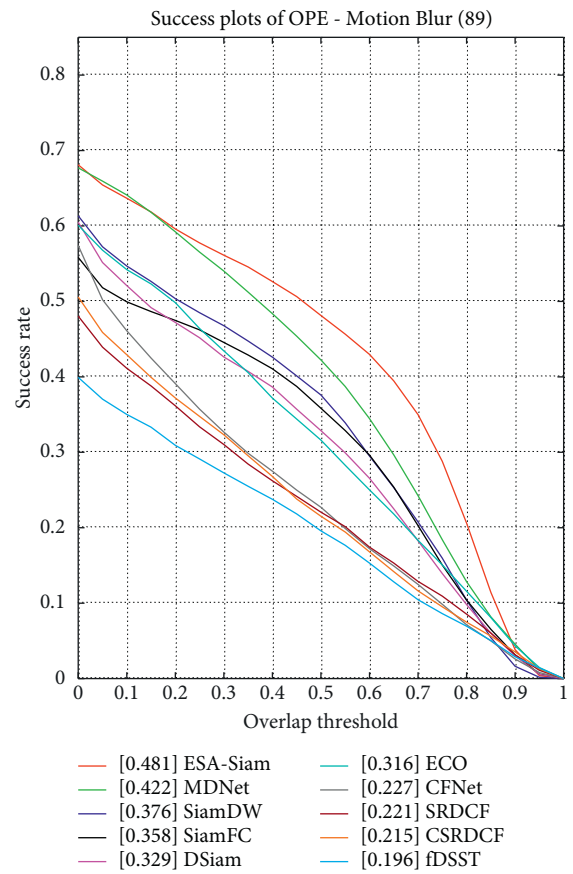
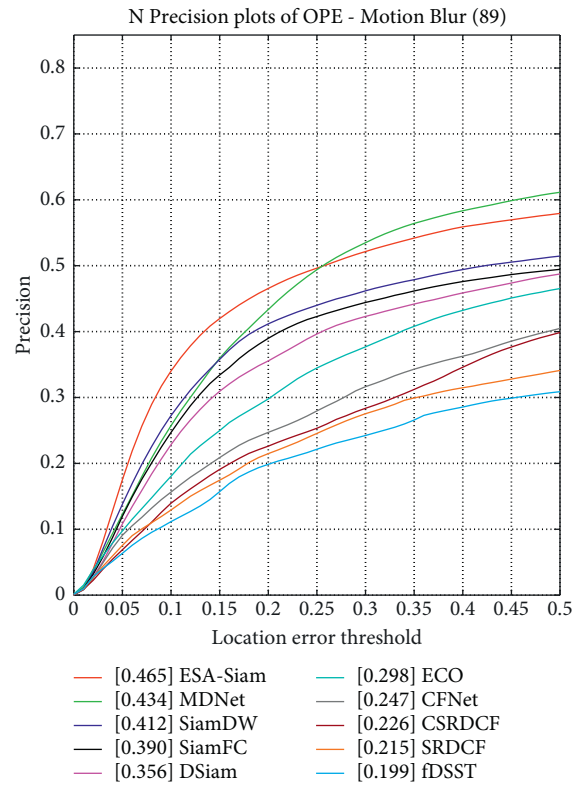


FIGURE 10: Continued.



(c)

FIGURE 10: The evaluation results of the proposed method on complex challenge scenarios on the LaSOT dataset. (a) Partial occlusion. (b) Deformation. (c) Motion Blur.

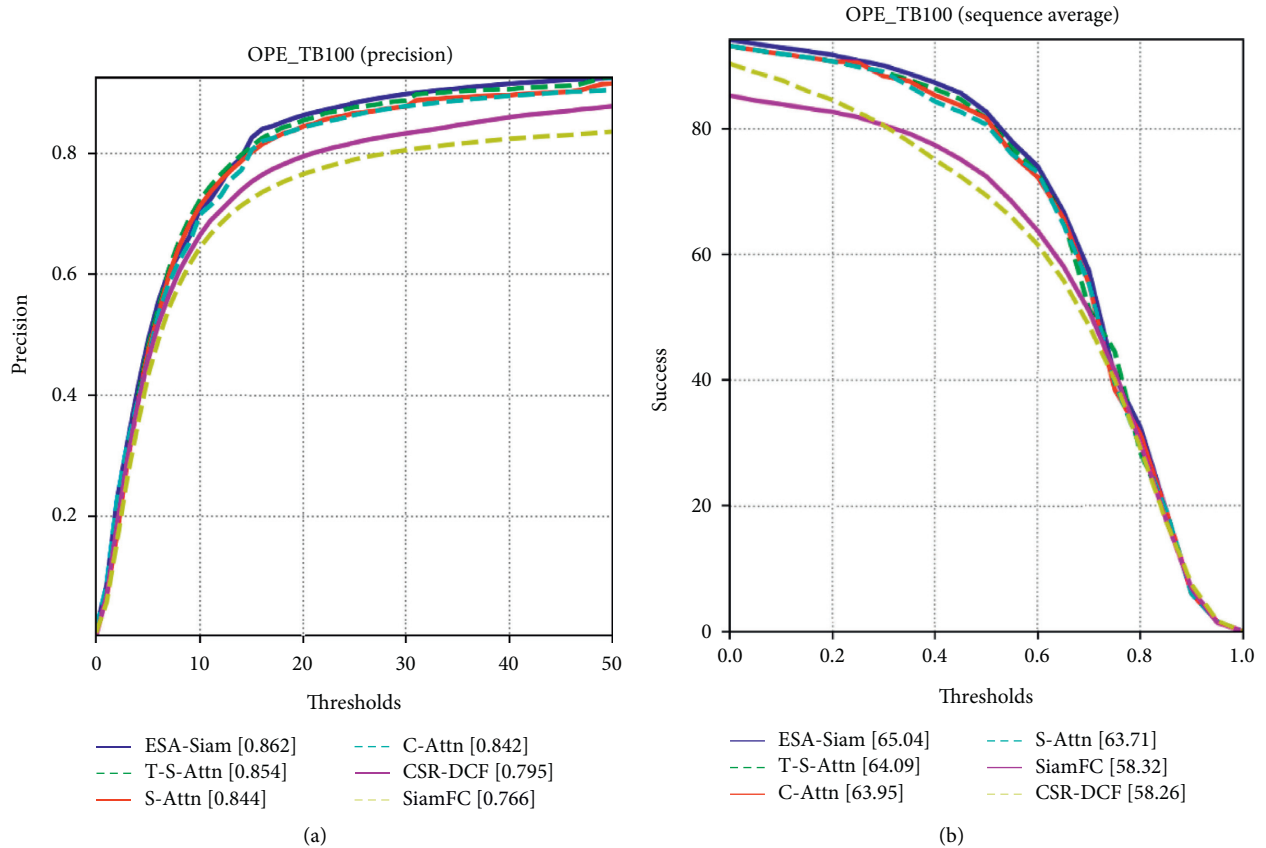


FIGURE 11: Precision and success plots of OPE on OTB100: (a) precision plots of OPE on OTB100; (b) success plots of OPE on OTB100. The performance of each component of ESA-Siam (C-Attn, S-Attn, and T-S-Attn) is better than that of the benchmark algorithms SiamFC and CSR-DCF.

TABLE 3: Results on VOT2018 on expected average overlap (EAO), accuracy (A), and robustness (R).

Trackers	Accuracy \uparrow	Robustness \downarrow	EAO \uparrow	Speed
Staple	0.530	0.688	0.169	13.4
SiamFC	0.503	0.585	0.188	84
DSiam	0.512	0.646	0.196	46.6
UpdateNet	0.518	0.454	0.244	10.5
CSR-DCF	0.491	0.356	0.256	12.7
C-COT	0.494	0.318	0.267	14.1
ECO	0.484	0.280	0.280	77.6
DeepSRDCF	0.489	0.293	0.293	20.5
SiamRPN	0.562	0.276	0.336	76.8
ESA-Siam	0.618	0.223	0.411	60

The values in bold highlights the algorithm with the first performance ranking, which can be seen intuitively.

TABLE 4: Results on LaSOT with success and normalized precision (Norm.Pr).

Trackers	Success	Norm.Pr
SiamFC	0.382	0.420
DSiam	0.362	0.405
CFNet	0.258	0.312
CSR-DCF	0.224	0.254
SiamDW	0.397	0.435
ECO	0.329	0.338
SRDCF	0.245	0.248
fDSST	0.196	0.208
MDNet	0.435	0.461
ESA-Siam	0.501	0.495

The values in bold highlights the algorithm with the first performance ranking, which can be seen intuitively.

TABLE 5: Ablation study on VOT2016 (base: SiamFC; GSP: gold stochastic pooling; CA: channel attention; SA: spatial self-attention; T-SA: template search collaboration attention).

Method	$A \uparrow$	$R \downarrow$	EAO \uparrow	Δ EAO (%)
Base	0.532	0.461	0.235	—
Base + GSP	0.547	0.433	0.241	+0.6
Base + GSP + CA	0.557	0.367	0.250	+1.5
Base + GSP + SA	0.562	0.362	0.253	+1.8
Base + GSP + T-SA	0.577	0.291	0.270	+3.5
Base + GSP + CA + SA	0.606	0.273	0.313	+7.8
Base + GSP + CA + SA + T-SA	0.622	0.231	0.353	+11.8

TABLE 6: Performance of the proposed ESA-Siam on OTB100 dataset using different thresholds for stochastic pooling.

Threshold	Precision	Success
ESA-Siam + $T = 0.367$	0.772	58.38
ESA-Siam + $T = 0.433$	0.797	59.76
ESA-Siam + $T = 0.525$	0.827	62.45
ESA-Siam + $T = 0.618$	0.862	65.04
ESA-Siam + $T = 0.780$	0.834	63.22
ESA-Siam + $T = 0.822$	0.830	62.31

The values in bold highlights the algorithm with the first performance ranking, which can be seen intuitively.

5. Conclusion

We propose an enhanced visual attention Siamese network that can update template features online for visual tracking. We introduce a template search collaboration attention module that can implicitly update target features online and combine the channel attention and spatial self-attention modules in the computationally efficient ECA module. Based on the Siamese network, combining with the visual attention mechanism can ensure that the algorithm is simple and efficient. ESA-Siam can keep the tracking speed in real-time and make the algorithm more robust. The algorithm we proposed can be applied to scenes disturbed by background, such as video surveillance, vehicle tracking, and UAV tracking.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

ZWQ and ZG were responsible for conceptualization, original draft preparation, and review and editing. ZWQ, ZG, ZZG, and LQ validated the study. ZWQ and ZZG were responsible for formal analysis and funding acquisition. ZZG visualized the study. ZWQ was responsible for

methodology, project administration, and resources. LQ curated the data and supervised the study. ZG was responsible for investigation. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This study was partially supported by the National Key Research and Development Program of China (grant nos. 2018AAA0100400 and 2019QY1604), National Natural Science Foundation of China (grant no. U1836217), Open Platform Innovation Foundation of Hunan Provincial Education Department (grant no. 20K046), Scientific Research Project of Hunan Provincial Department of Education (grant no. 17C0479), and Special Fund Support Project for the Construction of Innovative Provinces in Hunan (2019GK4009).

References

- [1] M. Elhoseny, "Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 611–630, 2020.
- [2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, pp. 2722–2730, Santiago, Chile, December 2015.
- [3] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "Autotrack: towards high-performance visual tracking for uav with automatic spatio-temporal regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11923–11932, Seattle, WA, USA, June 2020.
- [4] M. Felsberg, A. Berg, G. Hager et al., "The thermal infrared visual object tracking VOT-TIR2015 challenge results," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 76–88, Santiago, Chile, December 2015.

- [5] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5296–5305, Honolulu, HI, USA, July 2017.
- [6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.
- [7] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 4310–4318, Santiago, Chile, December 2015.
- [8] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1396–1404, Honolulu, HI, USA, July 2017.
- [9] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, Nottingham, England, September, 2014.
- [10] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integratio," in *Proceedings of the European conference on computer vision*, pp. 254–265, Zurich, Switzerland, September 2014.
- [11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the European conference on computer vision*, pp. 850–865, Amsterdam, Netherlands, October 2016.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [13] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8971–8980, Salt Lake City, UT, USA, June 2018.
- [14] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. Hilaire Torr, "Fast online object tracking and segmentation: a unifying approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1328–1338, Long Beach, CA, USA, June 2019.
- [15] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xiang, and J. Yan, "Siamrpn++: evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, Long Beach, CA, USA, June 2019.
- [16] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 101–117, Munich, Germany, September 2018.
- [17] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, and H. Wang, "Learning deep multi-level similarity for thermal infrared object tracking," *IEEE Transactions on Multimedia*, vol. 23, pp. 2114–2126, 2020.
- [18] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2020.
- [19] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based Systems*, vol. 194, Article ID 105526, 2020.
- [20] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," *ICML deep learning workshop*, vol. 2, 2015.
- [21] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Proceedings of the European Conference on Computer Vision*, pp. 791–808, Amsterdam, Netherlands, October 2016.
- [22] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
- [23] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 1763–1771, Venice, Italy, October 2017.
- [24] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, Long Beach, CA, USA, 2017.
- [25] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module, Computer Vision_ECCV 2018," in *Lecture Notes in Computer Science*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11211, Springer, Cham, Switzerland, 2018.
- [26] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, <https://arxiv.org/abs/1803.02155>.
- [27] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: residual attentional siamese network for high performance online visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4854–4863, Salt Lake City, UT, USA, June 2018.
- [28] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 548–557, Salt Lake City, UT, USA, June 2018.
- [29] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, Portland, OR, USA, June 2013.
- [30] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [31] M. P. Kristan, A. Lebeda, J. Mates et al., "The visual object tracking VOT2016 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshop*, pp. 777–823, Amsterdam, Netherland, October 2016.
- [32] M. Kristan, A. Leonardis, J. Matas et al., "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, p. 0, Munich, Germany, September 2018.
- [33] H. Fan, L. Lin, F. Yang et al., "Lasot: a high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, Long Beach, CA, USA, June 2019.
- [34] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, June 2010.

- [35] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Proceedings of the European conference on computer vision*, pp. 472–488, Amsterdam, Netherlands, October 2016.
- [36] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2805–2813, Honolulu, HI, USA, July 2017.
- [37] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4293–4302, Las Vegas, NV, USA, June 2016.
- [38] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 58–66, Santiago, Chile, December 2015.
- [39] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6638–6646, Honolulu, HI, USA, July 2017.
- [40] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843, Salt Lake City, UT, USA, June 2018.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [42] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.
- [43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, UT, USA, June 2018.
- [44] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," 2013, <https://arxiv.org/abs/1301.3557>.
- [45] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European conference on computer vision*, pp. 483–499, Amsterdam, Netherlands, October 2016.
- [46] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the 2020 IEEE. CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Seattle, WA, USA, June 2020.
- [47] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4591–4600, Long Beach, CA, USA, June 2019.
- [48] C. Ma, J.-B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 3074–3082, Santiago, Chile, December 2015.
- [49] A. Lukezic, T. Vojir, L. Cehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6309–6318, Honolulu, HI, USA, July 2017.
- [50] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6162–6171, Seoul, South Korea, April 2019.
- [51] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.
- [52] M. Danelljan, G. Häger, F. S. Khan, and S. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2016.
- [53] L. Zhang, A. Gonzalez-Garcia, J. V. D. Weijer, M. Danelljan, and F. Shahbaz Khan, "Learning the model update for siamese trackers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4010–4019, Seoul, South Korea, April 2019.
- [54] L. Huang, X. Zhao, and K. Huang, "Got-10k: a large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2021.

Research Article

Reinforcement Learning for Security-Aware Workflow Application Scheduling in Mobile Edge Computing

Binbin Huang ¹, Yuanyuan Xiang,¹ Dongjin Yu ¹, Jiaojiao Wang,² Zhongjin Li,¹ and Shangguang Wang³

¹School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China

²Institute of Intelligent Media Technology, Communication University of Zhejiang, Hangzhou 310018, China

³State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Binbin Huang; huangbinbin@hdu.edu.cn and Dongjin Yu; yudj@hdu.edu.cn

Received 2 March 2021; Accepted 15 May 2021; Published 25 May 2021

Academic Editor: Xiaolong Xu

Copyright © 2021 Binbin Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile edge computing as a novel computing paradigm brings remote cloud resource to the edge servers nearby mobile users. Within one-hop communication range of mobile users, a number of edge servers equipped with enormous computation and storage resources are deployed. Mobile users can offload their partial or all computation tasks of a workflow application to the edge servers, thereby significantly reducing the completion time of the workflow application. However, due to the open nature of mobile edge computing environment, these tasks, offloaded to the edge servers, are susceptible to be intentionally overheard or tampered by malicious attackers. In addition, the edge computing environment is dynamical and time-variant, which results in the fact that the existing quasistatic workflow application scheduling scheme cannot be applied to the workflow scheduling problem in dynamical mobile edge computing with malicious attacks. To address these two problems, this paper formulates the workflow scheduling problem with risk probability constraint in the dynamic edge computing environment with malicious attacks to be a Markov Decision Process (MDP). To solve this problem, this paper designs a reinforcement learning-based security-aware workflow scheduling (SAWS) scheme. To demonstrate the effectiveness of our proposed SAWS scheme, this paper compares SAWS with MSAWS, AWM, Greedy, and HEFT baseline algorithms in terms of different performance parameters including risk probability, security service, and risk coefficient. The extensive experiments results show that, compared with the four baseline algorithms in workflows of different scales, the SAWS strategy can achieve better execution efficiency while satisfying the risk probability constraints.

1. Introduction

In recent years, with the explosive growth of smart devices (such as smart cameras, smart glasses, smart bracelets, and smart phones), a large number of advanced mobile applications (such as real-time navigation systems, interactive online games, virtual reality, and augmented reality) are emerging rapidly. In order to efficiently process these mobile applications, mobile devices need to be equipped with abundant computing resources and battery capabilities [1, 2]. However, due to the limited size of mobile devices, they are usually resource-constrained. Therefore, the conflict between the ever-increasing resource requirements of

mobile applications and the limited resource capabilities of mobile devices brings great challenges to execute these mobile applications.

Mobile Edge Computing (MEC) as a new computing paradigm brings remote cloud resource to the edge servers nearby mobile users, enabling mobile users to offload partial or all computation tasks of mobile applications to edge servers for collaborative execution, and thereby greatly alleviating the conflict between resource supply and demand, effectively reducing the application completion time and the mobile devices' energy consumption [3–5].

Many mobile applications are typical workflow models, and they consist of a sequence of precedence-constrained

tasks. For example, a video streaming-based face recognition application mainly consists of motion detection and face recognition. The face recognition further consists of face detection, image preprocessing, feature extraction, and classification [3, 6]. In mobile edge computing, workflow application scheduling has a higher complexity in comparison to independent task scheduling [7–9]. In addition, it also faces two challenges for workflow application scheduling in mobile edge computing as follows. One is the edge environment dynamics, such as the time-varying channel quality and workload of edge servers, which can impact the workflow application scheduling decision. The other is the security problem of workflow application scheduling. Due to the open nature of the edge environment, the edge servers that aggregate an amount of user data frequently suffer from malicious attacks such as data leakage and tampering, which pose a serious threat to successfully execute these offloaded tasks [10–13]. Hence, it needs to employ various types of security services to effectively defend against the hostile attacks and protect these offloaded tasks. However, employing security services inevitably incurs additional security overhead, which will increase the completion time of workflow application. Therefore, it is a big challenge to design an efficient security-aware workflow scheduling scheme to reduce the completion time of workflow application while satisfying its security requirement.

To meet the aforementioned challenges, this paper formulates the security-aware workflow scheduling problem in MEC to be a Markov Decision Process (MDP) [14]. The environment state, which consists of the task list on each edge server, the workloads on each edge server, and the channel states between the mobile device and the edge servers, can be observed. Based on the environment state, the task nodes of the workflow are dynamically scheduled to edge servers. The deep reinforcement learning algorithm is suitable to solve decision-making problems with unknown prior knowledge [15–19]. To solve this problem, this paper proposes a deep reinforcement learning-based security-aware workflow scheduling scheme (SAWS). Its main objective is to optimize the completion time of workflow while satisfying its security requirement. To evaluate the effectiveness of the SAWS scheme, this paper implements average workload minimization (AWM), maximum SAWS (MSAWS), Greedy, and HEFT baseline algorithms. We compare the SAWS scheme with these four baseline algorithms under different risk probabilities, different security services, different risk coefficients, different edge server's computing capacities, and different number of edge servers. The experimental results demonstrate that the SAWS strategy can optimize the completion time of workflow application while satisfying the risk probability constraint. The main contributions of this paper can be summarized as follows:

This paper focuses on the security problem of workflow scheduling in a dynamic edge computing, which is more complex than independent task scheduling.

This paper formulates the security-aware workflow scheduling problem in mobile edge computing to be a

finite Markov decision process, and its main objective is to minimize the completion time of workflow while satisfying the risk probability constraint.

This paper proposes a deep Q-network-based security-aware workflow scheduling (SAWS) scheme to solve the workflow scheduling problem in a dynamic edge computing environment with malicious attacks. Extensive experimental results demonstrate that the SAWS scheme can greatly reduce the completion time of workflow application while satisfying the risk probability constraint.

The rest of this paper is organized as follows. In Section 2, the related work is summarized. In Section 3, the system model and problem formulation for security-aware workflow scheduling in MEC are presented. In Section 4, the deep reinforcement learning-based security-aware workflow scheduling scheme is described in detail. In Section 5, the simulation parameters are settled, and the experimental performance is analyzed. In Section 6, the work of this paper is concluded.

2. Related Work

The task offloading problem in the MEC has been studied in a lot of works. According to different optimization goals, these works can be classified into three categories. The first one is task offloading with the goal of optimizing the mobile device's energy consumption. For example, Huang et al. [7] propose a security and cost-aware task offloading scheme based on deep reinforcement learning for task offloading in single-user multiserver scenarios. Its main goal is to minimize the task processing delay and mobile device energy consumption while satisfying the security requirement for task. Chen et al. [20] formulate task offloading problem in single-user single-server scenario to be a stochastic optimization problem and decompose this problem into two deterministic optimization subproblems. To solve these two subproblems, a TOFEE algorithm is proposed to optimize the mobile device's energy consumption. Wu et al. [21] propose a Lyapunov optimization-based energy-efficient task offloading scheme to determine the operating position of the application, the objective of which is to minimize the average energy consumption of mobile devices while satisfying the average response time constraint. The second one is task offloading with the goal of optimizing the task processing delay. For example, Chalapathi et al. [22] propose a task scheduling scheme to solve the task offloading problem in multiple cloudlets, aiming at minimizing the task processing delay. Xu et al. [23] design an adaptive task offloading scheme, which leverages decomposition-based multiobjective evolutionary algorithms to generate feasible solutions, to optimize the task processing latency and resource utilization of edge system. The third one is task offloading with the goal of optimizing the weighted sum of the mobile device's energy consumption and the task processing delay. Wu et al. [24] propose a Lyapunov optimization-based energy-efficient task offloading scheme to control the computational and communication overheads

and further choose optimal computational location for the application to minimize energy consumption and task processing time. However, all above works mainly focus on the independent task scheduling in MEC. The task nodes of workflow are precedence-constrained. The above schemes are not suitable for workflow scheduling.

To further study the workflow scheduling problem in MEC, Xu et al. [25] construct a multiresource energy consumption model to solve the unity problem for traditional energy consumption model and propose a particle swarm algorithm-based energy-efficient multiresource workflow scheduling algorithm. Its main objective is to reduce the energy consumption of mobile devices while satisfying the completion time constraint for workflow. Wu et al. [26] construct a weighted resource sum graph based on resource consumption and further design a novel cost-efficient partitioning scheme, the objective of which is to find the optimal partitioning scheme to reduce execution time and energy consumption. Zhu et al. [27] formulate the workflow scheduling problem in MEC to be a joint optimization problem of energy consumption and time delay and adopt the deep Q network algorithm to solve the optimal scheduling scheme. However, the execution order of the workflow is assumed in advance, and how to calculate the execution order of workflow with precedence constraints is not introduced. In addition, this paper does not pay attention to the security problem of workflow scheduling in MEC. Liu [28] proposes a novel maximum probability function and deep Q network-based multiworkflow scheduling scheme to solve the scheduling problem in multiuser edge computing environment, which can find a high-quality workflow scheme in a dynamic environment. However, this paper does not pay attention to the security problem of workflow scheduling in dynamic MEC. Therefore, all the above scheduling schemes are not suitable for security-aware workflow scheduling in dynamic mobile edge computing.

With the escalation of data security threats in mobile edge computing [10–12, 29, 30], a lot of related works have taken some measures to protect security-critical applications and the large amount of data generated in mobile devices from malicious attacks. Huang [6] designs a workflow scheduling scheme based on Genetic Algorithms to minimize the mobile device's energy consumption under the completion time of workflow and risk probability constraints. Elgendy et al. [11] design a multidevice and single-server cooperative task offloading scheme to solve the security-aware multiuser resource allocation and task offloading problem. The goal is to minimize the time delay and energy consumption of the whole system. Jia et al. [31] design an identity-based anonymous authentication key agreement protocol to ensure the security of sensitive data in MEC. He et al. [32] design a security mechanism based on adaptive algorithms to solve the security problem of IoT applications in mobile edge computing. Chen et al. [33] propose a malicious application detection method based on deep learning on mobile devices, which greatly improves the security of mobile edge computing. Xu et al. [34] design a secure service

offload approach to promote Internet of vehicles service utility and edge utility while ensuring privacy security in software-defined networks enabled edge computing. Xu et al. [35] adopt a location-sensitive-hash (LSH) method to encrypt the feature information for the offloaded services and further design s LSH-based offloading scheme, the goal of which is to minimize the energy consumption and response time of all services while guaranteeing the service security. All above researches mainly design security strategies from different points to ensure the security of edge computing, and they do not pay attention to the security problem of workflow scheduling in a dynamic edge computing with unknown prior knowledge. Aiming at this problem, this paper mainly focuses on security-aware workflow scheduling problem in dynamic mobile edge computing environment with security threats.

3. System Model and Problem Formulation

In this section, we first introduce the mobile edge computing model, security cost model, communication model, and risk probability model in mobile edge computing environment, respectively, and then describe the security-aware workflow scheduling problem in detail.

3.1. Mobile Edge Computing Model. As illustrated in Figure 1, we consider a mobile edge computing system, which consists of a mobile device U and n edge servers $eNB = \{eNB_1, \dots, eNB_i, \dots, eNB_n\}$. The mobile device U can be denoted by a two-tuple $U = \{f_u, N_u\}$, where f_u denotes the CPU frequency of the mobile device, and N_u denotes the number of CPU cores of the mobile device. Due to the limited computing resources and battery capacity of mobile device, the workflow applications (such as a video streaming-based face recognition application) running on mobile device can be scheduled to edge servers through wireless network. Each edge server can be denoted by a two-tuple $eNB_i = \langle f_{c,i}, N_{c,i} \rangle$, where $f_{c,i}$ denotes the CPU frequency of the i th edge server, and $N_{c,i}$ denotes the number of CPU cores of the i th edge server. Each edge server has an execution queue $Q_{c,i}$ that is used to store the tasks scheduled to the i th edge server.

Each mobile application can be abstracted into a workflow model, which can be denoted by a directed acyclic graph (DAG) $G = \langle V, E \rangle$, in which $V = \{v_1, \dots, v_k, \dots, v_K\}$ denotes a set of task nodes, and $E = \{e_{kl} | v_k \in V, v_l \in V\}$ denotes a set of edges between task nodes. Each task node v_k can be characterized by a three-tuple $v_k = \langle W_k, D_k^{tx}, D_k^{rx} \rangle$, in which W_k denotes the workload (CPU Cycles) of task node v_k , D_k^{tx} denotes the input data size (MB) of task node v_k , and D_k^{rx} denotes the output data size (MB) of task node v_k . The edge e_{kl} represents the precedence constraint between task nodes. This means that task v_l can be executed only after task v_k is executed. The system time is logically divided to equal length time slots, and the time slot duration is T_{slot} . The index sets of time slots can be denoted by $\mathcal{T} = \{0, 1, \dots, \tau, \dots\}$. At the beginning of each time slot, a task node in workflow is scheduled to the edge server.

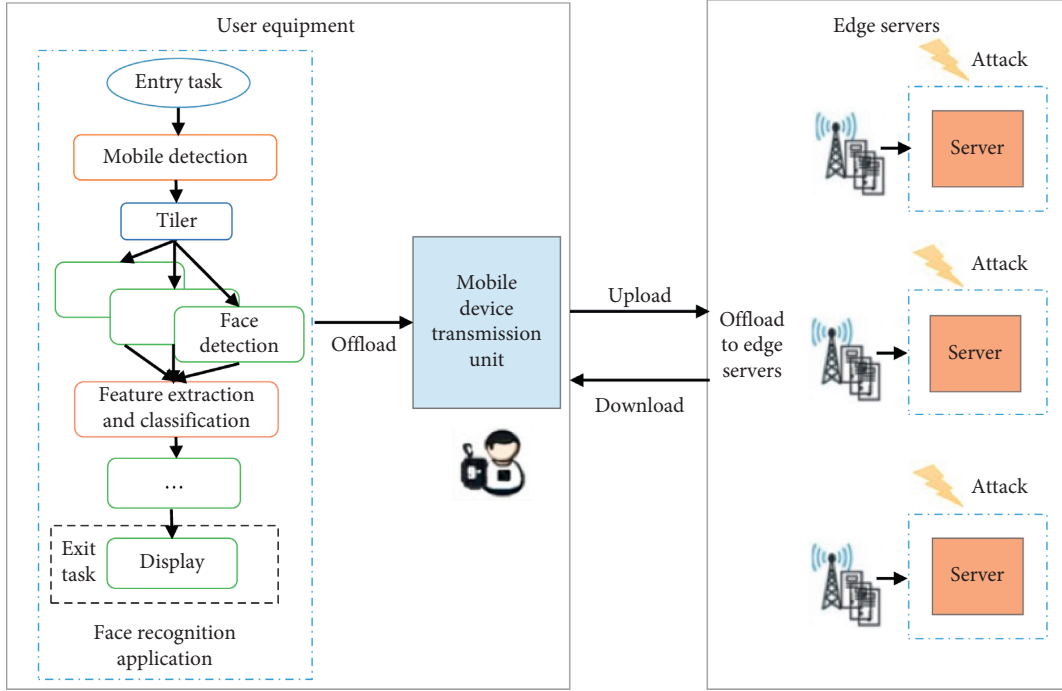


FIGURE 1: The architecture of workflow scheduling in a dynamic MEC with security threats.

3.2. Security Cost Model. The task nodes scheduled to edge servers are vulnerable to suffer from stealing and tampering security threats. In order to guard against these security threats, these task nodes need to employ encryption service cf and integrity service ig [36–38], respectively. Referring to the literature [38], encryption services cf mainly include IDEA, DES, Blowfish, AES, and RC4 algorithms. Each encryption algorithm has its own security level and encryption speed, which can be found in Table 1. The different encryption algorithms with different security levels can be flexibly selected to protect data from being stolen. Integrity services ig mainly include TIGER, RipeMD160, SHA-1, RipeMD128, and MD5 hash functions. Each hash function has its own security level and hash speed, which can be found in Table 2. The different hash algorithms with different security levels can be flexibly selected to protect data from being tampered. By flexibly selecting different encryption and hash algorithms with different security levels, an integrated security protection is formed to protect against security threats.

To ensure the security of task nodes scheduled to edge servers, the integrated security protection consisting of encryption and hash algorithms with different security levels needs to be employed. However, different security protection leads to different security cost. When the task node in the workflow is scheduled to the i th edge server, the total encryption cost on the mobile device can be calculated by [6]

$$T_{c,i}^E = \sum_{\text{type} \in \{cf, ig\}} \frac{(\varphi \cdot D_k^{tx})}{f_u \cdot N_u \cdot \text{sp}(\text{sl}_{c,i}^{\text{type}})}, \quad (1)$$

where $\varphi = 2.2$. When the task node is scheduled to the i th edge server, $\text{sl}_{c,i}^{cf}$ and $\text{sl}_{c,i}^{ig}$ denote the security levels of the

encryption service and integrity service, respectively. $\text{sp}(\text{sl}_{c,i}^{cf})$ denotes the encryption speed of encryption service with encryption level $\text{sl}_{c,i}^{cf}$. $\text{sp}(\text{sl}_{c,i}^{ig})$ denotes the hash speed of integrity service with security level $\text{sl}_{c,i}^{ig}$. When the edge server eNB_i receives the task v_k , it first decrypts the task v_k and the total decryption cost can be calculated by

$$T_{c,i}^{\text{DE}} = \frac{f_u \cdot N_u \cdot T_{c,i}^E}{f_{c,i} \cdot N_{c,i}}. \quad (2)$$

3.3. Communication Model. Due to the user's mobility, the channel state between the mobile device and different edge servers is dynamically changing. We assume that the channel state between the mobile device and the edge servers is constant in each time slot τ and is dynamically changing in different time slots. In each time slot τ , the transmission rate $R_{c,i}^u(\tau)$ between the mobile device and the i th edge server can be calculated by

$$R_{c,i}^u(\tau) = B_{c,i} \log_2 \left(1 + \frac{P_u G_{c,i}^u}{\sigma^2} \right), \quad (3)$$

where $B_{c,i}$ denotes the transmission bandwidth between the mobile device and the i th edge server, P_u denotes the transmission power of the mobile device, $G_{c,i}^u$ denotes the wireless channel gain between the mobile device and the i th edge server, and σ^2 denotes the Gaussian white noise power.

3.4. Risk Probability Model. To measure the risk degree of the task nodes scheduled to edge servers, it is necessary to establish a risk probability model to quantify the risk probability of these tasks.

TABLE 1: The encryption algorithms for confidential service.

Encryption algorithms	sl _{cf} : security level	V(sl _{cf}): processing rate (Mb/s)
IDEA	1.0	11.76
DES	0.85	13.83
Blowfish	0.56	20.87
AES	0.53	22.03
RC4	0.32	37.17

TABLE 2: The hash functions for integrity service.

Hash functions	sl _{ig} : security level	V(sl _{ig}): processing rate (Mb/s)
TIGER	1.0	75.76
RipeMD160	0.75	101.01
SHA-1	0.69	109.89
RipeMD128	0.63	119.05
MD5	0.44	172.41

Without loss of generality, referring to the literatures [36–38], the malicious attacks of data leakage and data tampering on the i th edge server are assumed to follow Poisson's distribution with parameters λ_i^{cf} and λ_i^{ig} . Therefore, the task node v_k in the workflow is scheduled to the edge server eNB _{i} , and the risk probability of data leakage or data tampering can be calculated by [6, 38]

$$P(\text{sl}_{c,i}^{\text{type}}) = 1 - \exp(-\lambda_i^{\text{type}}(1 - \text{sl}_{c,i}^{\text{type}})), \quad \text{type} \in \{cf, ig\}. \quad (4)$$

Based on the above the description, when the task v_k in the workflow is scheduled to the edge server eNB _{i} , the risk probability of the task v_k suffering from these two malicious attacks can be calculated by

$$P(v_k) = 1 - \prod_{\text{type} \in \{cf, ig\}} (1 - P(\text{sl}_{c,i}^{\text{type}})). \quad (5)$$

When the risk probability of each task v_k scheduled to the edge server does not exceed P_{\max} , the risk probability of task execution must meet the following risk constraint:

$$P(v_k) \leq P_{\max}. \quad (6)$$

3.5. Problem Formulation. In this section, we formulate the security-aware workflow scheduling problem in the mobile edge computing to be a Markov Decision Process. We first introduce the sorting strategy of workflow nodes and then define the state space, action space, and reward function of this problem. Finally, the objective function and constraints of this problem are defined.

3.5.1. Sorting of Workflow Nodes. In order to sort all the task nodes in the workflow, we assign a weight $\Pr(v_k)$ to each task node v_k [39]. The value of $\Pr(v_k)$ can be calculated by

$$\Pr(v_k) = \overline{\text{ET}(v_k)} + \max_{v_l \in \text{succ}(v_k)} \left\{ \frac{D_k^{rx}}{R_{kl}} + \Pr(v_l) \right\}, \quad (7)$$

where $\overline{\text{ET}(v_k)}$ denotes the average time of the task node v_k executing on all edge services; R_{kl} denotes the transmission

rate between edge servers, where the task node v_k and its successor node v_l are located; $\text{succ}(v_k)$ denotes the set of all successor nodes of the task node v_k . Since the edge server each task node v_k is scheduled to is not known in advance, the priority of the task node can be calculated by the average time of the task node v_k executing on all edge servers. The priorities of all task nodes in workflow can be calculated by equation (7). According to the priorities of all task nodes, these task nodes can be sorted in descending order.

3.5.2. State Space. In each time slot τ , the sorted task nodes are scheduled in turn. The edge server each task node v_k is scheduled to is dependent on the system state. The system state $s(\tau)$ in time slot τ can be denoted by

$$s(\tau) = (W_c(\tau), Q_c(\tau), G_c^u(\tau)), \quad (8)$$

where $W_c(\tau) = (W_{c,1}(\tau), \dots, W_{c,i}(\tau), \dots, W_{c,n}(\tau))$ is an n -dimensional vector, denoting the workload states of n edge server; $Q_c(\tau) = \{Q_{c,1}(\tau), \dots, Q_{c,i}(\tau), \dots, Q_{c,n}(\tau)\}$ denotes the state of the scheduled tasks in n edge servers; $G_c^u(\tau) = \{G_{c,1}^u(\tau), \dots, G_{c,i}^u(\tau), \dots, G_{c,n}^u(\tau)\}$ denotes the channel state between the mobile device and n edge servers. Specifically, $W_{c,i}(\tau)$ denotes the workload of the edge server eNB _{i} in time slot τ ; $Q_{c,i}(\tau)$ denotes a set of all task nodes scheduled to the edge server eNB _{i} in time slot τ ; $G_{c,i}^u(\tau)$ denotes the channel state between the mobile device and the edge server eNB _{i} in time slot τ .

3.5.3. Action Space. In each time slot τ , the system action $a(\tau)$ can be denoted by

$$a(\tau) = (a_c(\tau), \text{sl}_c^{cf}(\tau), \text{sl}_c^{ig}(\tau)), \quad (9)$$

where $a(\tau) = (a_{c,1}(\tau), \dots, a_{c,i}(\tau), \dots, a_{c,n}(\tau))$ is a n -dimensional vector, denoting the edge server the current task node is scheduled to. Specifically, $a_{c,i}(\tau)$ denotes whether the current task node is scheduled to the edge server eNB _{i} . If the value of $a_{c,i}(\tau)$ is 1, it denotes that the current task node is scheduled to the edge server eNB _{i} ; otherwise, it is the opposite. Note that, in each time slot τ , the current task node can only be scheduled to a single edge server. Therefore, the system action needs to meet the constraint condition $\sum_{i=1}^n a_{c,i}(\tau) = 1$. $\text{sl}_c^{cf}(\tau) = (\text{sl}_{c,1}^{cf}(\tau), \dots, \text{sl}_{c,i}^{cf}(\tau), \dots, \text{sl}_{c,n}^{cf}(\tau))$ denotes the security level of the encryption service employed by the task nodes scheduled to n edge servers. $\text{sl}_{c,i}^{cf}(\tau) \in \{0, 1.0, 0.85, 0.56, 0.53, 0.32\}$ denotes the security level of the encryption service employed by task node scheduled to the i th edge server. $\text{sl}_c^{ig}(\tau) = (\text{sl}_{c,1}^{ig}(\tau), \dots,$

$sl_{c,i}^{ig}(\tau), \dots, sl_{c,n}^{ig}(\tau)$ denotes the security level of the integrity service employed by the task nodes scheduled to n edge servers. $sl_{c,i}^{ig}(\tau) \in \{0, 1.0, 0.75, 0.69, 0.63, 0.44\}$ denotes the security level of the integrity service employed by the task node scheduled to the i th edge server.

3.5.4. Reward Function. In each time slot τ , given the system state $s(\tau)$, after taking an action $a(\tau)$, the immediate reward obtained by system is $R(\tau)$. The immediate reward $R(\tau)$ is defined as

$$R(\tau) = \begin{cases} -(\max T_{\text{end}}(M(a(\tau))) - \max T_{\text{end}}(M(a(\tau-1)))) & \text{if } \tau \neq 0, \\ -T_{\text{end}}(M(a(0))) & \text{if } \tau = 0, \end{cases} \quad (10)$$

where $M(a(\tau)) = v_k$ denotes that, in time slot, the task node scheduled by taking the action $a(\tau)$ is v_k . $\max T_{\text{end}}(M(a(\tau)))$ denotes the execution delay of the workflow until the τ th time slot, and $R(\tau)$ denotes the increment of the workflow execution delay after scheduling the task in time slice τ .

When the task node v_k is scheduled to the edge server eNB _{i} , the latest completion time $T_{\text{end}}(v_k)$ is needed to be calculated. In order to calculate $T_{\text{end}}(v_k)$, it is necessary to calculate the start time $T_{\text{start}}(v_k)$ of the task node v_k , the encryption time $T_{c,i}^E$ of the task node v_k on the mobile device, the transmission time $T_{c,i}^{\text{trans}}$ of the task node v_k transmitted from the mobile device to the edge server eNB _{i} , the waiting time $T_{c,i}^{\text{wait}}$ of the task node v_k on the edge server eNB _{i} , the decryption time $T_{c,i}^{\text{DE}}$ of the task node v_k on the edge server eNB _{i} , and the execution time $T_{c,i}^{\text{exec}}$ of the task node v_k on the edge server eNB _{i} . In general, there may be multiple predecessor nodes for a task node v_k . Therefore, in order to calculate the start time $T_{\text{start}}(v_k)$ of task node v_k , it needs to calculate the maximum sum of the completion time $T_{\text{end}}(v_h)$ and the transmission time $T_{h,k}^{\text{tr}}$ for all the predecessor nodes v_h of the task node v_k . $T_{\text{start}}(v_k)$ and $T_{\text{end}}(v_k)$ can be calculated by equations (11) and (12), respectively:

$$T_{\text{start}}(v_k) = \max_{v_h \in \text{pre}(v_k)} \{T_{\text{end}}(v_h) + T_{h,k}^{\text{tr}}\}, \quad (11)$$

$$T_{\text{end}}(v_k) = T_{\text{start}}(v_k) + T_{c,i}^E + T_{c,i}^{\text{trans}} + T_{c,i}^{\text{wait}} + T_{c,i}^{\text{DE}} + T_{c,i}^{\text{exec}}, \quad (12)$$

where $\text{pre}(v_k)$ denotes the set of all predecessor nodes of the task node v_k ; v_h is a predecessor node of v_k . $T_{\text{end}}(v_h)$ is the completion time of the task node v_h ; $T_{h,k}^{\text{tr}}$ is the transmission time between the scheduled node v_k and its predecessor node v_h .

When the task nodes are scheduled to different edge servers, they will be exposed to different risk probabilities, thereby incurring different start time and different completion time. Therefore, this paper needs to find an optimal scheduling strategy π^* in a dynamic MEC with security threats, the main goal of which is to minimize the completion time of the workflow while satisfying the risk probability of the task nodes.

$$\text{Maximize: } R(\tau), \quad (13)$$

$$\text{Subject to: } P(v_k) \leq P_{\text{max}}, \quad \forall v_k \in V. \quad (14)$$

The objective of this paper can be denoted by equation (13). The risk probability constraint of the task node can be denoted by equation (14).

Due to the fact that the MEC environment is dynamical, and its state change is unknown (such as the gain state of the wireless channel), it is difficult for traditional optimization methods to solve the security-aware workflow scheduling problem in a dynamic MEC with security threats. However, the deep reinforcement learning algorithm, as a model-free machine learning approach, is good at solving such dynamic stochastic optimization problems. In the next section, the deep reinforcement learning-based security-aware workflow scheduling scheme is introduced in detail.

4. Deep Reinforcement Learning-Based Security-Aware Workflow Scheduling Scheme

The security-aware workflow scheduling problem in a dynamic MEC with security threats is formulated to be a finite Markov Decision Process. The action space of this problem is discrete. To solve the optimal workflow scheduling scheme, this paper proposes a SAWS scheme based on deep Q network (DQN).

As shown in Figure 2, the DQN framework consists of three main functional components: (1) the evaluated Q network: the evaluated Q network is consisting of one input layer, one hidden layer, and one output layer. The number of neurons in the input layer is equal to the number of dimensions of the state, the number of neurons in the hidden layer is taken as 2048 in this paper, and the number of neurons in the output layer is equal to the number of dimensions of the action. (2) The target Q network: the structure of the target Q network is the same as that of the evaluated Q network. To continuously approach the Q function, the parameters of the target Q network are periodically updated by the parameters of the evaluated Q network. (3) The replay memory: the function of replay memory is to store these state transition experiences $\langle s(\tau), a(\tau), R(s(\tau), a(\tau), s(\tau+1)) \rangle$. A minibatch of state transition experiences are randomly chosen from the replay memory to train the Q network in the direction of minimizing a sequence of the loss function. The detailed processes of deep Q-network-based SAWS scheme are described in Algorithm 1.

During the training stage, the system state $s(\tau)$ in each time slot τ is first observed and fed into the evaluated Q network. Then, the evaluated Q network computes the evaluated Q values $Q(s(\tau), \cdot)$ for all possible actions $a(\tau)$ corresponding to the system state $s(\tau)$. The action with the largest Q value is chosen with $(1 - \epsilon)$ probability, and the action is chosen randomly with ϵ probability, and the immediate reward $R(s(\tau), a(\tau))$ can be calculated. Next, the system state $s'(\tau+1)$ in the next time slot $(\tau+1)$ can be

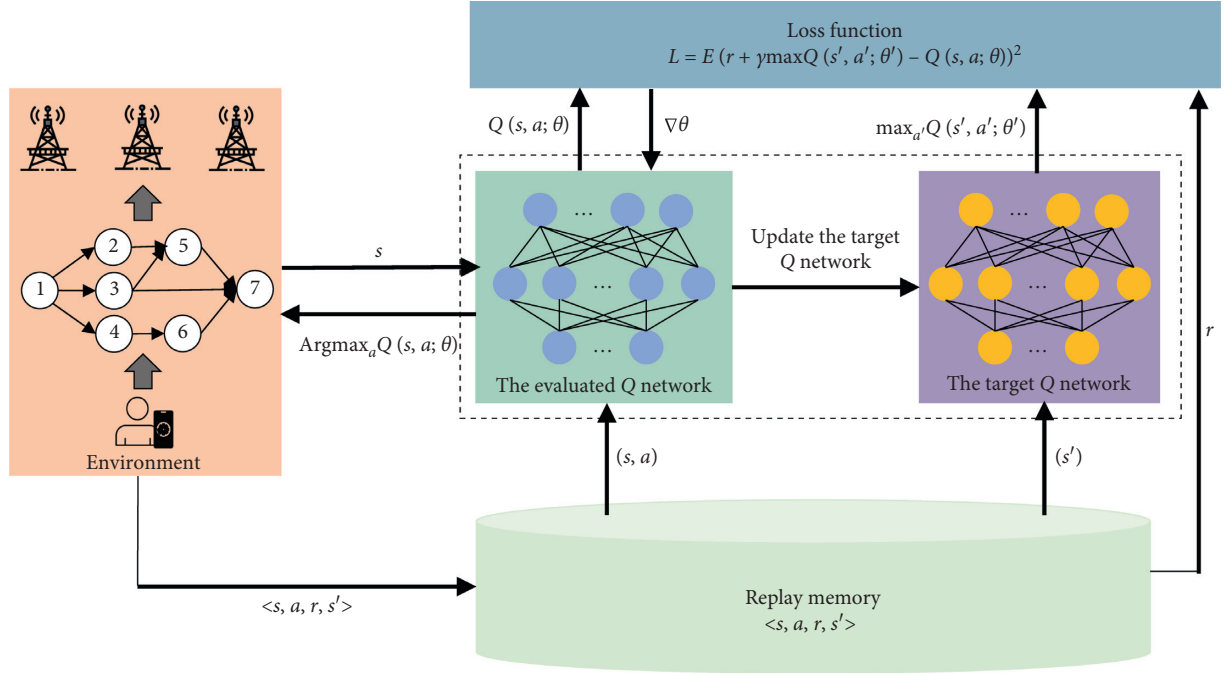


FIGURE 2: Deep Q-network-based security-aware workflow scheduling strategy.

BEGIN

- (1) Initialize the replay memory with the size of `buf_size`, and a minibatch of the state transition experiences with the size of `mini_batch`;
- (2) **for** episode = 1: `MAX_EPI` **do**
- (3) Resetting the system state $s(\tau)$;
- (4) **for** step = 1: `K` **do**
- (5) At the beginning of each time slot τ , the current state $s(\tau)$ of the system is observed;
- (6) Based on the current state $s(\tau)$, randomly select an action with ϵ probability and select the action $a(\tau)$ with the largest Q value with $(1 - \epsilon)$ probability;
- (7) The immediate reward $R(s(\tau), a(\tau))$ can be calculated and the system state $s'(\tau)$ in the next time slot $(\tau + 1)$ can be observed;
- (8) The state transition experience $\langle s(\tau), a(\tau), R(s(\tau), a(\tau), s'(\tau + 1)) \rangle$ can be obtained and stored into the replay memory;
- (9) The immediate rewards $R(s(\tau), a(\tau))$ at each step are accumulatively summed;
- (10) Randomly sample `mini_batch` state transition experiences from the replay memory to train the Q network;
- (11) Calculate the expectation of the mean-squared error between the current evaluated Q value $Q(s(\tau), a(\tau); \theta_\tau)$ and the target Q value $R(s(\tau), a(\tau)) + \gamma \max_{a(\tau+1)} Q(s(\tau + 1), a(\tau + 1); \theta_{\tau+1})$;
- (12) **end for**
- (13) **end for**

ALGORITHM 1: Deep Q network-based security-aware workflow scheduling scheme.

observed, and the state transition experience $\langle s(\tau), a(\tau), R(s(\tau), a(\tau), s'(\tau + 1)) \rangle$ can be obtained and stored into the replay memory with size `buf_size`. Finally, a minibatch of samples are randomly selected from the replay memory to train the Q network in the direction of

minimizing the loss function $L(\theta_\tau)$ and the corresponding network parameters θ_τ are saved.

The loss function $L(\theta_\tau)$ is defined as the expectation of the mean-squared error between the current evaluated Q value $Q(s(\tau), a(\tau); \theta_\tau)$ and the target Q value $R(s(\tau), a(\tau)) + \gamma \max_{a(\tau+1)} Q(s(\tau + 1), a(\tau + 1); \theta_{\tau+1})$:

$$L(\theta_\tau) = E \left[\left(R(s(\tau), a(\tau)) + \gamma \max_{a(\tau+1)} Q(s(\tau + 1), a(\tau + 1); \theta_{\tau+1}) - Q(s(\tau), a(\tau); \theta_\tau) \right)^2 \right]. \quad (15)$$

During the testing stage, the system state is first reset, and the learned network parameters are loaded. Then, at the beginning of each time slot, the current system state $s(\tau)$ is observed and fed into the trained neural network. Next, the neural network selects an optimal action $a(\tau)$ for the system state $s(\tau)$ and the corresponding reward is calculated.

5. Experimental Evaluation

To demonstrate the effectiveness of the proposed SAWS scheme in this paper, a lot of comparative experiments can be conducted. In this section, the simulation parameters are first set. Then, MSAWS, AWM, Greedy, and HEFT baseline algorithms are introduced. Finally, the performance of the SAWS scheme in comparison with these four baseline algorithms is analyzed under different simulation parameters.

5.1. Parameter Settings. This paper mainly considers a mobile edge computing system consisting of a mobile user U and n edge servers. Different workflow applications generated on the mobile device need to be scheduled in a dynamic MEC with security threats. Referring to the literatures [6, 7], the detailed parameter settings in experiment are introduced as follows:

- (1) The parameter settings of the mobile device: the CPU frequency f_u and the CPU core number N_u of the mobile device are set to $f_u = 2.5$ GHz and $N_u = 4$, respectively.
- (2) The parameter settings of edge servers: the number of edge servers is set to $n = 4$. The CPU frequencies of five edge servers are set to $f_{c,1} = \dots = f_{c,5} = 2.5$ GHz. The numbers of CPU cores are $N_{c,1} = 6$, $N_{c,2} = 7$, $N_{c,3} = 8$ and $N_{c,4} = 9$, respectively. The risk coefficients of confidentiality service for these five edge servers are $\lambda_{c,1}^{cf} = 1.8$, $\lambda_{c,2}^{cf} = 2.1$, $\lambda_{c,3}^{cf} = 2.4$ and $\lambda_{c,4}^{cf} = 2.7$, respectively. And the risk coefficients of integrity service for these five edge servers are $\lambda_{c,1}^{ig} = 1.2$, $\lambda_{c,2}^{ig} = 1.5$, $\lambda_{c,3}^{ig} = 1.8$ and $\lambda_{c,4}^{ig} = 2.1$, respectively.
- (3) The communication parameter settings: the transmission power of each edge server is $P_{c,i} = 40$ W, the maximum bandwidth is $B_{c,i} = 100$ MHz, the white Gaussian noise power is $\sigma^2 = -174$ dBm/Hz, the path loss constant is $\partial = 2$, the path loss exponent is $\theta = 4$, and the reference distance is $d_o = 1$ m [6, 7]. The distance between the mobile device and each edge server is $d_i \in (0, 350]$ m.
- (4) The parameter settings of workflow: the number of task nodes in different workflows is set to 50, 100, and 150, respectively. The out degree or in degree of each intermediate task node is less than 5, and every two task nodes can be connected with 10% probability to form an edge. The workload W_k of each task node v_k is in the range of 1GHz · s to 10 GHz · s. The input data size D_k^{in} of each task node v_k is in the range of 10 MB to 100 MB, and its output data size D_k^{out} is

set from 1 MB to 10 MB. The maximum risk probability of each task node v_k is $P_{\max} = 0.4$.

- (5) The parameter settings of the neural network: the evaluated Q network is consisting of one input layer, one hidden layer, and one output layer, and the number of neurons in the hidden layer is 2048. The learning rate is 0.003, and the learning discount factor γ is 0.9. The size `buf_size` of the replay memory is 3000, and the size `mini_batch` of the state transition experiences randomly sampled from the replay memory is 64. The maximum value of episodes is set to `MAX_EPI` = 1000. The maximum value K of steps in each episode is equal to the number of task nodes in workflow.

5.2. Performance Analysis. To demonstrate the effectiveness of the proposed SAWS scheme, this paper implements MSAWS, AWM, Greedy, and HEFT baseline algorithms and compares the SAWS scheme with these four baseline algorithms under different experimental parameters.

Average Workload Minimization (AWM): In each time slot, the AWM strategy chooses the edge server with the smallest average workload to schedule the task node.

SAWS: This abbreviation represents a security-aware workflow scheduling scheme. Its main goal is to minimize the completion time of workflow while satisfying the risk probability constraint.

MSAWS: Based on the SAWS scheme, the security service with the security level 1 is chosen for these scheduled task nodes.

Greedy: In each time slot, the Greedy algorithm selects the edge server that enables each scheduled task node to complete at the earliest based on the current environment.

HEFT [40]: This abbreviation represents heterogeneous earliest finish time. This algorithm is a workflow scheduling strategy based on list and is widely used in workflow scheduling. It first needs to calculate the priority of task nodes based on their computational and communication costs. Then, the task node is scheduled to the server that can complete it at the earliest.

5.2.1. The Convergence Analysis of SAWS. Three different types of workflows with 50, 100, and 150 task nodes are scheduled by the SAWS scheme. Figure 3 shows their learning curves, respectively. It can be observed that the completion time gradually decreases and tends to be stabilized with the increasing of learning time (i.e., the number of Episodes). This result indicates that the proposed SAWS scheme can learn an optimal policy to schedule workflow applications with different task nodes. The optimal policy can minimize the completion time of workflow while satisfying risk probability constraint. Moreover, as shown in Figure 3, it can be further observed that the completion time

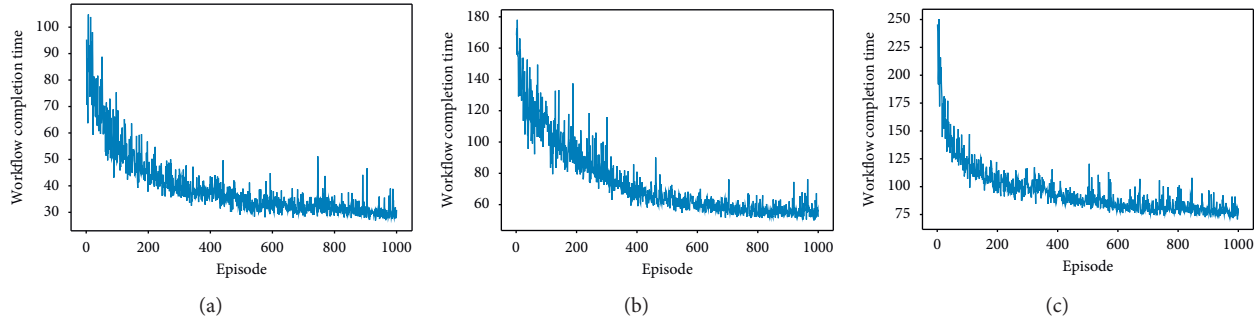


FIGURE 3: Learning curves with respect to different workflows. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

of workflow application with 50 task nodes is smallest, that of workflow application with 100 task nodes is medium, and that of workflow application with 150 task nodes is the largest. This is because the larger the scale of the workflow application, the larger the completion time.

5.2.2. The Impact of Different Risk Probabilities. To examine the impact of different risk probabilities on the completion times of different workflows, the risk probability is varied from 0.2 to 1.0 with the increment of 0.2 for workflows with 50, 100, and 150 task nodes, respectively. Figure 4 shows the completion time of the SAWS, MSAWS, AWM, Greedy, and HEFT algorithms under different risk probabilities for workflows with 50, 100, and 150 task nodes. As shown in Figure 4(a), the completion time of the SAWS algorithm is less than that of the MSAWS, AWM, Greedy, and HEFT algorithms. The main reason is that the SAWS algorithm can learn a security-aware workflow scheduling scheme in a dynamic MEC with security threats. This scheme can make an optimal scheduling decision according to different system states, thereby minimizing the completion time of the workflow while satisfying the risk probability constraint. The AWM algorithm selects the edge server with the least average workload to execute task node; hence, it is difficult to obtain an optimal solution. Although the Greedy and HEFT algorithms select the edge server that enables the task node to execute the task node at the earliest completion, it does not consider the after effect of task scheduling and is difficult to get an optimal solution. The MSAWS algorithm always selects the security service with the security level 1 to encrypt these scheduled task nodes. The MSAWS algorithm can effectively ensure the risk probability but significantly increases the completion time of workflow application. Moreover, we can observe that the completion time of five algorithms gradually decreases with the increase of the risk probability. It is because the greater the risk probability, the lower the security service level employed by task node to ensure its risk probability, and thereby the shorter the completion time of the workflow.

In addition, we can observe from Figure 4 that the completion time of workflow gradually decreases with the increase of the number of task nodes in workflow. The reason for this is the same as discussed in Section 5.2.1.

5.2.3. The Impact of Different Security Services. To evaluate the impact of different security services on the completion times of different workflows, only encryption service or only integrity service is employed by task nodes in different workflows. For simplicity's sake, only encryption service and only integrity service are denoted by *Confi_Only* and *Integ_Only*, respectively. Figure 5 shows that the completion time of *Confi_Only* and *Integ_Only* gradually decreases with the increase of the risk probability. It can be explained that the higher the risk probability, the lower the security level employed, the higher the processing rate of the security service, and thereby the shorter the completion time of the workflow. Moreover, it can be further observed that the completion time of *Integ_Only* is shorter than that of *Confi_Only*. This is because when the security level of the encryption service is approximately equal to that of the hash service, the processing rate of the hash service is higher than that of the encryption service. At last, it can be observed from Figure 5 that, with the increase of workflow nodes, the completion times of *Confi_Only* and *Integ_Only* gradually increase. The reason for this is the same as that discussed above.

5.2.4. The Impact of Different Risk Coefficients. Figure 6 shows the impact of different risk coefficients on the completion times of different workflows. We vary the risk coefficients of stealing and tampering security threats from 0.3 to 3, with the increment of 0.3. We can observe from Figure 6 that the completion time of *Confi_Only* and *Integ_Only* gradually increases with the increase of the risk coefficient. It is due to the fact that the task nodes are attacked more frequently with the increase of risk coefficient. In order to satisfy the risk probability constraint, the security service with a higher level is employed, which leads to longer task processing delay and the completion time of workflow. Moreover, we can observe from Figure 6 that the completion time of *Confi_Only* is higher than that of *Integ_Only*. The main reason is that when the security level of the encryption service is approximately equal to that of the hash service, the processing rate of the encryption service is lower than that of the hash service, which leads to a longer task processing delay and the completion time of workflow. Finally, we can see from Figure 6 that the completion time of *Confi_Only* and *Integ_Only* gradually increases with the increase of the number of the task nodes in workflow. The reason for this is the same as that discussed in Section 5.2.1.

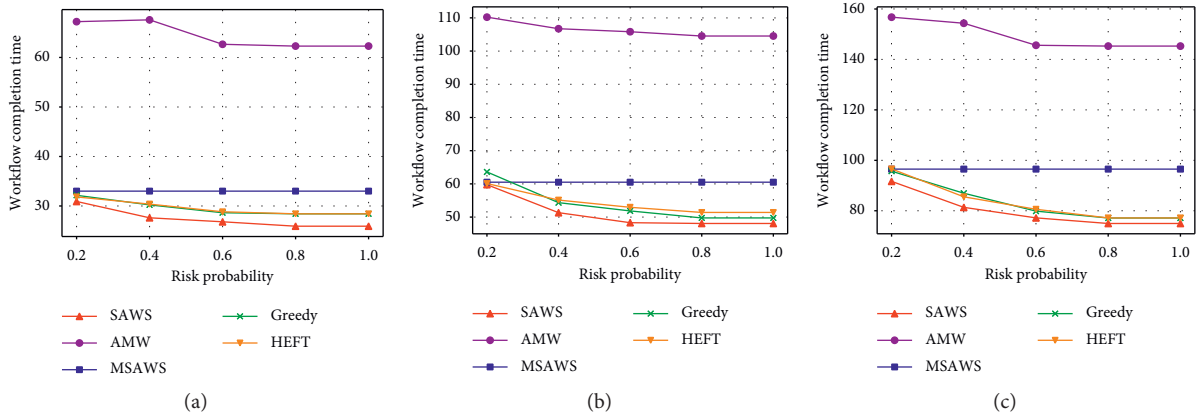


FIGURE 4: The impact of different risk probabilities. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

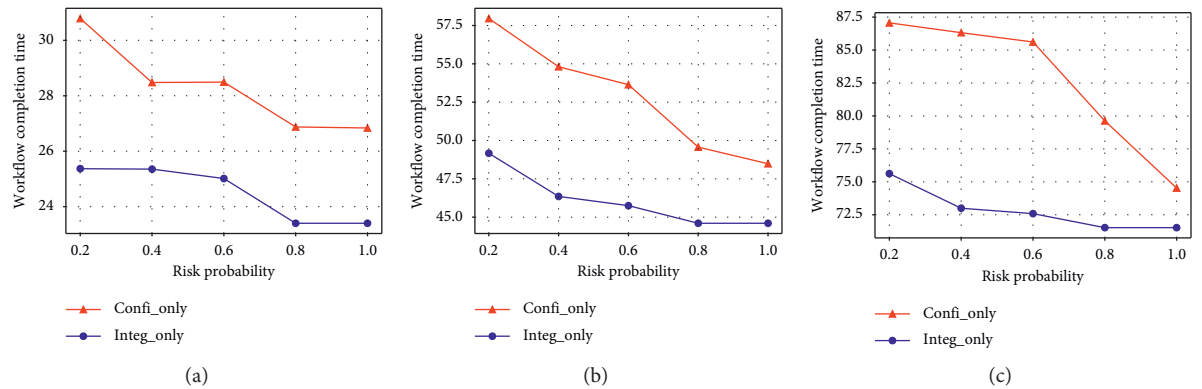


FIGURE 5: The impact of different security services. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

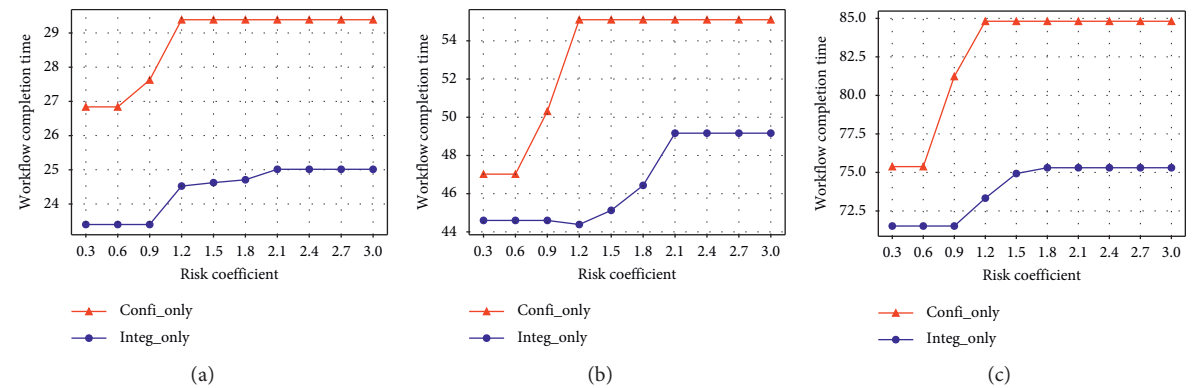


FIGURE 6: The performance of different risk coefficients. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

5.2.5. *The Impact of Different Edge Server's Computing Capacities.* Figure 7 shows the impact of different edge server's computing capabilities on the completion time of different workflows. As shown in Figure 7, we can see that the completion time of the SAWS, MSAWS, AWM, Greedy, and HEFT algorithms decreases with the increase of the number of the CPU cores. The main reason is that the more the CPU cores, the stronger the edge server's computing

capacity, and thereby the shorter the task processing delay. Therefore, the completion time of workflow gradually decreases. In addition, we can further observe from Figure 7 that the SAWS algorithm performs better than the MSAWS, AWM, Greedy, and HEFT algorithms in terms of completion time of workflow. The reason for this is the same as that discussed in Section 5.2.2. Finally, we can observe that the completion time of the SAWS, MSAWS, AWM, Greedy, and

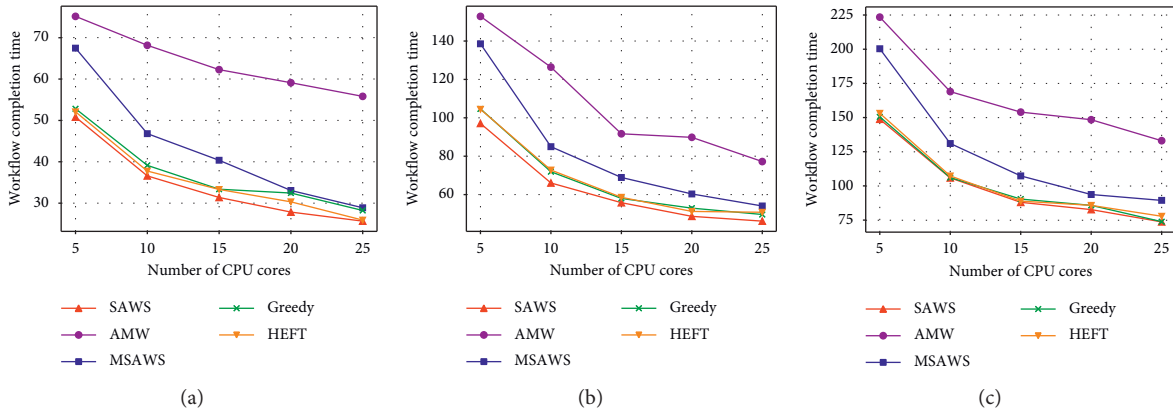


FIGURE 7: The impact of different edge server’s computing capacities. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

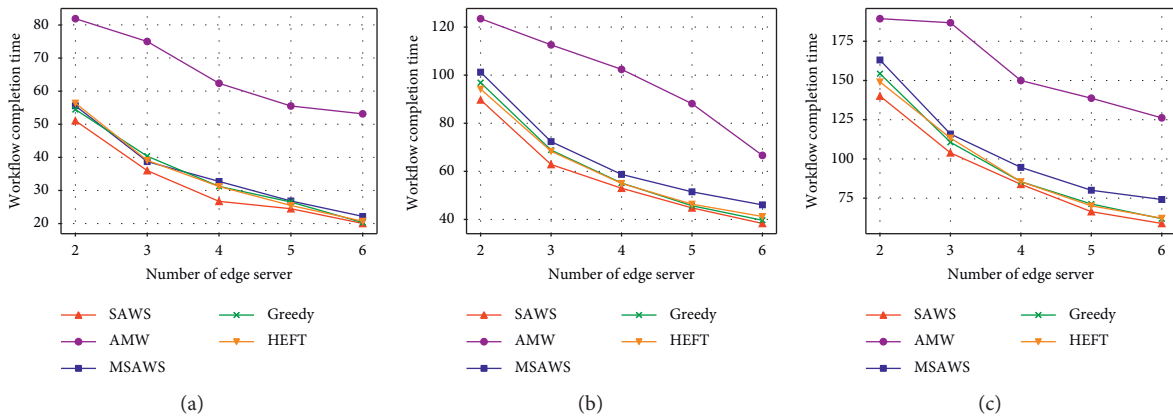


FIGURE 8: The impact of the number of edge servers. (a) The workflow with 50 nodes. (b) The workflow with 100 nodes. (c) The workflow with 150 nodes.

HEFT algorithms gradually increases with the increase of the number of task nodes in workflow. The reason for this is the same as that discussed in Section 5.2.1.

5.2.6. *The Impact of the Number of Edge Servers.* Figure 8 shows the impact of different number of edge servers on the completion time of different workflows with 50, 100, and 150 task nodes, respectively. To investigate the impact of different number of edge servers on performance, we vary the number of edge servers from 2 to 6 with the increment of 1. As shown in Figure 8, we can observe that the completion time of the SAWS, MSAWS, AWM, Greedy, and HEFT algorithms gradually decreases with the increase of the number of edge servers. It can be explained that the greater the number of edge servers, the stronger the computing capacity of the whole system, and thereby the shorter the completion time of workflow. Moreover, we can further observe that the completion time of the SAWS algorithm is lower than that of the MSAWS, AWM, Greedy, and HEFT algorithms. The reason for this is the same as that discussed in Section 5.2.5. At last, we can observe that, with the increase of task nodes in workflow, the completion times of the SAWS, MSAWS, AWM, Greedy, and HEFT algorithms

gradually increase. The reason for this is the same as that discussed above.

6. Conclusions and Future Work

This paper proposes a reinforcement learning-based security-aware workflow scheduling (SAWS) scheme to solve the workflow scheduling problem in a dynamic MEC with security threats. This paper first constructs the mobile edge computing model, security cost model, communication model, and risk probability model, respectively. Then, this paper formulates the security-aware workflow scheduling problem to be a finite Markov Decision Process. To solve this problem, this paper adopts a deep Q network approach to learn an optimal security-aware workflow scheduling policy. The SAWS scheme enables minimization of the completion time of workflows while satisfying the risk probability. To verify the effectiveness of the SAWS scheme, this paper implements the MSAWS, AWM, Greedy, and HEFT baseline algorithms and compares the SAWS scheme with these four baseline algorithms under different experimental parameters such as the risk probability, the security service, the risk coefficient, the edge server’s computing capacity, and the number of edge servers. The extensive experimental

results demonstrate the effectiveness of the proposed SAWS scheme.

Data Availability

The experiment data supporting this experiment analysis are from previously reported studies, which have been cited. The experiment data used to support the findings of this study are included within the article. The experiment data are described in Section 5 in detail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (Nos. 62002316, 61802095, 61572162, and 61572251), the Zhejiang Provincial National Science Foundation of China (Nos. LQ19F020011 and LQ17F020003), the Zhejiang Provincial Key Science and Technology Project Foundation (No. 2018C01012), and the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (No. SKLNST-2019-2-15).

References

- [1] C. Yi, J. Cai, and Z. Su, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 29–43, 2020.
- [2] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: task allocation and computational frequency scaling," *IEEE Transactions on Communication*, vol. 65, no. 8, pp. 3571–3584, 2017.
- [3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [4] Z. Yuezhi and Z. Di, "Near-end cloud computing: opportunities and challenges in the post-cloud computing era," *Chinese Journal of Computers*, vol. 42, no. 4, pp. 677–700, 2019.
- [5] C. Calero, "5Ws of green and sustainable software," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 401–414, 2020.
- [6] B. Huang, "Security modeling and efficient computation offloading for service workflow in mobile edge computing," *Future Generation Computer System*, vol. 97, pp. 755–774, 2019.
- [7] B. Huang, Y. Li, Z. Li et al., "Security and cost-aware computation offloading via deep reinforcement learning in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 3816237, 20 pages, 2019.
- [8] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4642–4655, 2018.
- [9] S. Ranadheera, S. Maghsudi, and E. Hossain, "Mobile edge computation offloading using game theory and reinforcement learning," 2017, <https://arxiv.org/abs/1711.09012>.
- [10] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, 2017.
- [11] I. A. Elgendy, W. Zhang, Y. C. Tian, and K. Li, "Resource allocation and computation offloading with data security for mobile edge computing," *Future Generation Computing System*, vol. 100, pp. 531–541, 2019.
- [12] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges," *Future Generation Computing System*, vol. 78, pp. 680–698, 2018.
- [13] T. Hongliang, Z. Yong, L. Chao, and X. Chunxiao, "Overview of research on database confidentiality protection technology in cloud environment," *Journal of Computers*, vol. 40, no. 10, pp. 2245–2270, 2017.
- [14] O. Pedreira, F. Garcia, M. Piattini, A. Cortinas, and A. Cerdeira-Pena, "An architecture for software engineering gamification," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 776–797, 2020.
- [15] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Enriching the transfer learning with pre-trained lexicon embedding for low-resource neural machine translation," *Tsinghua Science and Technology*, vol. 40, p. 1, 2020.
- [16] E. A. A. Alaoui, S. C. K. Tekouabou, S. Hartini, Z. Rustam, H. Silkan, and S. Agoujil, "Improvement in automated diagnosis of soft tissues tumors using machine learning," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 33–46, 2021.
- [17] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaïdi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.
- [18] A. Guezaz, Y. Asimi, M. Azrou, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, 2021.
- [19] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [20] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 10, p. 1, 2019.
- [21] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 570–584, 2020.
- [22] G. Chalapathi, V. Chamola, C. K. Tham, S. Gurunayanan, and N. Ansari, "An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets," *Future Generation Computer Systems*, vol. 102, pp. 862–875, 2020.
- [23] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation System*, vol. 99, pp. 1–10, 2020.
- [24] H. Wu, K. Wolter, P. Jiao, Y. Deng, Y. Zhao, and M. Xu, "EEDTO: an energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2163–2176, 2020.
- [25] J. Xu, X. Li, R. Ding, and X. Liu, "Multi-resource computing offloading strategy for energy optimization in mobile edge computing," *Computer Integrated Manufacturing System*, vol. 25, no. 04, pp. 954–961, 2019.

- [26] H. Wu, W. J. Knottenbelt, and K. Wolter, "An efficient application partitioning algorithm in mobile environments," *IEEE Transactions on Parallel Distributed System*, vol. 30, no. 7, pp. 1464–1480, 2019.
- [27] A. Zhu, S. Guo, M. Ma et al., "Computation offloading for workflow in mobile edge computing based on deep Q-learning," in *Proceedings of 2019 28th Wireless and Optical Communications Conference WOCC 2019*, July 2019.
- [28] H. Liu, "Scheduling multi-workflows over edge computing resources with time-varying performance, a novel probability-mass function and ∂ DQN-based approach," in *Proceedings of International Conference on Web Services*, pp. 197–209, Beijing, China, October 2020.
- [29] M. C. Sanchez, J. M. C. de Gea, J. L. Fernández-Alemán, J. Garcerán, and A. Toval, "Software vulnerabilities overview: a descriptive study allow," *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 270–280, 2020.
- [30] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emar, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85–101, 2020.
- [31] X. Jia, D. He, N. Kumar, and K. K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2020.
- [32] D. He, S. Chan, and M. Guizani, "Security in the internet of things supported by mobile edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 56–61, 2018.
- [33] Y. Chen, Y. Zhang, S. Maharjan, M. Alam, and T. Wu, "Deep learning for secure mobile edge computing in cyber-physical transportation systems," *IEEE Networks*, vol. 33, no. 4, pp. 36–41, 2019.
- [34] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation System*, vol. 10, pp. 1–10, 2020.
- [35] X. Xu, Q. Huang, Y. Zhang, S. Li, L. Qi, and W. Dou, "An LSH-based offloading method for IoMT services in integrated cloud-edge environment," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 16, no. 3, 2021.
- [36] H. Chen, X. Zhu, D. Qiu, L. Liu, and Z. Du, "Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds," *IEEE Transactions on Parallel Distributed Systems*, vol. 28, no. 9, pp. 2674–2688, 2017.
- [37] Y. Wu, J. Shi, K. Ni et al., "Secrecy-based delay-aware computation offloading via mobile edge computing for internet of things," *IEEE Internet Things Journal*, vol. 6, no. 3, pp. 4201–4213, 2019.
- [38] Z. Li, J. Ge, C. Li et al., "Energy cost minimization with job security guarantee in Internet data center," *Future Generation Computing Systems*, vol. 73, pp. 63–78, 2017.
- [39] Y. Qin, H. Wang, S. Yi, X. Li, and L. Zhai, "An energy-aware scheduling algorithm for budget-constrained scientific workflows based on multi-objective reinforcement learning," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 455–480, 2020.
- [40] H. Topcuoglu, S. Hariri, and M. Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel Distributed System*, vol. 13, no. 3, pp. 260–274, 2002.