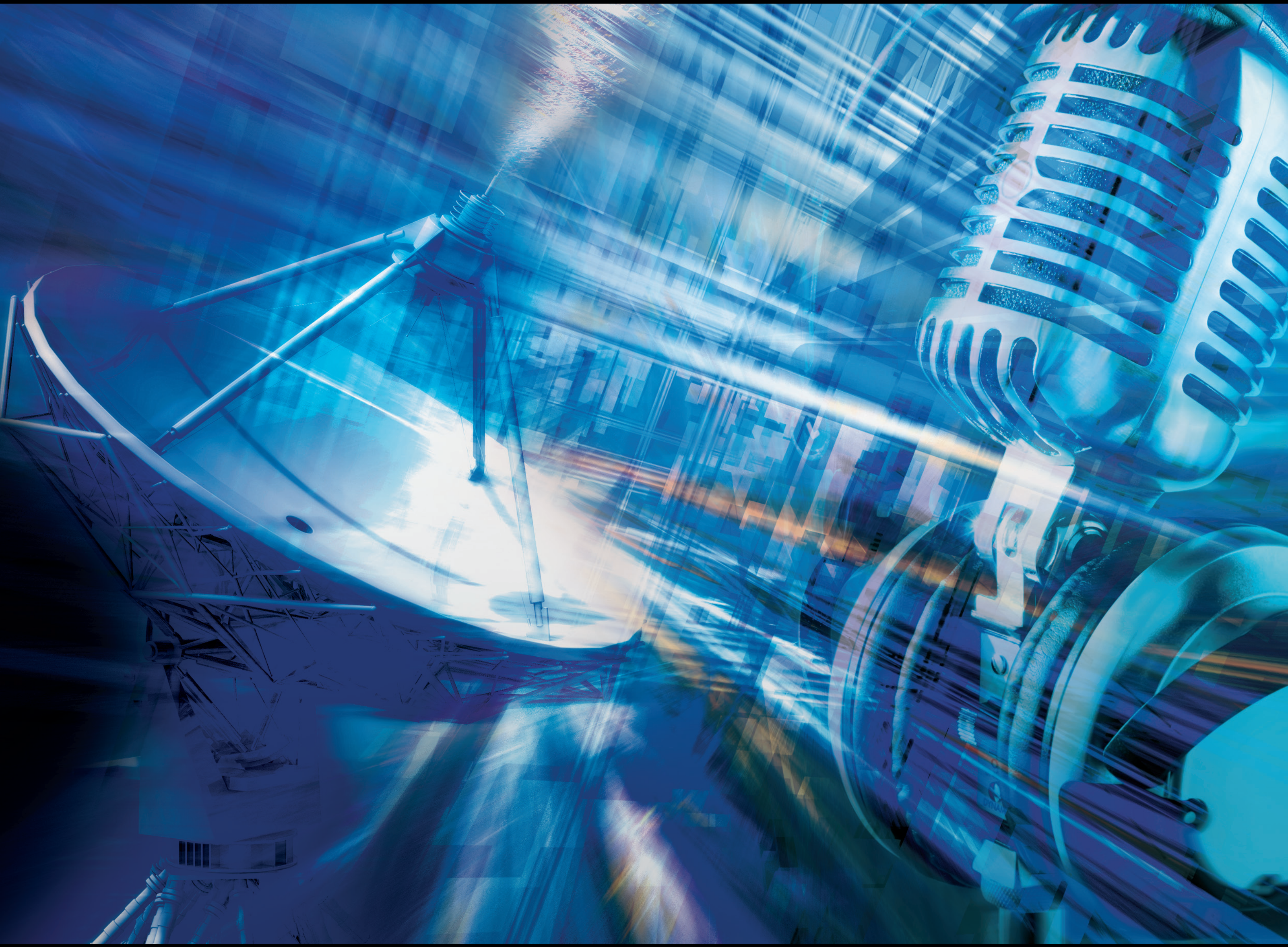
International Journal of Digital Multimedia Broadcasting

Emerging Technologies, Infrastructures, and Applications for Video Streaming over Future Communication Networks

Lead Guest Editor: Yuanlong Cao
Guest Editors: Yuming Fang, Jiyan Wu, Wei Quan, and Kai Wang



Emerging Technologies, Infrastructures, and Applications for Video Streaming over Future Communication Networks

International Journal of Digital Multimedia Broadcasting

Emerging Technologies, Infrastructures, and Applications for Video Streaming over Future Communication Networks

Lead Guest Editor: Yuanlong Cao
Guest Editors: Yuming Fang, Jiyan Wu, Wei Quan, and Kai Wang

Editorial Board

Contents

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2019, Article ID 1294380, 2 pages
https://doi.org/10.1155/2019/1294380

Editorial

Emerging Technologies, Infrastructures, and Applications for Video Streaming over Future Communication Networks

Yuanlong Cao,[1] Yuming Fang,[2] Jiyan Wu,[3] Wei Quan,[4] and Kai Wang[5]

[1] *Jiangxi Normal University, Nanchang, China*
[2] *Jiangxi University of Finance and Economics, China*
[3] *Singapore University of Technology and Design, Singapore*
[4] *University of Waterloo, Waterloo, Canada*
[5] *Tsinghua University, China*

Correspondence should be addressed to Yuanlong Cao; ylcao@jxnu.edu.cn

Received 17 December 2018; Accepted 17 December 2018; Published 25 February 2019

The objective of this special issue is to provide a premier forum for researchers working on emerging technologies and applications for video streaming to present their recent research results. A substantial number of articles were submitted, and after a thorough peer review process, eight high-quality articles were selected to be included in this special issue. These papers cover emerging technologies and applications for content-rich real-time video streaming services, including QoS prediction mechanism, multimedia content transmission technologies, multimedia object detection, and so on. We believe that the research articles collected in this special issue will introduce readers to the recent advancements and interesting research directions in the field of multimedia technologies.

The paper by G. X. Hu et al. introduces a new small image objection detection model to achieve good image detection accuracy. The proposed model not only ensures the integrity of the feature of the large object but also preserves the full detail feature of the small objects by extracting the multiscale feature of the image.

The paper by R. Liu et al. presents a novel homomorphic encryption technology-based method for verification of video data integrity. The simulation results show that the proposed method is superior to comparison schemes in all aspects, and it suggests that the proposed scheme is serving better for the video data integrity verification purpose in the cloud environment.

The paper by T. Yi and C. Fang jointly combines the Matter Element Analysis (MEA) method, fuzzy set theory, and TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) to design a fuzzy matter element model, which can be used to evaluate the complexity of UML class diagram and predict the complexity of software quality.

The paper by W. Lu et al. proposes a priority-aware multipath TCP (MPTCP)-based data distribution solution in order to provide multimedia applications with many attractive benefits including goodput performance improvement, latency reduction, and high-quality users' quality of experience for multimedia streaming services.

The paper by Y. Song et al. introduces an improved artificial bee colony (ABC) algorithm for efficient multimedia streaming over wireless network environments. The experimental results show that the proposed algorithm not only ensures the real-time transmission requirement, but also reduces the power consumption of wireless network.

The paper by Q. Liu et al. presents a novel loss-aware multipath data delivery solution, which extends to the Stream Control Transmission Protocol (SCTP)'s concurrent multipath transfer mechanism, necessitating the following aims: (i) providing SCTP-CMT with a loss-aware bandwidth aggregation and parallel transmission mechanism and (ii) improving data delivery performance in a heterogeneous wireless network environment.

The paper by W. Ma et al. first introduces a novel neural collaborative filtering model for QoS prediction using transfer learning technology. Next, the authors design a novel interaction layer to represent the relationship between latent embedding factors of the nodes. Finally, the authors utilize partial fine-tuning and maximum mean discrepancy (MMD) measurement to train the target domain model to implement domain adapting. The experimental results show that the proposed model can outperform the existing QoS prediction models in a video streaming P2P network environment.

The paper by W. Wang et al. regards a video streaming scheduling event in wireless networks as a soft real-time scheduling event and then proposes a dynamic soft real-time scheduling mechanism combined with preemption threshold for streaming media. The simulation results demonstrate the benefits of applying the proposed mechanism to a soft real-time system.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

We would like to thank all the authors who contributed to this special issue. This publication would not be possible without the participation of our expert reviewers, who provided valuable feedback and constructive comments throughout the review process.

Yuanlong Cao
Yuming Fang
Jiyan Wu
Wei Quan
Kai Wang

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2019, Article ID 9474057, 8 pages
https://doi.org/10.1155/2019/9474057

Research Article

Loss-Aware CMT-Based Multipathing Scheme for Efficient Data Delivery to Heterogeneous Wireless Networks

Qinghua Liu ,[1] Fenfen Ke,[2] Zhihua Liu,[2] and Jiaxin Zeng[3]

[1] *School of Software, Jiangxi Normal University, Nanchang, 330022, China*
[2] *Jiangxi Modern Polytechnic College, Nanchang, 330022, China*
[3] *Division of Intelligence and Computing, Tianjin University, Tianjin, 300354, China*

Correspondence should be addressed to Qinghua Liu; 004799@jxnu.edu.cn

Received 7 September 2018; Accepted 13 November 2018; Published 20 February 2019

Guest Editor: Jiyan Wu

With the rapid development of wireless networks, multiple network interfaces are gradually being designed into more and more mobile devices. When it comes to data delivery, Stream Control Transmission Protocol (SCTP)-based Concurrent Multipath Transfer (CMT) has proven to be quite useful solution for multiple home networks, and it could become the key transport protocol for the next generation of wireless communications. The CMT delay caused by data rearrangement has been noticed by researchers, but they have seldom considered the frequent occurrence of packet loss that occurs in the high-loss networks. In this paper, we proposed an original loss-aware solution for multipath concurrent transmission (CMT-LA) that achieves the following goals: (1) identifying packet loss on all paths, (2) distributing packets adaptively across multiple available paths according to their packet loss and loss variation, and (3) maintaining the features of bandwidth aggregation and parallel transmission of CMT while improving the throughput performance. The results of our simulations showed that the proposed CMT-LA reduces reordering delay and unnecessary fast retransmissions, thereby demonstrating that CMT-LA is a more efficient data delivery scheme than classic CMT.

1. Introduction

The development of wireless network technologies has been extremely rapid, leading to an increased amount of mobile equipment with multiple communication interfaces (e.g., WiFi, 4G, 5G) [1]. To support the wide use of wireless communication technology and the prevalence of multihoming terminals, properly equipped mobile devices can use multiple network interfaces to transfer network data, thereby increasing transmission efficiency, maximizing network resource utilization, and improving system robustness [2]. However, the traditional Transmission Control Protocol (TCP), with its single path structure, cannot make use of the multihoming characteristics of mobile terminal equipment to improve the data transmission rate and throughput performance.

Multihoming devices are capable of utilizing multiple wireless technologies to improve the content delivery of multimedia [3]. Moreover, users find it convenient to access multimedia streaming services (e.g., video and voice streaming) from anywhere at any time. Nevertheless, supporting real-time multimedia streaming services remains a challenging task, mainly because these applications require high bandwidth and delay intolerance [4].

More importantly, real-time services performance of diverse media also requires low latency and reliable data transmission. Being a solution to support multipath transmission, Stream Control Transmission Protocol (SCTP) [5] has already discussed by scholars [6, 7]. Like TCP, SCTP is a reliable protocol that offers dependable data transfer. In addition, SCTP can satisfy the bandwidth requirements for content-rich real-time multimedia streaming delivery, but only one primary path can be selected. Therefore, SCTP maintains the reliability of transmission but cannot transmit data in parallel paths. For this reason, SCTP may not be able to provide maximum utilization of network resources, and it barely meets the demand for efficient data delivery needed by next generation mobile communications.

Nowadays, CMT has been recognized as a promising means for improving network resource utilization. It not only supports the transmission of available paths in parallel, but

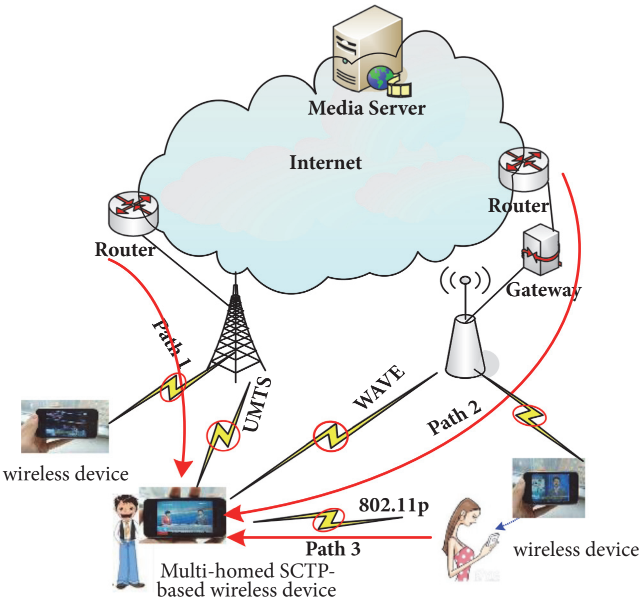


Figure 1: CMT-based multimedia streaming in heterogeneous wireless network.

also aggregates bandwidth. By applying SCTP's multihomed architecture, CMT can use heterogeneous network interfaces to transmit data among multiple paths simultaneously [8]. Figure 1 depicts a typical CMT-based heterogeneous wireless scenario for multimedia content delivery. From Figure 1, we can see that a multihoming wireless device can communicate with a media server via three paths simultaneously, illustrating how CMT supports heterogeneous wireless networks for delivery of real-time multimedia content. Consequently, CMT-based association paths can offer attractive benefits, such as end-to-end throughput, utilization of network resources, and load balancing.

However, critical issues remain to be considered regarding classic CMT because this approach primarily adopts a plain round-robin process for distributing application layer data to the multiple available paths. CMT rarely takes into consideration the quality differences of various paths in terms of quality-of-service (QoS)-related parameters. Thus, a plain round-robin distribution method inevitably results in receiver buffer blocking [9–12]. In order to reduce such blocking and achieve more reliable transmission, we try our best to ensure that the receiver will transmit ordered packets to the upper layer. Therefore, the receiver buffer must firstly store the out-of-order packet and wait for the arrival of the preamble packets (e.g., lost or delayed packets) before sending the data. Considering the characteristic of limited storage capacity, excessive place disorder packets will certainly lead to a severe buffer blocking problem. Once this happens, buffer blocking will affect the quality of the users experience, especially for content-rich streaming media.

To address the above issues, we proposed an original loss-aware packet scheduling scheme under the structure of CMT called CMT-LA. We designed CMT-LA to meet three functional goals: (1) to identify packet loss for the paths, (2) to distribute data packets across the multiple available paths adaptively according to their packet loss and loss variation, and (3) to maintain the features of bandwidth aggregation and parallel transmission of CMT while improving the throughput performance. Our research provided the following contributions to the field:

(i) The new model introduced an optimal packet loss detection algorithm to estimate the paths packet loss rate (PLR).

(ii) The proposed solution combined the PLR estimation and the jitter indicator of PLR at the transport layer to reflect the data transmission condition of the paths.

(iii) By utilizing the path-based loss-aware method, this novel, fast data distribution model improved the efficiency of data transmission and reduced packet loss differences.

In Section 2, readers can have a general understanding for currently research. Section 3 describes CMT-LA solution. In Section 4, we provide details of our simulations and an analysis, and Section 5 presents our conclusion and intentions for future work.

2. Related Work

For the transport layer, SCTP is recognized to be more reliable than most other transport protocols, which breaks TCPs limitations while retaining the advantages of the User Datagram Protocol (UDP). Nevertheless, unlike TCP and UDP, SCTP is equipped with two capacities, multihoming and multistreaming, that help to increase availability [13]. Liu et al. [9] developed a comprehensive retrospective view of SCTP and then discussed three aspects of the protocol: managing switching with an integrated approach, multipath transmission in a concurrent way, and cross-activity between layers. Baharudin et al. [14] presented a novel path selection solution based on ant colony optimization aimed at enhancing the efficiency of SCTPs selection of a primary path. Dreibholz et al. [15] provided an overview of SCTP and its extensions and then focused on the continuous SCTP standardization procedure.

While CMT is viewed as a promising means for enhancing the efficiency of data transmission, research has sought further improvements. Iyengar et al. [16] described three negative side effects of CMT and introduced three algorithms to avoid these side effects effectively. In addition, their paper proposed and estimated five retransmission policies for CMT. Yang et al. [17] introduced the limitations for throughput of CMT: receiving buffer size and the longest round trip time (RTT) in all transmission paths. The authors established a CMTs throughput model to analyze the selection of different paths via it and then applied the analysis results to selection strategy on the basis of available paths. Therefore, the throughput of the CMT has improved. Each of the above solutions is different in terms of data handling capabilities. However, all of the solutions continued to adopt the round-robin policy for transmitting data over all available paths.

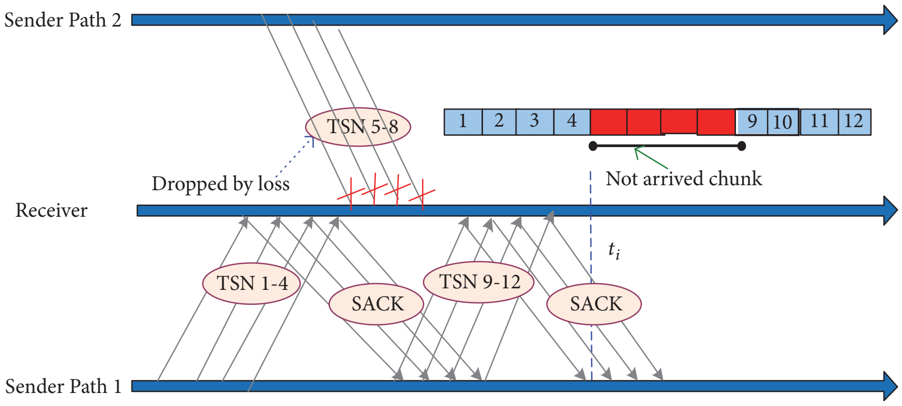


FIGURE 2: The delayed delivery occurred because of packet loss.

Recently, a growing volume of research has been directed toward applying the CMT technique for multimedia data delivery. Xu et al. [18] sought to improve the performance of real-time multimedia data distribution by adopting multihoming SCTP while making use of different Single Path Transfer (SPT) and CMT mechanisms separately. Within a certain loss range, Baek et al. [19] presented a multicast protocol based on improved SCTP. Its main purpose is to achieve partial reliability, reduce the message overhead dramatically, and tolerate partial loss. Huang et al. [20] attempted to conjunct with various techniques, including a prioritized stream, to provide services for multimedia streaming. Therefore, PR-CMT was produced on the basis of CMT with the characteristic of new and partial reliability. However, in general, the existing research failed to consider that path diversity is bound to data reordering, which severely affects CMT performance.

In recent years, many researchers have attempted to employ a cross-layer concept to solve some of the problems that may result from CMT [21–25]. Cao et al. [21] presented an innovative cross-layer QoS-aware adaptive CMT (CMT-CQA) to satisfy three requirements: identification of wireless errors, mitigation of buffer blocking problems, and efficient bandwidth aggregation. CMT-CQA was able to aggregate bandwidth by considering multiple factors, depict a suitable multimedia delivery strategy, and take on a reinforced fast recovery scheme. Xu et al. [23] designed an optimal solution for monitoring and analyzing the quality of paths. This method cleverly utilizes the concept of cross-layer based on SCTP and the fairness driving characteristics based on CMT, which improves the users experience of multimedia streaming services while maintaining fairness of competing TCP flows.

In addition, many researchers have been paying attention to CMT-based data reordering problems. Xu et al. [26] took Network Coding (NC) ideas to improve the performance of SCTP-based CMT and thus introduced an enhanced version of multipath concurrent transmission scheme, aiming to avoid data reordering and alleviate buffer blocking. Cao et al. [27] designed an adaptive receiver-cooperative path aggregation model to reduce data rearrangement and prevent buffer blocking. However, the above work did not consider the high-loss situation which suddenly lost packets in actual networks. Undoubtedly, if the losses occurred and CMT cannot recognize them, there is no good throughput performance.

In this paper, we have attempted to fill the gap by developing CMT-LA, an approach that includes a proper packet loss detection module along with a novel loss-aware packet scheduler to ensure that the SCTP packets arrive in order.

3. The CMT-LA Solution

Asymmetric paths with multihomed communications may suffer from disparate undesirable characteristics including packet loss rates, delays, round trip times, and bandwidth. These conditions may lead to out-of-order data and degraded data delivery. In addition, the packet loss rate is generally the most important feature of a transmission path considered when CMT utilizes all available paths to send data in parallel. To provide a better understanding of blocked data transmission in heterogeneous wireless networks, Figure 2 provides a representation of how delayed delivery occurs as a result of packet loss.

We record transmission sequence number transmitted by the sender as TSNs. It can be seen from Figure 2 that Path 1 has TSNs 1-4 and 9-12, and the transmitted sequence in Path 2 is TSNs 5-8. Due to differences in path quality loss, the packets of TSNs 5-8 at Path 2 that cannot arrive at time t_i will be dropped. While groups for TSNs 1-4 and 9-12 can get it before time t_i, so it is necessary to place them in the receiver buffer for reordering. However, the out-of-order packets will not be submitted because of the reliability of CMT. In this way, packet loss leads to out-of-order data problems and eventually degrades the overall performance.

To alleviate the differences of packet loss and improve the efficiency of data delivery, we extended our $R_{tx^+}+$ solution to propose CMT-LA, a packet loss detection and packet scheduling solution. This approach aimed to identify the packet loss of the various paths and then distribute data packets adaptively across multiple available paths according to their packet loss. CMT-LA can be associated with existing CMT solutions.

By using the Mathis model [21], the goodput of path P_i, which is denoted as G_{P_i}, can be calculated by

$$G_{P_i} = \frac{M}{RTT_{P_i} \times \sqrt{PLR_{P_i}}}, \tag{1}$$

where PLR_{P_i} denotes the packet loss rate of path P_i. RTT_{P_i} is path P_i's RTT, which can be calculated as

$$\begin{aligned} RTT_{P_i} &= \omega \times RTT_{P_i} + (1-\omega) \times \Phi_{P_i}, \\ \Phi_{P_i} &= t - T_{send} - \Delta T \end{aligned} \tag{2}$$

where RTT_{P_i} presents the current RTT value of path P_i. T_{send} is a timestamp to record the transmit time, while the timestamp t is the time when receiver sends Selective Acknowledgment (SACK) chunk to the sender. ω stands for the weighting parameter and its value is set to 7/8 by default according to [27]. ΔT represents time interval at which the receiver processes each packet. Meanwhile, we can estimate the PLR_{P_i} value from (1), so PLR_{P_i} can be obtained as shown:

$$PLR_{P_i} = \left(\frac{M}{G_{P_i} \times RTT_{P_i}}\right)^2, \tag{3}$$

where M uses the constant value $1.22 \times DCS$ by default, and DCS is set to 1,500 MTUs (maximum transmission units) in the simulations, i.e., $M = 1.22 \times 1500 = 1830$.

Further, Bisoy et al. [28] proposed the Vegas model to calculate the goodput of path P_i; namely, the value of G_{P_i}, introduced into (3), can be obtained as follows:

$$G_{P_i} = \frac{cwnd_{P_i}}{RTT_{P_i}^{min}}, \tag{4}$$

where $RTT_{P_i}^{min}$ is the minimum value in all paths RTT to decrease dispersive expense. $cwnd_{P_i}$ denotes the congestion window size of path P_i.

So far, the PLR_{P_i} of each path can be estimated by using (2), (3), and (4). However, the PLR_{P_i} cannot reflect the transmission condition for path P_i objectively. To address this issue, we can use ΔPLR_{P_i}, which represents the jitter indicator of PLR_{P_i}, to identify whether path P_i is in an unreliable transmission condition. ΔPLR_{P_i} can be expressed as

$$\Delta PLR_{P_i} = \frac{PLR_{P_i} - \overline{PLR_{P_i}^{avg}}}{PLR_{P_i}^{max} - PLR_{P_i}^{min}}, \tag{5}$$

where PLR_{P_i} is the current calculated PLR on P_i, and the $\overline{PLR_{P_i}^{avg}}$ value can be obtained by

$$\overline{PLR_{P_i}^{avg}} = \frac{1}{k} \times \sum_{n=1}^{k} PLR_{P_i}^{n}. \tag{6}$$

```
Definition:
$P_i$: the $i^{th}$ path within the SCTP association.
$PLR_{Pi}$: the estimated packet loss rate of path $P_i$
for all path $P_i$ within the SCTP association do
    if status of $P_i$ == ACTIVE then
        calculated $PLR_{Pi}$ by Eq. (2)
        calculated $\Delta PLR_{Pi}$ by Eq. (5)
        put $P_i$ into $P_{list}$
    end if
end for
```

Algorithm 1

Assuming the measured PLR values on P_i are $\{PLR_{P_i}^1, PLR_{P_i}^2, \ldots, PLR_{P_i}^k\}$, then $PLR_{P_i}^{max}$ and $PLR_{P_i}^{min}$ can be acquired as shown

$$\begin{aligned} PLR_{P_i}^{max} &= \max\left\{PLR_{P_i}^1, PLR_{P_i}^2, \ldots, PLR_{P_i}^k\right\} \\ PLR_{P_i}^{min} &= \min\left\{PLR_{P_i}^1, PLR_{P_i}^2, \ldots, PLR_{P_i}^k\right\}. \end{aligned} \tag{7}$$

As mentioned previously, ΔPLR_{P_i} can reflect the transmission condition for path P_i. Therefore, we determine that P_i is in an unreliable transmission condition if ΔPLR_{P_i} and $(\Delta PLR_{P_i})_{t_n} - (\Delta PLR_{P_i})_{t_{n-1}} > 0$ (where t_n and t_{n-1} are the observed time, and $t_n > t_{n-1}$). In other words, when $\Delta PLR_{P_i} > 0$, the increase of ΔPLR_{P_i} indicates that P_i is in an unreliable transmission condition. This detection method for packet loss is described in detail in Algorithm 1, which shows the pseudocode for the CMT-LA method. Finally, we can identify packet loss for the paths and distribute data packets adaptively according to the PLR and ΔPLR values, which also meets the goals of CMT-LA.

Algorithm 1 is one part of CMT-LA method and its pseudocode shows how we estimate the PLR and ΔPLR value of each path in the SCTP association.

As discussed above, the classic CMT adopts a plain round-robin method for data distribution on the multiple transmission paths within a multihomed SCTP association.

However, this approach seldom considers highly dissimilar path characteristics over a heterogeneous wireless network. For this reason, using the plain round-robin method may result in disorder packets and blocking the receiver buffer. Inspired by the fact that the number of sending packets is related mainly to the PLR value, we designed an optimal CMT-LA data distribution algorithm that includes the following steps.

(1) Use Algorithm 1 to estimate the PLR and ΔPLR values of all available paths for the ACTIVE state in the SCTP association.

(2) Arrange paths P_{list} in ascending order according to the corresponding PLR value.

(3) Choose the first path ($P_{list(0)}$) in P_{list} as the candidate path (P_{send}) for packets transmission.

(4) If two or more paths have the smallest P_{send}, select the path with the smallest ΔPLR as the P_{send}.

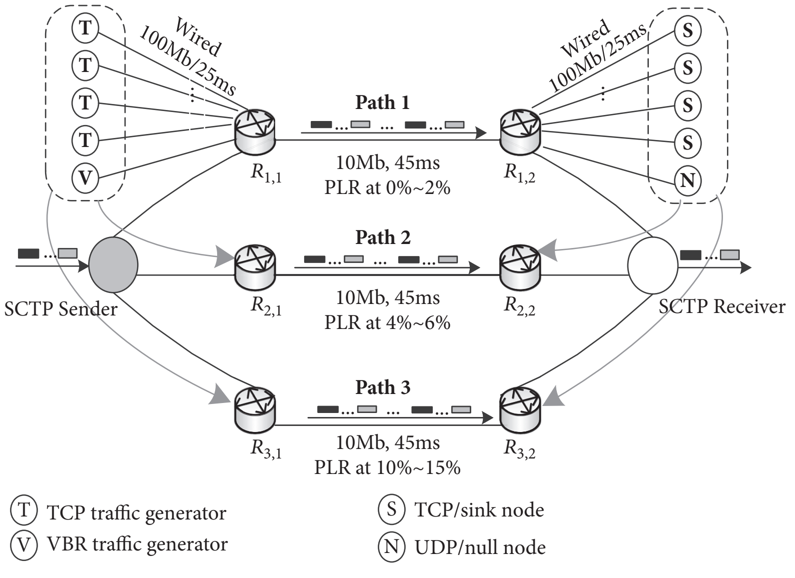


Figure 3: Simulation topology.

(5) Select the next path in P_{list} as the P_{send}, if the path cwnd is full.

The CMT-LA data distribution algorithm greatly improves data delivery throughput and realizes load balancing as well. A detailed description of the pseudocode for the CMT-LA data distribution method is given in Algorithm 2.

4. Simulations and Analysis

4.1. Simulation Steps. To evaluate CMT-LA, all simulation experiments were performed in Network Simulator version 3 (NS-3) [29]. In this paper the topology of the simulation experiment is mainly composed of four parts, namely, SCTP sender, receiver, terminals, and multiple transmission paths, as in Figure 3, but it just takes three transmission paths (Paths 1, 2, and 3) as examples to connect two terminals, respectively. The PLR of Path 1 took a varied value from 0% to 2%, the PLR of Path 2 changed from 4% to 6%, and the PLR of Path 3 ranged from 10% to 15%. We used 64KB as the default receive buffer (rbuf). As for the other SCTP parameters, we merely used the default value supplied by NS-3. In addition, we set 120 seconds as the total simulation time.

For the purpose of reflecting the advantages of CMT-LA, we injected background traffic to simulate a complex Internet environment. We would add four TCP generators to each router when a UDP generator is added in it, so about one-fifth of the total traffic generated by each path is UDP traffic, and four-fifths is TCP. All traffic generators (FTP/TCP and VBR/UDP) connected to routers ($R_{1,1}, R_{2,1}$ and $R_{3,1}$) were configured with 100Mb bandwidth and 25 ms propagation delay. For the sake of reasonable consumption of bandwidth, we set 1 Mbps as the transmission rate for VBR traffic, whereas the FTP used the system value supplied by NS-3.

```
Definition:
$P_i$: the $i^{th}$ path within the SCTP association.
$PLR_{P_i}$: the estimated packet loss rate value of path $P_i$
set $PLR_{P_j} = PLR_{P_{list(0)}}$;
for ($i = 1; i <= count(P_{list}); i++$) do
    sort $P_i$ in an ascending order
    if ($PLR_{P_j} > PLR_{P_{list(i)}}$) then
        set $j = i$
        set $PLR_{P_j} = PLR_{P_{list(i)}}$;
    end if
end for
if !(($P_k \in P_{list}$) && ($PLR_{P_k} = PLR_{P_j}$) &&
($k \neq j$)) then
    set $P_{send} = P_j$;
else if ($\Delta PLR_{P_k} > \Delta PLR_{P_j}$)
    $P_{send} = P_k$
end if
while cwnd of $P_{send}$ is full do
    set $P_{send} = P_{send} \longrightarrow$ next;
end while
```

Algorithm 2

4.2. Simulation Results

4.2.1. Packet Sending and Receiving Times. Figure 4 depicts the sending and receiving times of several packets when taking advantage of classic CMT and CMT-LA, respectively. The results for time t from 0 to 120s are to make a better comparison. We can see that CMT-LA tends to send and receive TSNs better than classic CMT. The reason is that the loss difference among multiple paths is ignored when CMT delivers data within SCTP association. Such a loss-blind data scheduler leads the path to lose a larger volume of data chunks

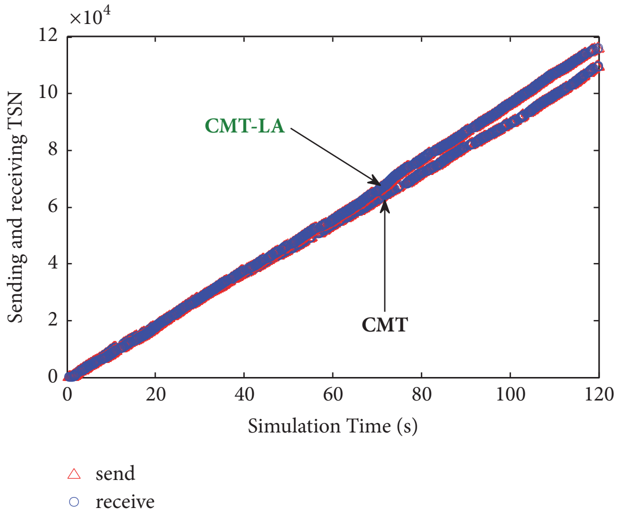


FIGURE 4: Comparison results in terms of sending and receiving time of packets (rbuf=64KB).

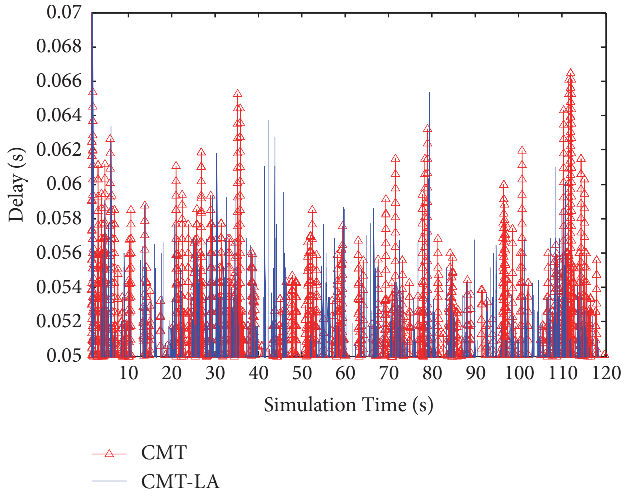


FIGURE 6: Comparison of end-to-end delay (rbuf=64KB).

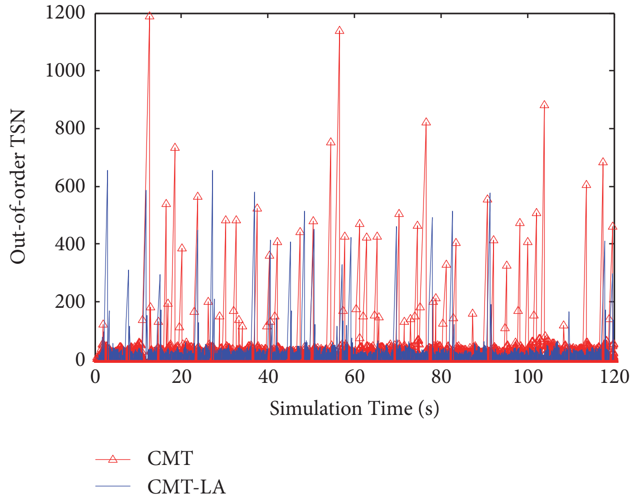


FIGURE 5: Comparison of disorder TSN (rbuf=64KB).

and possibly even fail. Failure to complete the data chunk may hinder the new data transfer work of the SCTP sender.

Conversely, in the SCTP association, CMT-LA can (a) accurately and timely recognize the loss conditions for each path, supported by its pass loss detection module, and (b) split SCTP packets over the paths according to their measured loss condition. Thereby, CMT-LA outperformed classic CMT.

4.2.2. Out-of-Order Packets. The disorder TSN metric is able to convey packet transport traits of CMT-based through multiple connection heterogeneous networks. Before using a disorder TSN metric, you must first acquire the currently and recently received data chunks of TSN. Next, we can calculate the difference using two data chunks received consecutively to get our desired metric. In our experiment, we applied the disorder TSN metric to compare the pros and cons of classic CMT and CMT-LA. Figure 5 presents the comparative results in terms of out-of-order TSNs for each method. The results of time t changing from 0 to 120s are presented to create a better comparison. As Figure 5 shows, classic CMT tended to produce more disorder data chunks and sorting again in increasing numbers when compared with CMT-LA. Additionally, it can also be observed that peak of TSN value at the SCTP receiver was about 1,200 when adopting classic CMT, but less than 700 when utilizing the CMT-LA method.

4.2.3. End-to-End Delay. Figure 6 indicates the performance comparison of classic CMT and CMT-LA when it comes to the end-to-end (e2e) delay. The simulations demonstrated that CMT-LA could lower loss rate, as well as ameliorating the e2e delay when packets were assigned to a different path. This finding results from CMT-LA being equipped with a loss-aware data scheduling strategy that fully takes into account the loss differences on all available paths. In comparison with classic CMT, CMT-LA obtained better e2e delay performance. In addition, the results showed that the average e2e delay of CMT-LA and classic CMT was about 0.048 seconds and 0.051 seconds, respectively. Therefore, CMT-LA attained an advantage of 5.88% over classic CMT with respect to average e2e delay.

4.2.4. Average Throughput. We considered that the rbuf size varied between 32KB, 64KB, and 128KB in order to compare average throughput in the process of sending data chunks. The comparative results are shown in Figures 7, 8, and 9, respectively. Because of its path loss detection module and loss-aware packet scheduler, CMT-LA was able to assign as many SCTP packets as possible over the paths with low loss rates. As can be seen in Figures 7, 8, and 9, the average throughput increased as the rbuf size increased. Specifically, when 32KB, 64KB, and 128KB were used as the rbuf sizes, the CMT-LA throughput was, respectively, 14.0%, 28.4%, and 13.8% higher than the throughput of classic CMT. These results showed that CMT-LA had the ability to decrease the possibility of packet loss while enhancing the goodput.

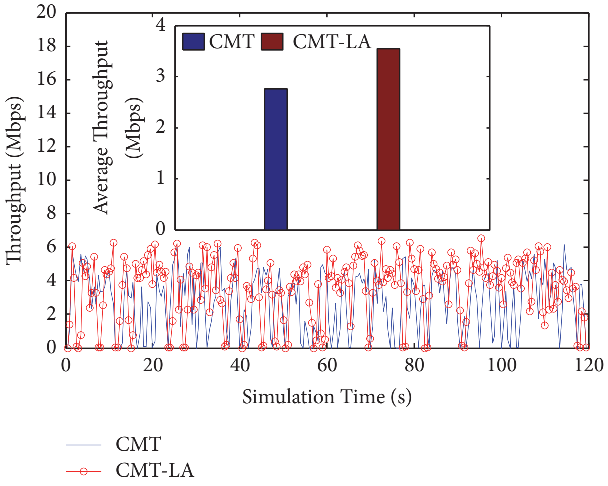


Figure 7: Comparison of average throughput (rbuf=64KB).

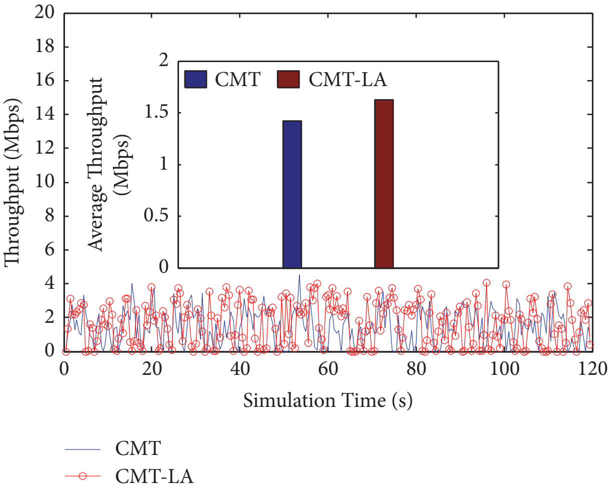


Figure 8: Comparison of average throughput (rbuf=32KB).

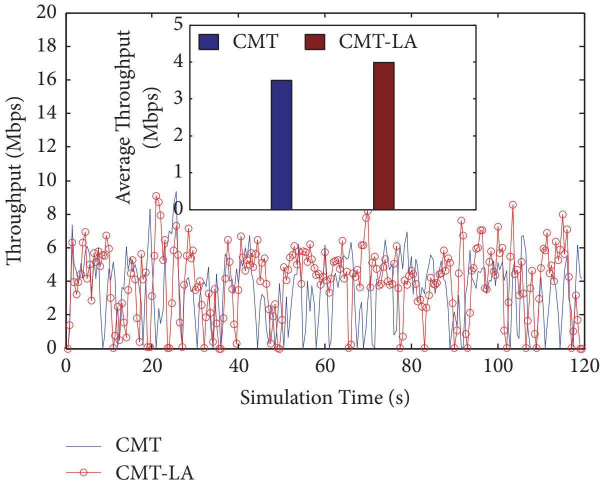


Figure 9: Comparison of average throughput (rbuf=128KB).

5. Conclusion and Future Work

This paper presents CMT-LA, a new loss-aware packet scheduling solution for CMT. CMT-LA makes full use of paths packet loss and the loss variation in order to distribute data packets across the multiple available paths adaptively. Experiments in the simulator show that CMT-LA provides better performance than classic CMT, including improved average throughput, reduced e2e latency, and lowered the amount of disorder data reception.

CMT-LA provides a novel data distribution model to improve the efficiency of data delivery and reduce data packet loss differences by utilizing the path-based loss-aware method. However, the complexity of deploying CMT-LA in the transport networks has led us to further research. Therefore, our future work will concentrate on deploying CMT-LA in real-world systems and satisfying the high-bandwidth requirements and latency zero tolerance for multimedia applications.

Data Availability

The research data can be got from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no. 61562044 and by the Natural Science Foundation of Jiangxi Province under grant nos. 20171BAB212014 and 20161BAB212046.

References

[1] J. Moysen and L. Giupponi, “From 4G to 5G: Self-organized network management meets machine learning,” *Computer Communications*, vol. 129, pp. 248–268, 2018.

[2] J. Eklund, K. Grinnemo, and A. Brunstrom, “Using multiple paths in SCTP to reduce latency for signaling traffic,” *Computer Communications*, vol. 129, pp. 184–196, 2018.

[3] Y. Cao, F. Song, G. Luo et al., “(PU)2M2: a potentially underperforming-aware path usage management mechanism for secure MPTCP-based multipathing services,” *Concurrency Computation*, vol. 30, no. 3, pp. 1–11, 2018.

[4] M. Li, C.-L. Yeh, and S.-Y. Lu, “Real-time QoE monitoring system for video streaming services with adaptive media playout,” *International Journal of Digital Multimedia Broadcasting*, vol. 2018, Article ID 2619438, 11 pages, 2018.

[5] R. Stewart, “Stream control transmission protocol,” IETF RFC 4960 (Proposed Standard), 2007.

[6] Y. Cao, Q. Liu, Y. Zuo, G. Luo, H. Wang, and M. Huang, “Receiver-assisted cellular/wifi handover management for efficient multipath multimedia delivery in heterogeneous wireless networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.

[7] J. Wu, B. Cheng, M. Wang, and J. Chen, “Energy-aware concurrent multipath transfer for real-time video streaming over heterogeneous wireless networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 2007–2023, 2018.

[8] W. Wang, X. Wang, and D. Wang, “Bandwidth scheduling for multipath TCP based concurrent multipath transfer,” in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference, WCNC 2017*, pp. 1–6, San Francisco, CA, USA, March 2017.

[9] Q. Liu, Y. Cao, G. Luo, F. Ke, and Z. Liu, “Rtx+: a novel fast recovery strategy for efficient multipath data delivery over lossy links,” *Journal of Computational Information Systems*, vol. 11, no. 15, pp. 5461–5467, 2015.

[10] J. Liu, X. Bai, and X. Wang, “The strategy for transmission path selection in concurrent multipath transfer,” *Journal of Electronics & Information Technology*, vol. 34, no. 6, pp. 1521–1524, 2012.

[11] J. Li, G. Wei, D. Ding, and Y. Li, “Quantized control for networked switched systems with a more general switching rule,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–9, 2018.

[12] L. P. Verma and M. Kumar, “An adaptive data chunk scheduling for concurrent multipath transfer,” *Computer Standards & Interfaces*, vol. 52, pp. 97–104, 2017.

[13] Y. Cao, Q. Liu, G. Luo, and M. Huang, “Receiver-driven multipath data scheduling strategy for in-order arriving in SCTP-based heterogeneous wireless networks,” in *Proceedings of the 26th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2015*, pp. 1835–1839, China, September 2015.

[14] M. A. Baharudin, Q. T. Minh, and E. Kamioka, “Evaluation of the SCTP optimal path selection with ant colony optimization probabilistic equation implementation,” in *Proceedings of the IEEE 75th Vehicular Technology Conference, VTC Spring 2012*, Tokyo, Japan, June 2012.

[15] T. Dreibholz, E. P. Rathgeb, I. Rüngeler, R. Seggelmann, M. Tüxen, and R. R. Stewart, “Stream control transmission protocol: Past, current, and future standardization activities,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 82–88, 2011.

[16] J. R. Iyengar, P. D. Amer, and R. R. Stewart, “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, 2006.

[17] W. Yang, H. Li, F. Li, Q. Wu, and J. Wu, “RPS: Range-based Path Selection Method for Concurrent Multipath Transfer,” in *Proceedings of the the 6th International Wireless Communications and Mobile Computing Conference*, pp. 944–948, Caen, France, June 2010.

[18] C. Xu, P. Zhang, S. Jia, M. Wang, and G.-M. Muntean, “Video streaming in content-centric mobile networks: challenges and solutions,” *IEEE Wireless Communications Magazine*, vol. 24, no. 5, pp. 157–165, 2017.

[19] J. Baek, P. S. Fisher, M. Jo, and H.-H. Chen, “A lightweight sctp for partially reliable overlay video multicast service for mobile terminals,” *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 754–766, 2010.

[20] C. Huang, Y. Chen, and S. Lin, “Packet scheduling and congestion control schemes for multipath datagram congestion control protocol,” *The Computer Journal*, vol. 58, no. 2, pp. 188–203, 2015.

[21] Y. Cao, F. Song, Q. Liu, M. Huang, H. Wang, and I. You, “A LDDoS-Aware energy-efficient multipathing scheme for mobile cloud computing systems,” *IEEE Access*, vol. 5, pp. 21862–21872, 2017.

[22] W. Gong, X. Yang, M. Zhang, and K. Long, “An adaptive traffic distribution scheme for CMT based on Lotka-Volterra model in multihomed networks,” *China Communications*, vol. 14, no. 2, pp. 79–89, 2017.

[23] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, “Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, 2015.

[24] S. Khan and M. A. Qadir, “Deterministic time markov chain modelling of simultaneous multipath transmission schemes,” *IEEE Access*, vol. 5, pp. 8536–8544, 2017.

[25] J. Wu, B. Cheng, and M. Wang, “Improving multipath video transmission with raptor codes in heterogeneous wireless networks,” *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 457–472, 2018.

[26] C. Xu, Z. Li, L. Zhong, H. Zhang, and G.-M. Muntean, “CMT-NC: improving the concurrent multipath transfer performance using network coding in wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1735–1751, 2015.

[27] Y. Cao, Q. Liu, Y. Zuo, and M. Huang, “Receiver-driven cooperation-based concurrent multipath transfer over heterogeneous wireless networks,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 7, pp. 2354–2370, 2015.

[28] L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: New techniques for congestion detection and avoidance,” *Computer Communication Review*, vol. 24, no. 4, pp. 24–35, 1994.

[29] MPTCP-NS3 Project, Jul. 2017, Available: http://code.google.com/p/mptcp-ns3.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2019, Article ID 1326831, 10 pages
https://doi.org/10.1155/2019/1326831

Research Article

QoS Prediction for Neighbor Selection via Deep Transfer Collaborative Filtering in Video Streaming P2P Networks

Wenming Ma,[1] **Qian Zhang**,[1] **Chunxiao Mu**,[1] **and Meng Zhang**[2]

[1] *School of Computer and Control Engineering, Yantai University, Yantai 264005, China*
[2] *China National Nuclear Corporation, Beijing 100045, China*

Correspondence should be addressed to Wenming Ma; mwmytu@126.com

Received 21 September 2018; Accepted 25 October 2018; Published 1 January 2019

Guest Editor: Yuanlong Cao

To expand the server capacity and reduce the bandwidth, P2P technologies are widely used in video streaming systems in recent years. Each client in the P2P streaming network should select a group of neighbors by evaluating the QoS of the other nodes. Unfortunately, the size of video streaming P2P network is usually very large, and evaluating the QoS of all the other nodes is resource-consuming. An attractive way is that we can predict the QoS of a node by taking advantage of the past usage experiences of a small number of the other clients who have evaluated this node. Therefore, collaborative filtering (CF) methods could be used for QoS evaluation to select neighbors. However, we might use different QoS properties for different video streaming policies. If a new video steaming policy needs to evaluate a new QoS property, but the historical experiences include very few evaluation data for this QoS property, CF methods would incur severe overfitting issues, and the clients then might get unsatisfied recommendation results. In this paper, we proposed a novel neural collaborative filtering method based on transfer learning, which can evaluate the QoS with few historical data by evaluating the other different QoS properties with rich historical data. We conduct our experiments on a large real-world dataset, the QoS values of which are obtained from 339 clients evaluating on the other 5825 clients. The comprehensive experimental studies show that our approach offers higher prediction accuracy than the traditional collaborative filtering approaches.

1. Introduction

In recent years, video content accounts for a large proportion of global Internet consumption. Video steaming is gradually becoming the most attractive service [1–3]. However, Internet does not provide any quality of service guarantees to video content delivery. To expand the server capacity and reduce the video streaming bandwidth, P2P technologies are widely adopted by many content delivery systems [4–7]. In a P2P network, a peer not only downloads the media data from the network but also uploads the download data to other clients in the same network. To get a better user experience of watching videos, each client (or node) in the P2P network should select some other nodes as its neighbors in terms of the quality of service (QoS) for this client [8–10]. For example, a client might prefer to select nodes with high bandwidth. Due to the different locations and network conditions, different clients might have different QoS experience for the same node. To get the neighbors with the best QoS, one might want to evaluate the QoS of all the other nodes for each client. Unfortunately, the video streaming P2P network usually includes an extremely large number of users, and evaluating the QoS of all the nodes is time-consuming and resource-consuming.

An attractive way is that we can predict the QoS value of a node by taking advantage of the past usage experiences of a small number of the other clients who have evaluated this node. This refers to a famous technology, collaborative filtering (CF), which has been extremely studied in recommender systems [11–13]. With the help of CF method, each client only needs to know a small number of the real QoS values of the other nodes to select neighbors. The core idea is that if two clients have similar evaluation values of a specific QoS for some known nodes, they might also have similar QoS evaluation values for the other unknown nodes.

However, the neighbor selection policy might need to be changed to improve the quality of video content delivery. If the new policy uses the new QoS property to select neighbors, but the historical user experiences include very few data of this new QoS property, CF methods would incur severe overfitting issues, and then each client might get worse neighbor recommendation list. Transfer learning aims to adapt a model trained in a source domain with rich labeled data for use in a target domain with less labeled data, where the source and target domain are usually related but under different distributions [14–16]. Recently, deep neural networks have yielded remarkable success on many applications, especially on the computer vision, speech recognition, and natural language processing. Deep neural networks are powerful for learning general and transferable features. There are two major transfer learning scenarios, fine-tuning the pretrained network or treating the pretrained network as a fixed feature exactor. Instead of random initialization, we can initialize the network with a pretrained network, or we can freeze the weights of some layers of the network [17–19].

Unlike many supervised transfer learning tasks, we cannot simply fine-tune or freeze the weights of the network. The only information about the nodes in the video streaming P2P network is their identifiers (IDs) and the QoS evaluation historical experience. There is no raw feature for each node, and we need to lean abstract features for the nodes using embedding. Freezing the embedding features seems unreasonable. Furthermore, different QoS properties have different value ranges, and fine-tuning will make the final weights differ greatly from the initial weights pretrained in the source domain. Due to the sparsity of target domain labeled data, fine-tuning too much would incur severe overfitting problem.

In this paper we proposed a novel neural style collaborative filtering method, DTCF (Deep Transfer Collaborative Filtering). We can first train the model using the QoS evaluation data in the source domain and then adapt the model in the target domain with different QoS property. The core idea is that we only use the weights of first several layers to initialize the same layers of the model in the target domain, and randomly initialize the remaining layers. To control the degree of fine-tuning, we integrate the maximum mean discrepancy (MMD) measurement into the loss function [20–22]. The main contributions of our work are as follows:

(i) We propose a novel neural collaborative filtering model for QoS prediction using transfer learning technology.

(ii) We provide a novel interaction layer to represent the relationship between latent embedding factors of the nodes.

(iii) We adopt partial fine-tuning and MMD measurement to train the target domain model to implement domain adapting.

The remainder of this paper is organized as follows: We introduce the related work in Section 2. Section 3 presents the design details of our method. Section 4 describes our experiments and Section 5 concludes this paper.

2. Related Work

Distributed user-generated videos delivery poses a new challenge to large-scale streaming systems. To stream live videos generated by users, many existing video streaming systems rely on a centralized network architecture [23–25]. Even these streaming systems use Content Delivery Network (CDN) for video delivery, such a kind of solution is not cost-effective [26–28]. The unit price of content delivery over the Internet has dramatically decreased in recent years. However, there are higher requirements in terms of resolution, frame rate, or bitrate than before. Therefore, the amount of bandwidth consumed per user has grown at a faster rate. To reduce the bandwidth or the costs and improve the user experience, the P2P architectures can be adopted instead.

Collaborative filtering is a rational QoS prediction technology to select neighbors for each client in the P2P video streaming network [29–31]. To select the best neighbors with high delivery quality for the clients, CF should predict the QoS values between the clients and then select the top k best neighbors in terms of the QoS values. Each client only knows partial information about the QoS values for all the nodes in the network. Memory-based CF methods are some kinds of generalized k-nearest-neighbors (KNN) algorithms [32, 33], which have two types of models: user-based and item-based. Model-based CF methods are more popular, which act like generalized regression or classification algorithms, but they deal with abstract features not concrete or raw features. Among many model-based CF methods, matrix factorization has become the most popular technology to handle such kind of issues [34–40]. Probabilistic Matrix Factorization (PMF) model considers that the QoS values obey a normal Gaussian distribution, and the latent factors should be learned from zero-mean Gaussian distribution [41]. Nonnegative matrix factorization (NMF) can learn the nonnegative latent factors for the users or items, but it usually deals with the implicit feedback [42–44].

However, even if matrix factorization CF algorithms have obtained remarkable success, they have difficulty in dealing with cross-domain learning tasks if the output values of the source and target domain have different ranges. Deep neural networks can easily learn general and transferable features. More and more cross-domain applications adopt deep learning technologies and have yielded remarkable performance [45–47]. However, the exploration of deep neural networks on recommender systems or QoS prediction has received relatively less attraction. Recently, some studies have proposed some deep learning-based collaborative filtering models. Two impressive technologies are Google's Wide & Deep [48] and Microsoft's Deep Crossing [49]. The input of these models is side information, not the interaction between the users and items. Neural Collaborative Filtering (NCF) models are designed purely for user and item interactions [50]. However, none of them are designed for cross-domain QoS prediction.

3. DTCF Model

For the cross-domain QoS prediction in the video streaming P2P network, we are given a *source* domain $\mathscr{D}_s = \{\langle x_i^s, x_j^s, r_{ij}^s \rangle\}_{i \neq j}$ with n_s examples, which is characterized by the probability distribution p and a *target* domain $\mathscr{D}_t = \{\langle x_k^t, x_l^t, r_{kl}^t \rangle\}_{k \neq l}$ with n_t examples which is characterized by the probability q. Usually the size of examples in the target domain is extremely small, $n_s \succ n_t$. Our work aims to build a deep neural network to learn transferable features that bridge these two domains' discrepancy.

3.1. Model Architecture Overview. We propose a novel neural architecture, outlined in Figure 1. The source domain and the target domain share the same network architecture. The input of the model is the identifier number of the nodes. For example, if size of nodes in the P2P network is n, the ID of each node is an integer number from 1 to n. The output of the mode is the QoS value that the node x_i evaluates on the node x_j.

Since we do not use any concrete feature for each node, we need to learn abstract features for them. Here, we use embedding layer to learn a continuous latent vector/factor $\mathbf{u}$ for each node. The details of designing embedding layers are described in Section 3.1.

If we get two latent vectors for x_i and x_j, $\mathbf{u}_i$ and $\mathbf{u}_j$, one might expect that we should concatenate these two vectors and then use affine function $\mathbf{W}\left[\mathbf{u}_i \ \ \mathbf{u}_j\right]^{\mathrm{T}} + \mathbf{b}$ to transform the latent vectors into the input of the other hidden layer above. However, there is no interactive action between the latent factors, but only weighted summation of elements of the vectors. Some studies use the dot product of the vectors to represent the interaction, which is described as follows.

$$\mathbf{u}_i \odot \mathbf{u}_j = \left\langle u_{i,1} \times u_{j,1}, u_{i,2} \times u_{j,2}, \cdots, u_{i,d} \times u_{j,d} \right\rangle \tag{1}$$

Unfortunately, it is too simple to completely represent the complex interaction between nodes. In this paper, we propose a novel interaction layer to tackle this problem, which has powerful representation capacity. We will give the design details in Section 3.2.

Above the interaction layer, we use ReLU as the hidden layer. We might need multiple ReLU layers. The ReLU activation function is as follows.

$$\begin{aligned} \mathbf{z}_o &= \mathbf{W}\mathbf{a}_i + \mathbf{b} \\ \mathbf{a}_o &= (\mathbf{0}, \mathbf{z}_o) \end{aligned} \tag{2}$$

Finally, we use a fully connected layer to generate the output. When training the model in the source domain, we use the regression loss. We then use the all the layers of the pretrained model but the last FC layer to construct the model for target domain. The weights of these layers are kept as the initialized weights of the target domain model, but the final FC layer is initialized randomly. To avoid the overadaptation problem, we use both the domain loss and regression loss to train the target domain model. We will describe how to design the domain loss in Section 3.3.

3.2. Embedding Layer. Since we can assign a unique integer number as the identifier for each node in the network, we can use a one-hot vector to represent the identifier. If we have at most n nodes in the network, the ith nodes can be expressed as follows.

$$\begin{aligned} \mathbf{v}_i &= \left[v_{i,1}, v_{i,2}, \ldots, v_{i,n}\right] \\ v_{i,r} &= \begin{cases} 1, & if\ r = i,\ 1 \leq r \leq n \\ 0, & if\ r \neq i,\ r \leq l \leq n \end{cases} \end{aligned} \tag{3}$$

Our embedding layer is defined as follows:

$$\mathbf{u}_i = \mathbf{W}\mathbf{v}_i \tag{4}$$

where $\mathbf{W}$ is a $d \times n$ matrix. Expanding the formulas $\mathbf{W}\mathbf{v}_i$, we can see the following.

$$\mathbf{W}\mathbf{v}_i = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ldots & \vdots \\ w_{k1} & w_{k2} & \cdots & w_{km} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{ii} \\ \vdots \\ w_{ki} \end{bmatrix} \tag{5}$$

Therefore, $\mathbf{u}_i$ is the ith column of matrix $\mathbf{W}$. Since the node identifier number is transformed to a one-hot vector, the result of matrix multiplication is exactly a specific latent vector for each node. This weight matrix is jointly trained with the other parameters of the whole network.

3.3. Interaction Layer. There are two inputs of the interaction layer, $\mathbf{u}_i$ and $\mathbf{u}_j$. Suppose any single vector is a column vector, and concatenating the two inputs will get a longer vector. This concatenation vector will be transformed to another vector, encoding interactive information between these two inputs. The transformation process is outlined in Figure 2.

Suppose the output of interaction layer is a vector $\mathbf{h}$, the length of which is k. The sth element of the vector $\mathbf{h}$ is defined as follows.

$$h_c = \left[\mathbf{u}_i^{\mathrm{T}} \ \ \mathbf{u}_j^{\mathrm{T}}\right] \mathbf{W}_s \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{bmatrix} + b_s \tag{6}$$

If the length of $\mathbf{u}_i$ is d, $\mathbf{W}_s$ is a $2d \times 2d$ square matrix. h_c is a scalar, the value of which is determined by the matrix $\mathbf{W}_s$ and the bias b_s. If the length of $\mathbf{h}$ is k, we need k weight matrices and biases.

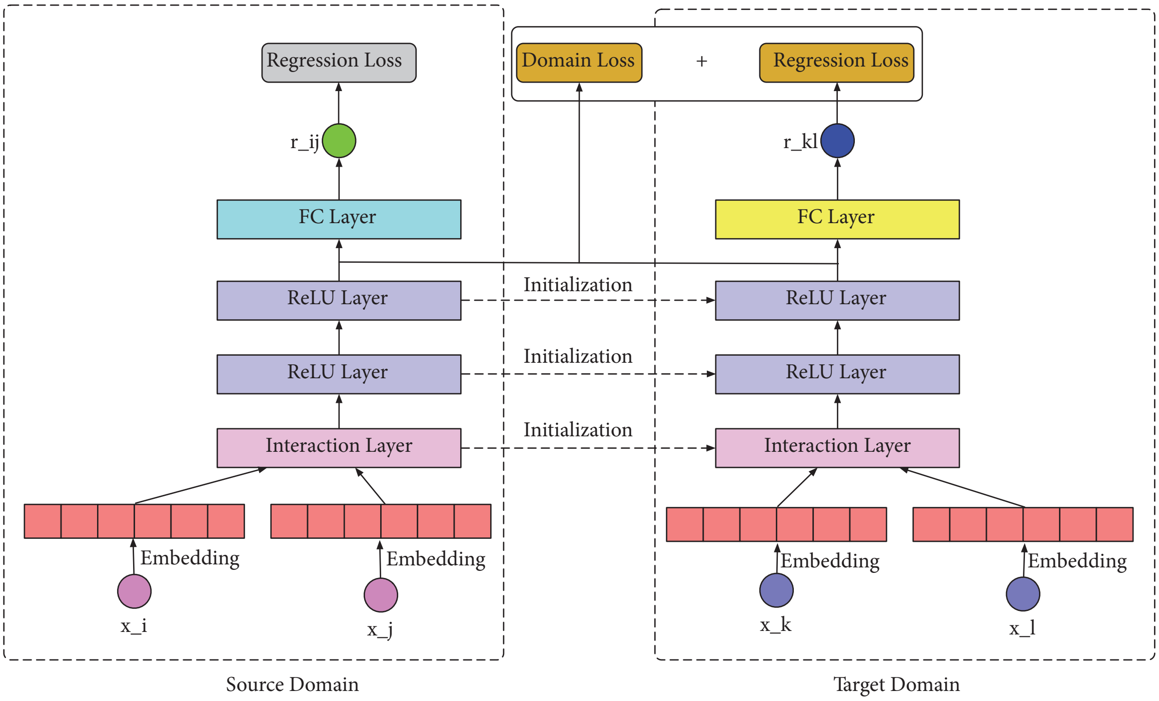


Figure 1: DTCF architecture.

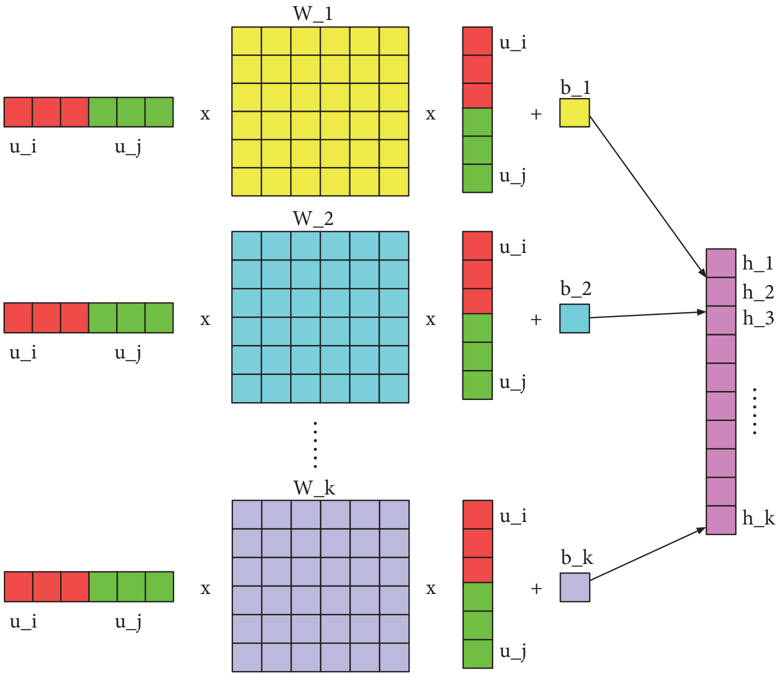


Figure 2: Interaction layer.

h_c include all the possible interaction relationships between $\mathbf{u}_i$ and $\mathbf{u}_j$. Denote $\mathbb{U}^{\mathrm{T}} = \left[\mathbf{u}_i^{\mathrm{T}}\ \mathbf{u}_j^{\mathrm{T}}\right] = [\mathbb{U}_1, \mathbb{U}_2, \ldots, \mathbb{U}_{2k}]$, and we can see that $h_c = \sum w_s^{ij} \mathbb{U}_i \mathbb{U}_j + b_s$, where w_s^{ij} is the element at the ith row and the jth column in the matrix $\mathbf{W}_s$

3.4. Domain Loss. The output of the last ReLU layer of the model in the source domain is denoted as $\mathbf{h}^s$, and the output of the last ReLU layer of the model in the target domain is denoted as $\mathbf{h}^t$. If we want to avoid the overadaptation problem, one possible way is to minimize the differences between the distributions $p = \Pr(x_i^s, x_j^s, \mathbf{h}^s(x_i^s, x_j^s))$ and $q = \Pr(x_i^t, x_j^t, \mathbf{h}^t(x_i^t, x_j^t))$, where $x_i^s = x_i^t$ and $x_j^s = x_j^t$.

Let $(\mathscr{X}, D)$ be a metric space, and $\langle x_i^s, x_j^s, \mathbf{h}^s(x_i^s, x_j^s)\rangle \in \mathscr{X}$, $\langle x_i^t, x_j^t, \mathbf{h}^t(x_i^t, x_j^t)\rangle \in \mathscr{X}$. Let $\mathscr{F}$ be a class of functions $f : \mathscr{X} \longrightarrow \mathbb{R}$, and the Maximum Mean Discrepancy (MMD) is as [22]

$$\mathrm{MMD}[\mathscr{F}, p, q] = \sup_{f\in\mathscr{F}} \left(\mathbb{E}_{\mathbf{s}\sim p}[f(\mathbf{s})] - \mathbb{E}_{\mathbf{t}\sim q}[f(\mathbf{t})]\right) \tag{7}$$

where $\mathbf{s} = \langle x_i^s, x_j^s, \mathbf{h}^s(x_i^s, x_j^s)\rangle \in \mathbf{S}$ and $\mathbf{t} = \langle x_i^t, x_j^t, \mathbf{h}^t(x_i^t, x_j^t)\rangle \in \mathbf{T}$.

Denote $\langle x_i^s, x_j^s\rangle \in \mathbf{S}_x$ and $\langle x_i^t, x_j^t\rangle \in \mathbf{T}_x$. The biased empirical estimated of the MMD is defined as follows.

$$\begin{aligned}&\mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}] \\ &\quad = \sup_{f\in\mathscr{F}} \frac{1}{|\mathbf{S}_x \cap \mathbf{T}_x|} \sum_{\mathbf{s}_i, \mathbf{t}_i \in \mathbf{S}_x \cap \mathbf{T}_x} ([f(\mathbf{s}_i)] - [f(\mathbf{t}_i)])\end{aligned} \tag{8}$$

If the function class $\mathscr{F}$ is too large, it is not practical to work with this rich function class in the finite sample setting. A rational choice of the function class is a universal reproducing kernel Hilbert space $\mathscr{H}$, named *universal RKHS*. Therefore, we have that $f(x) = \langle f, \phi(x)\rangle_{\mathscr{H}}$, where $f \in \mathscr{H}$. The kernel function $k(x, y)$ is equal to $\langle \phi(x), \phi(y)\rangle$.

Denote $f = \phi(a) = k(a, \cdot)$, and we can get the mean embedding of the distribution p; that is, $\mu_p(a) = \langle \mu_p, k(a, \cdot)\rangle = \mathbb{E}_x k(a, x)$. From [22], we can obtain the following.

$$\begin{aligned}&\mathrm{MMD}[\mathscr{F}, p, q]^2 \\ &\quad = \left[\sup_{\|f\|_{\mathscr{H}}\le 1} \left(\mathbb{E}_{\mathbf{s}\sim p}[f(\mathbf{s})] - \mathbb{E}_{\mathbf{t}\sim q}[f(\mathbf{t})]\right)\right]^2 \\ &\quad = \left[\sup_{\|f\|_{\mathscr{H}}\le 1} \left\langle \mu_p - \mu_q, f\right\rangle_{\mathscr{H}}\right]^2 = \left\|\mu_p - \mu_q\right\|_{\mathscr{H}}^2 \\ &\quad = \mathbb{E}_{\mathbf{s},\mathbf{s}'}\left[k\left(\mathbf{s}, \mathbf{s}'\right)\right] - 2\mathbb{E}_{\mathbf{s},\mathbf{t}}[k(\mathbf{s}, \mathbf{t})] + \mathbb{E}_{\mathbf{t},\mathbf{t}'}\left[k\left(\mathbf{t}, \mathbf{t}'\right)\right]\end{aligned} \tag{9}$$

Similarly, the empirical estimate can be defined now as follows.

$$\begin{aligned}\mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]^2 &= \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j\ne i}^{L} k\left(\mathbf{s}_i, \mathbf{s}_j\right) \\ &\quad + \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j\ne i}^{L} k\left(\mathbf{t}_i, \mathbf{t}_j\right) \\ &\quad - \frac{2}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L} k\left(\mathbf{s}_i, \mathbf{t}_j\right)\end{aligned} \tag{10}$$

In this paper, we use the empirical estimate of MMD^2 as the domain loss. What we need to do is to select suitable universal kernel function. Here, we adopt Gaussian kernel function, which is defined as follows.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\delta^2}\right) \tag{11}$$

3.5. Algorithm. The total loss of target domain includes regression loss and MMD loss. We use the minibatch to train the model. Only a small group of examples are used to compute the loss per training iteration. Denote the set of the minibatch examples in the source domain $\mathscr{M}_s$ and the set of the minibatch examples in the target domain $\mathscr{M}_t$. The loss function of the model in the source domain is defined as follows.

$$\mathscr{L}(\mathscr{M}_s) = \frac{1}{|\mathscr{M}_s|}\sum\left(\hat{r}_{ij} - r_{ij}\right)^2 \tag{12}$$

However, the loss function of the model in the target domain is defined as

$$\mathscr{L}(\mathscr{M}_t) = \frac{1}{|\mathscr{M}_t|}\sum\left(\hat{r}_{ij} - r_{ij}\right)^2 + \mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]^2 \tag{13}$$

where $\mathbf{S}_x \cap \mathbf{T}_x = \mathscr{M}_t$.

To optimize our model, we need to compute the gradient of each weight. For any weigh related to both of the regression and domain loss, its gradient is computed as follows.

$$\begin{aligned}&\frac{\partial \mathscr{L}(\mathscr{M}_t)}{\partial w} \\ &\quad = \frac{1}{|\mathscr{M}_t|}\sum\frac{\partial\left(\hat{r}_{ij} - r_{ij}\right)^2}{\partial w} + \frac{\partial \mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]^2}{\partial w} \\ &\quad = \frac{2}{|\mathscr{M}_t|}\sum\frac{\left(\hat{r}_{ij} - r_{ij}\right)\partial\hat{r}_{ij}}{\partial w} \\ &\qquad + 2\mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]\frac{\partial \mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]}{\partial w}\end{aligned} \tag{14}$$

$$\begin{aligned}&\frac{\partial \mathrm{MMD}[\mathscr{F}, \mathbf{S}, \mathbf{T}]}{\partial w} \\ &\quad = -\frac{2}{|\mathscr{M}_t|}\sum_{i\ne j}\exp\left(-\frac{\left\|\mathbf{h}_i^t - \mathbf{h}_j^t\right\|^2}{2\delta^2}\right)\left\|\mathbf{h}_i^t - \mathbf{h}_j^s\right\|\frac{\partial\mathbf{h}_i^t}{\partial w} \\ &\qquad - \frac{2}{|\mathscr{M}_t|}\sum_{i\ne j}\exp\left(-\frac{\left\|\mathbf{h}_i^t - \mathbf{h}_j^t\right\|^2}{2\delta^2}\right)\left\|\mathbf{h}_i^t - \mathbf{h}_j^s\right\|\frac{\partial\mathbf{h}_j^t}{\partial w} \\ &\qquad + \frac{4}{|\mathscr{M}_t|}\sum\exp\left(-\frac{\left\|\mathbf{h}_i^s - \mathbf{h}_j^t\right\|^2}{2\delta^2}\right)\left\|\mathbf{h}_i^s - \mathbf{h}_j^t\right\|\frac{\partial\mathbf{h}_j^t}{\partial w}\end{aligned} \tag{15}$$

Note that $\mathbf{h}_i^s$ is not used for computing gradients, because we only train the target domain model after pretraining in the source domain. The training process is described as follows:

(i) We first train the model of the source domain using the loss function $\mathscr{L}(\mathscr{M}_s)$. The gradient of each weigh is computed according to formula (13).

(ii) After training, we use the weights of this model to initialize the model in the target domain except the weights of the last FC layer. The last FC layer of the model of the target domain is initialized randomly.

(iii) While training the model of the target domain, we use the loss function $\mathscr{L}(\mathscr{M}_t)$.

(iv) For each training iteration, we randomly select examples in the dataset, and compute the gradient according to formulas (14) and (15).

(v) We use ADAM (Adaptive Moment Estimation) as the optimizer.

4. Experimental Results

4.1. Dataset and Evaluation Metrics. We conduct our experiments on a publicly large accessible dataset, WS-DREAM dataset#1, obtained from 339 hosts doing QoS evaluation on the other 5825 hosts. There are two types of QoS properties in this dataset: response time and throughput. Here, we use the response time as the source domain, and the throughput as the target domain.

For the source domain, we randomly extract 30% (density) of the data as the source training set. For the target domain, we construct 5 different training sets with different density of 0.5%, 1%, 1.5%, 2%, 2.5%, and 3%. Consequently, the remaining data is the test set.

We adopt a common evaluation metric: Mean Absolute Error (MAE), which is widely employed to measure the QoS prediction quality.

$$MAE = \frac{\sum_{(i,j,r_{i,j})\in \mathbf{Q}_E} \left| r_{i,j} - \hat{r}_{i,j} \right|}{\left| \mathbf{Q}_E \right|} \tag{16}$$

4.2. Performance Comparison. We compare our methods with some traditional collaborative filtering methods: UPCC, IPCC, UIPCC [34], and matrix factorization (MF). UPCC is a user-based CF method, which uses PCC (Pearson Correlation Coefficient) to calculate the similarity between users. IPCC is similar to UPCC, except that it calculates the similarity between items. UIPCC combines the advantages of these two methods by balancing the proportions of them in the final results. For UPCC, IPCC, and UIPCC, different tradeoff parameters $k = 5, 10, 15, 20, 25, 30$ (the parameters of top k similar users or services) are tried, and finally we choose $k = 10$. For MF and DTCF, the sizes of latent factors are also set to 10. For DTCF, different hidden ReLU layers and different hidden unit sizes are tried. Here, the maximum number of hidden layers is limited to 5. We tested the batch size of [128, 256, 512, 1024], the learning rate of [0.0001,0.0005, 0.001, 0.005], and the training epoch of [10, 20, 30, 40, 50, 60, 70, 80]. The bandwidth δ is set to the median pairwise distance on the source training data.

We conduct 10 experiments for each model and each sparsity level and then average the prediction accuracy values.

The results are reported in Figures 3 and 4. We can make the following observations:

(i) As the sparsity level increases, the MAEs of all the models decrease.

(ii) Our DTCF methods outperform the other traditional collaborative filtering methods, especially when the training set is extremely sparse.

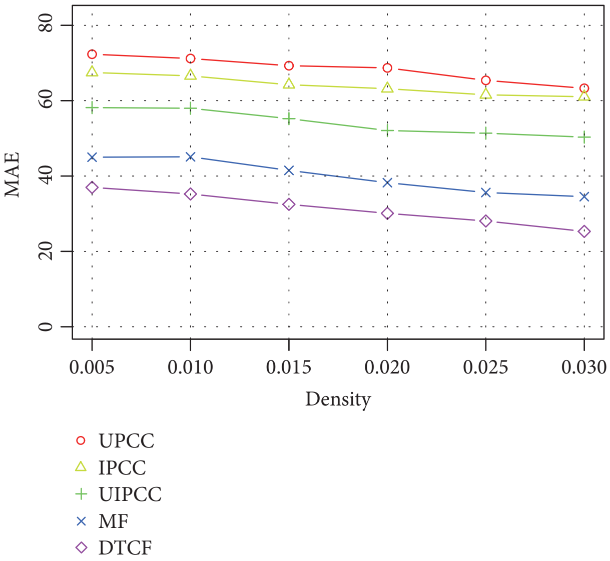


Figure 3: MAE with respect to density.

(iii) DTCF model has more weights that need to be trained than the other models, but it gets the best performance, which indicates that the relationship between nodes is very complex, and shallow models cannot capture these structures.

Although shallow models are not easily overfitting when the target domain training dataset is extremely sparse, they cannot transfer rich information from the source domain. The deep models might easily incur overfitting problem, but they can learn common latent features from the source domain. To balance this dilemma, we need to control the degree of fine-tuning the deep model. This experiment shows that MMD domain loss is an efficient way of controlling the adapting degree.

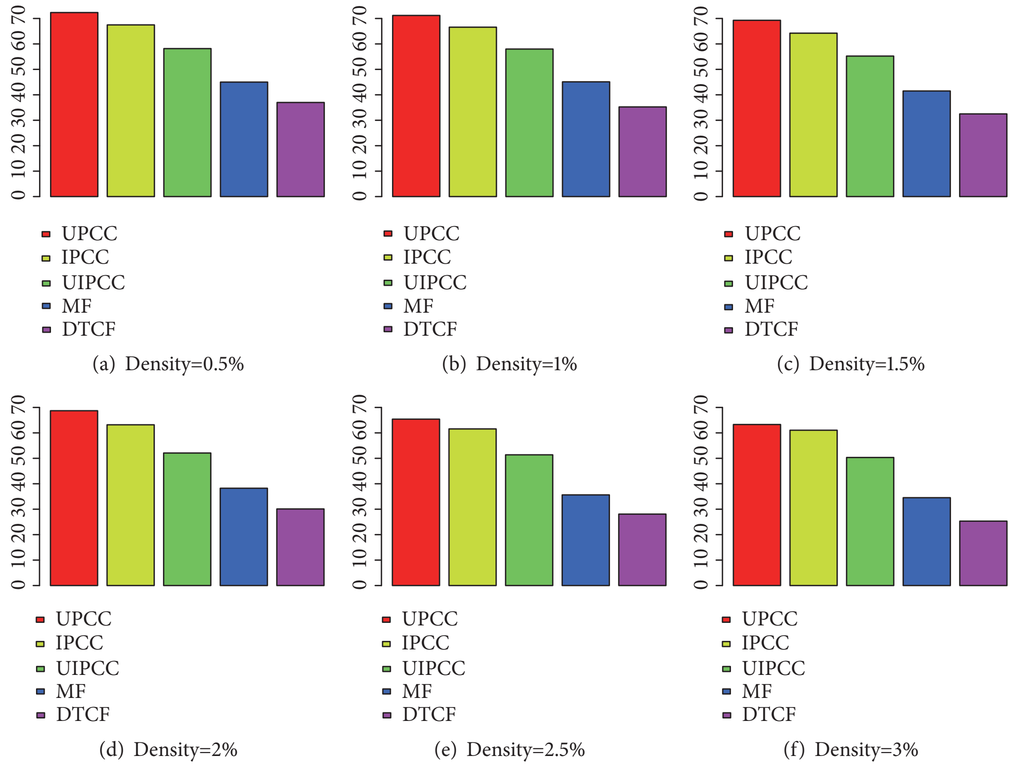
4.3. Impact of the Network Depth. The network depth usually has important impact on the prediction performance. Here, the number of neurons of each ReLU is set to 128, and we add the number of ReLU layers from 1 to 6 to see how the MAE values change. The experimental result is outlined in Figure 5, from which we can see the following:

(i) Adding more ReLU layers can get better prediction performance, but when the depth exceeds a limited value, the performance starts to become worse.

(ii) Although adding more ReLU layers can improve the performance, it seems that enlarging the size of the training data would be more helpful.

(iii) Sometimes, adding more layers would not improve the performance anymore, but it also does not get worse prediction performance. This indicates that deep neural network has some kind of regularization property.

Actually, if the training dataset is very large, adding more layers usually does not incur overfitting problems, but for the



(a) Density=0.5%

(b) Density=1%

(c) Density=1.5%

(d) Density=2%

(e) Density=2.5%

(f) Density=3%

Figure 4: MAE comparison for each density.

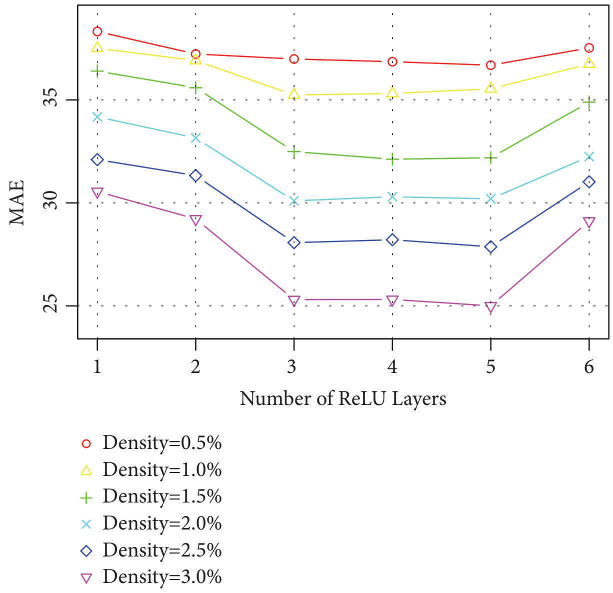


Figure 5: MAE with respect to number of ReLU layers.

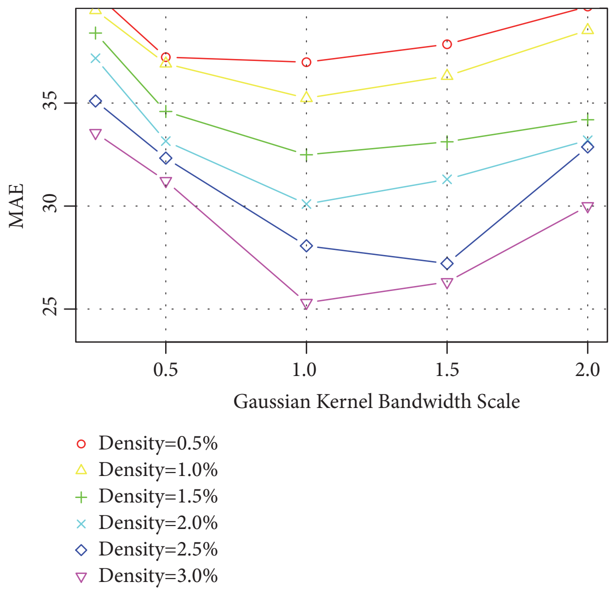


Figure 6: MAE with respect to Gaussian kernel bandwidth scale.

cross-domain learning, the target domain has very little data, so the network depth needs control.

4.4. Impact of the Gaussian Kernel Bandwidth. Another hyperparameter that we need to determine is the Gaussian kernel bandwidth. By default, it is set to the median pairwise distance on the source training data. We scale the default value from 0.25 to 2.0, and the experimental result is outlined in Figure 6.

(i) Obviously, the default value is a rational choice, and scaling too small or too large would get worse prediction performance.

(ii) If the bandwidth is too large, the kernel will be approximately equal to 1, and the nodes would look the same. We cannot propose personal recommendation for them.

(iii) If the bandwidth is too small, the kernel will be approximately equal to 0, and the nodes cannot find similar neighbors to follow their past experiences.

5. Conclusion

Selecting neighbors in terms of the QoS is an effective way of providing high quality contents in video streaming P2P networks. Due to the heterogeneous network conditions, the QoS between any pairs of nodes is different. However, evaluating the QoS of all the nodes for each user is resource-consuming. An attractive way is to adopt collaborative filtering technologies, which use only a small amount of past usage experience.

Unfortunately, the video content providers might often choose different QoS properties to select neighbors. Traditional CF methods cannot solve the cross-domain QoS prediction problem. This paper proposed a novel neural style CF method based on transfer learning. We first outlined our model architecture and then introduced the details of important parts of this model. To avoid the overadaptation problem, we combined domain loss and prediction loss together to train the model of the target domain. We adopted MMD distance as our domain loss, and we also provide its principle and how to compute the gradient. Finally, we conducted our experiments on a real-world public dataset. The experimental results show that our DTCF model can outperform the other models for cross-domain QoS prediction.

Data Availability

The WS-Dream data used to support the finding of this study is owned by a third party, which is an open dataset and is deposited in "https://github.com/wsdream/wsdream-dataset".

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work are supported by the National Nature Science Foundation of China (No. 61602399 and No. 61502410).

References

[1] B. E. Mada, M. Bagaa, and T. Taleb, "Efficient Transcoding and Streaming Mechanism in Multiple Cloud Domains," in *Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM 2017)*, pp. 1–6, Singapore, December 2017.

[2] H. Du, Q. Zheng, W. Zhang, and X. Gao, "A Bandwidth Variation Pattern-Differentiated Rate Adaptation for HTTP Adaptive Streaming over an LTE Cellular Network," *IEEE Access*, vol. 6, pp. 9554–9569, 2017.

[3] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient Video Pricing and Caching in Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8744–8751, 2016.

[4] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A Blockchain based Truthful Incentive Mechanism for Distributed P2P Applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.

[5] S.-H. Lin, R. Pal, B.-C. Wang, and L. Golubchik, "On market-driven hybrid-P2P video streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 984–998, 2017.

[6] G. Huang, Y. Gao, L. Kong, and K. Wu, "An incentive scheme based on bitrate adaptation for cloud-assisted P2P video-on-demand streaming systems," in *Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 404–408, Chengdu, April 2018.

[7] J. Junchen, S. Shijie, V. Sekar, and H. Zhang, "Pytheas: Enabling data-driven quality of experience optimization using group based exploration-exploitation," in *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation*, pp. 393–406, 2017.

[8] L. Wang, D. Zhang, and H. Yang, "Qos-Awareness variable neighbor selection for mesh-based P2P live streaming system," in *Proceedings of the 2013 IEEE 3rd International Conference on Information Science and Technology, ICIST 2013*, pp. 1197–1201, China, March 2013.

[9] A. T. Liem, I.-S. Hwang, A. Nikoukar, C.-Z. Yang, M. S. Ab-Rahman, and C.-H. Lu, "P2P Live-Streaming Application-Aware Architecture for QoS Enhancement in the EPON," *IEEE Systems Journal*, vol. 12, no. 1, pp. 648–658, 2018.

[10] J. Li, Y. Bai, N. Zaman, and V. C. M. Leung, "A Decentralized Trustworthy Context and QoS-Aware Service Discovery Framework for the Internet of Things," *IEEE Access*, vol. 5, pp. 19154–19166, 2017.

[11] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, no. 99, pp. 1–6, 2018.

[12] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Computer Communications*, vol. 41, pp. 1–10, 2014.

[13] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[14] P. Peng, Y. Tian, T. Xiang, Y. Wang, M. Pontil, and T. Huang, "Joint Semantic and Latent Attribute Modelling for Cross-Class Transfer Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1625–1638, 2018.

[15] H. Chang, J. Han, C. Zhong, A. M. Snijders, and J.-H. Mao, "Unsupervised Transfer Learning via Multi-Scale Convolutional Sparse Coding for Biomedical Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1182–1194, 2018.

[16] Y. Guo, G. Ding, J. Han, and Y. Gao, "Zero-shot learning with transferred samples," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3277–3290, 2017.

[17] G. Wang, W. Li, M. A. Zuluaga et al., "Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[18] X. An, X. Zhou, X. Lü, F. Lin, and L. Yang, "Sample Selected Extreme Learning Machine Based Intrusion Detection in Fog Computing and MEC," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7472095, 10 pages, 2018.

[19] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *Proceedings of the 2015 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2015*, pp. 1–6, Turin, Italy, July 2015.

[20] M. J. Gangeh, H. Tadayyon, L. Sannachi, A. Sadeghi-Naini, W. T. Tran, and G. J. Czarnota, "Computer Aided Theragnosis Using Quantitative Ultrasound Spectroscopy and Maximum Mean Discrepancy in Locally Advanced Breast Cancer," *IEEE Transactions on Medical Imaging*, vol. 35, no. 3, pp. 778–790, 2016.

[21] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 945–954, USA, July 2017.

[22] A. Gretton, K. M. Borgwardt, M. J. Rasch et al., "A kernel two-sample test," *Journal of Machine Learning Research (JMLR)*, vol. 13, pp. 723–773, 2012.

[23] Online. 2018. http://www.twitch.tv/.

[24] Online. 2018. http://www.ustream.tv.

[25] Online. 2018. http://www.livestream.com/.

[26] A. O. Al-Abbasi and V. Aggarwal, "EdgeCache: An optimized algorithm for CDN-based over-the-top video streaming services," in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 202–207, Honolulu, HI, USA, April 2018.

[27] R. Denis, S. Matias, R. Juergen et al., "Service migration from cloud to multi-tier fog nodes for multimedia dissemination with QoE support," *Sensors*, vol. 18, no. 2, 2018.

[28] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, and A. Ashkan, "Cache content-selection policies for streaming video services," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, USA, April 2016.

[29] X. Wu, B. Cheng, and J. Chen, "Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method," *IEEE Transactions on Services Computing*, vol. 10, no. 3, pp. 352–365, 2017.

[30] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Transactions on Services Computing*, vol. 10, no. 3, pp. 686–699, 2016.

[31] D. Margaris, C. Vassilakis, and P. Georgiadis, "An integrated framework for adapting WS-BPEL scenario execution using QoS and collaborative filtering techniques," *Science of Computer Programming*, vol. 98, pp. 707–734, 2015.

[32] A. Bellogín, P. Castells, and I. Cantador, "Neighbor Selection and Weighting in User-Based Collaborative Filtering: A Performance Prediction Approach," *ACM Transactions on the Web (TWEB)*, vol. 8, no. 2, pp. 1–30, 2014.

[33] Z. Jia, Y. Yang, W. Gao, and X. Chen, "User-based collaborative filtering for tourist attraction recommendations," in *Proceedings of the IEEE International Conference on Computational Intelligence and Communication Technology (CICT '15)*, pp. 22–25, February 2015.

[34] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service Qos prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2013.

[35] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS Prediction for Runtime Service Adaptation via Adaptive Matrix Factorization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2911–2924, 2017.

[36] R. Zhu, D. Niu, and Z. Li, "Robust web service recommendation via quantile matrix factorization," in *Proceedings of the 2017 IEEE Conference on Computer Communications, INFOCOM 2017*, USA, May 2017.

[37] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proceedings of the 30th IEEE International Symposium on Reliable Distributed Systems (SRDS '11)*, pp. 1–10, IEEE, Madrid, Spain, October 2011.

[38] D. Yu, Y. Liu, Y. Xu, and Y. Yin, "Personalized QoS prediction for web services using latent factor models," in *Proceedings of the 11th IEEE International Conference on Services Computing, (SCC '14)*, pp. 107–114, July 2014.

[39] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, New York, NY, USA, August 2008.

[40] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "An Extended Matrix Factorization Approach for QoS Prediction in Service Selection," in *Proceedings of the 2012 9th IEEE International Conference on Services Computing (SCC)*, pp. 162–169, Honolulu, HI, USA, June 2012.

[41] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS'07)*, pp. 1257–1264, 2007.

[42] D. Daniel, H. Lee, and S. Sebastian, "Algorithms for Non-negative Matrix Factorization," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS'00)*, pp. 556–562, 2000.

[43] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[44] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowledge-Based Systems*, vol. 97, pp. 188–202, 2016.

[45] H. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[46] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1717–1724, IEEE, Columbus, OH, USA, 2015.

[47] H. Chen, Q. Dou, D. Ni et al., "Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks," in *Proceedings of the International Conference*

On Medical Image Computing & Computer Assisted Intervention, pp. 507–514, 2015.

[48] H.-T. Cheng, K. Levent, H. Jeremiah et al., "Wide & Deep Learning for Recommender Systems," in *Proceedings of the1st Workshop on Deep Learning for Recommender Systems*, pp. 7–20, 2016.

[49] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. C. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 255–262, USA, August 2016.

[50] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural Collaborative Filtering," in *Proceedings of the 26th International Conference*, pp. 173–182, Perth, Australia, April 2017.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2019, Article ID 5284968, 7 pages
https://doi.org/10.1155/2019/5284968

Research Article

Dynamic Soft Real-Time Scheduling with Preemption Threshold for Streaming Media

Wenle Wang,[1] **Yuan Wang**,[1] **JiangYan Dai**,[2] **and Zhonghua Cao**[3]

[1]*School of Software, Jiangxi Normal University, Nanchang 330022, Jiangxi, China*
[2]*School of Computer Engineering, Weifang University, Weifang 261061, Shandong, China*
[3]*School of Software & Communication Engineering, Jiangxi University of Finance and Economics, Nanchang 330038, Jiangxi, China*

Correspondence should be addressed to JiangYan Dai; daijyan@163.com

Received 21 September 2018; Revised 19 November 2018; Accepted 3 December 2018; Published 1 January 2019

Guest Editor: Kai Wang

Over the last decades, the advancements in networking technology and new multimedia devices have motivated the research on efficient video streaming mechanisms under wireless. We consider combing soft real-time video streaming scheduling with threshold to minimize the ineffective preemption. Based on the value density and urgency of soft real-time task, the dynamic scheduling with preemption threshold strategy (DSPT) is proposed in the paper. By analyzing the response time and preemption relationship of tasks, the preemption thresholds are assigned. Simulation results show that the DSPT strategy achieves improvements about success rate, delay time, and benefit of the system.

1. Introduction

With the advancements in networking technology and new multimedia devices, video streaming mechanisms under various network and devices have attracted more and more attention. Nowadays, multimedia communications over the internet especially wireless network are the necessarily part of modern life. Many researchers have studied on video streaming management over wireless, including video coding [1–4], network coding [5–8], and wireless communication [9–11]. In wireless networks, video streaming applications are almost the largest consumer of mobile wireless data, so scheduling video streaming in wireless networks is very important.

Video streaming scheduling in wireless networks has been developed for recent years, such as given quality on service mechanisms [12–16] and real-time adaptive [17–19]. Many of video streaming scheduling studies are about hard real-time [20–22] that requires all the packets complete before the deadline; otherwise the packets will be considered invalid and dropped. However, for some actual video streaming media application, the packets of the transmission with delay can still bring a certain service and these packets can be forwarded and represented by soft real-time tasks. The soft real-time task model with value function is also applicable to video streaming with delay [23, 24]. For soft real-time scheduling, task executing under delay-constrained can bring goodput and benefit.

Generally, real-time scheduling can be divided into preemptive scheduling and non-preemptive scheduling according to whether the task can preempt each other or not. In a dynamic network, the preemptive method is more applicable. However, arbitrary preemption between tasks will affect the scheduling performance. Therefore, considering the drawbacks of the two methods, a compromise approach is utilized to limit the arbitrary or invalid preemption by setting preemption threshold. Furthermore, the limited preemption scheduling strategy is a correct choice for improving the performance.

Soft real-time task model is conducted for the video streaming under wireless. We propose a soft real-time threshold preemption scheduling strategy by combining value density and urgency for video streaming in the paper. Simulation results show that the strategy can improve the goodput, reduce the delay, and increase the benefit comparing with the EDF and LSF methods.

In this paper, we only consider the CPU in the system scheduling and ignore the time caused by context switching. We study the video streaming transmission problem of gateway devices in wireless network at the specific level of packets of transport tiers.

Organization. The structure of the paper is organized as follows: Section 2 describes the related work. Section 3 designs the structure of DSPT strategy. Section 4 defines the task model and problem analysis in detail. Section 5 is priority and preemption thresholds description. The details of scheduling based on preemption threshold are introduced in Section 6. The simulation setup and results are given in Section 7. Finally, the conclusions and future work are discussed.

2. Related Work

In wireless networks, the scheduling strategy should ensure high quality. For this purpose, lots of studies were proposed. The works in [1, 2] enabled high-quality HTTP streaming, in which video pre-encoded into segments that can only start to play if all the packet have received. However, much packet lost for video application in the above work. The literature [3] optimized the dynamic HTTP live streaming service by segment adaption. Some literatures proposed video streaming scheduling using video coding with network coding. For example, the literatures [5, 6] optimized video streaming by combining video coding and network coding for P2P network and distributed network, respectively. However, it might suffer from bandwidth inefficiencies because of unrecoverable packets. The work [7] proposed streaming scheduling for multiple server by Markov decision processes, but it had high computational complexity. In [9], the authors designed a crosslayer optimization scheme for dynamic wireless video streaming. However, these methods cannot contain the packet urgency for streaming control.

With the emergence of various networks, the research on heterogeneous wireless networks had been motivated. The literatures [11, 19] studied the concurrent multipath transfer of the capacity-limited heterogeneous wireless platform. Another work [17] proposed a real-time adaptive video streaming deliver method over multiple wireless networks. In [23, 24], delay-constrained video transmission with multiple interface in heterogeneous network was proposed.

Some researchers considered video streaming as hard real-time; that is, the streaming packet missing the deadline will be considered invalid. For example, the literature [20] proposed a feedback-based control approach under hard real-time workload to improve admission rate. In literatures [18, 21], the author studied in hard real-time video stream allocation to improve the utilization of video decoding. Nevertheless, most video streaming data meets the characteristics of soft real-time task, so delay-constrained of video is considered acceptable in applications.

3. DSPT Detail Design

We consider video streaming deliver over wireless networks as soft real-time scheduling. The streaming packet is expressed as task, and its quality and urgency are the metrics in scheduling. The task's delay will affect the quality. Therefore, we introduced the value to measure the quality. Response time analysis (RTA) is used to test the schedulability of real-time task, and the preemption threshold-based scheduling approach is to reduce invalid preemption among tasks. The scheduling algorithm can effectively reduce the delay and improve the success rate and benefit. The flow diagram of DSPT strategy is shown in Figure 1. In this paper, we assume that the value function $V_i(t)$ of the soft real-time task T_i keeps unchanged in the interval $[a_i, d_i)$ and decreases linearly in the interval. While executing in the interval $[d_i, cr_i)$, the task T_i is not allowed to be preempted in order to protect the execution of the task.

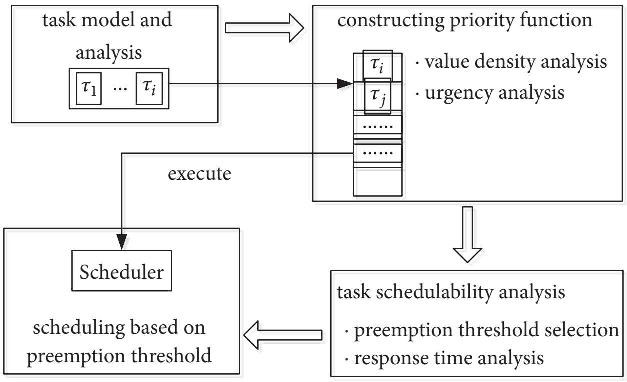


Figure 1: The flow diagram of DSPT strategy.

Contribution 1. We construct the priority assignment function combining value density and urgency, improving the success and quality of task soft real-time scheduling.

Contribution 2. We propose the task scheduling strategy DSPT based on preemption threshold, which can obtain better performance.

4. Task Model and Analysis

Considering video streaming with delay-constrained under wireless, we conduct soft real-time task model in this section.

4.1. Soft Real-Time Task Model. We refer to a task as an aperiodic task model, and the task is soft real-time. This section provides the definitions of soft real-time task model used in the paper.

Task T_i: An aperiodic task T_i is defined as a triple $T_i := \{a_i, et_i, d_i, cr_i\}$, where a_i,et_i,d_i, and cr_i are the arrival time, the estimated execution time, the relative deadline, and the critical time of task T_i.

Having executed time ht_i: ht_i denotes that the task has been executing ht_i units at time t_0.

Required execution time rt_i: rt_i denotes the remaining execution time of task T_i. Thus, it meets $rt_i = et_i - ht_i$.

Executable time lt_i: lt_i is the executable time of the task T_i prior to its deadline d_i, where at the time $t_0 (t_0 < d_i)$.

Soft Executable time slt_i: When missed deadline d_i, task can continue to execute until cr_i. slt_i is the executable time prior to the critical time cr_i, that is, $slt_i = cr_i - t_0$.

Slack time st_i: The slack time st_i denotes that the time unit after finish execution until deadline d_i, thus, $st_i = lt_i - rt_i$. st_i remains stable when the task T_i is in execution, and st_i decreases when the task T_i is preempted. Once $st_i < 0$, the task T_i cannot finish before d_i.

Delay Time dt_i: The delay time dt_i denotes the finish time of task T_i after d_i. At the time $t_0 (d_i < t_0 \le cr_i)$, it meets $dt_i = t_0 - d_i$.

Priority pr_i: pr_i is defined as the priority of task T_i. The larger pr_i's value, the higher T_i's priority.

Threshold θ_i: θ_i is the preemption threshold of task T_i, where $\theta_i \ge pr_i$.

Value function $V_i(t)$: $V_i(t)$ defines the value of the system for executing T_i at time t, where $t \in [a_i, cr_i)$. The task T_i 's value will change when T_i keeps executing, and the function $V_i(t)$ is dynamic. $V_i(t)$ will be analyzed in detail in Section 4.

Value density function $VD_i(t)$: $VD_i(t)$ is the ratio between $V_i(t)$ and required execution time rt_i, that is, $VD_i(t) = V_i(t)/rt_i$. Obviously, $VD_i(t)$ is growing with the decreasing of rt_i.

4.2. Value and Value Density Function of Soft Real-Time Task. According to assumption, the value function $V_i(t)$ of the soft real-time task T_i at time t is defined as

$$V_i(t) = \begin{cases} V_i & t < d_i \\ \dfrac{cr_i - t}{cr_i - d_i} \times V_i & d_i \le t < cr_i \end{cases} \tag{1}$$

From (1), we can see that when $t < d_i$, $V_i(t)$ is stable at V_i; however when $d_i \le t < cr_i$, $V_i(t)$ is linearly diminishing until 0.

Value density function $VD_i(t)$ is denoted as

$$VD_i(t) = \begin{cases} \dfrac{V_i}{rt_i} & t < d_i \\ \dfrac{cr_i - t}{cr_i - d_i} \times \dfrac{V_i}{rt_i} & d_i \le t < cr_i \end{cases} \tag{2}$$

We can see that $VD_i(t)$ is not only concerned with $V_i(t)$ but also related to rt_i. According to the definition, $rt_i = et_i - ht_i$. As we known, when the task T_i is executing, rt_i decreases with the growing of the time t. And when the task T_i is blocked, the rt_i remains unchanged. The discussion of $VD_i(t)$ is as follows:

(1) $t < d_i$. Within this interval, $VD_i(t) = V_i/rt_i$. When T_i is executing, $VD_i(t)$ grows with the decreasing of rt_i. While T_i is blocked, $VD_i(t)$ remains unchanged.

(2) $d_i \le t < cr_i$. In this interval, $VD_i(t)$ can be derived as follows: $VD_i(t) = (V_i/rt_i) \times ((cr_i - t)/(cr_i - d_i))$

From the above expression, the affect part of the $VD_i(t)$ is

$$\frac{cr_i - t}{rt_i} \tag{3}$$

Because the task in scheduler can finish without any interference of any other task, $(cr_i - t)/rt_i > 1$. In this case, if the task T_i keeps executing, the molecular part and the denominator part of (3) decrease at the same time. Further analysis is needed:

(1) At t_0 time, (3) is recorded as: $(cr_i - t_0)/rt_i$. When task T_i has executed Δt from t_0, (3) is formulated to $(cr_i - (t_0 + \Delta t))/(rt_i - \Delta t)$. Because $slt_i/rt_i > 1$, then $(cr_i - (t_0 + \Delta t))/(rt_i - \Delta t) > (cr_i - t_0)/rt_i$, where $\Delta t > 0$. That is, when task T_i is executing, $VD_i(t)$ grows with the decreasing of rt_i.

(2) When the task T_i is blocked, the molecular part of (3) decreases with the growth of t, while the denominator rt_i is kept unchanged. Thus, (3) decreases with the growth of t. Therefore, Theorem 1 can be obtained.

Theorem 1. *While the task keeps executing in the progress, it increases with time t.*

4.3. Analysis of the Task's Urgency. The more urgency of the task is, the sooner execution is required.

Definition 2. At any t, the urgency $Ug_i(t)$ of task T_i is defined as $Ug_i(t) = 1/e^{st_i}$.

$Ug_i(t)$ is affected by st_i, the discussion is as follows:

(1) When T_i remains in execution, st_i does not change with the growing of the time t and then $Ug_i(t)$ remains unchanged.

(2) When T_i is blocked, st_i decreases with the growth of t, which makes $Ug_i(t)$ increase.

5. Priority and Preemption Thresholds for Soft Real-Time Tasks

Task scheduling is driven by priority, while task priority function combines value density with urgency.

5.1. The Construction of Task's Priority Function. Considering the urgency and the benefits of task, task scheduling is based on priority. Only the completed task can gain value, so relative to the value density, the urgency is more important. In system scheduling, the first consideration is system success rate and then is value brought by tasks. It means that urgency is given much greater weight than value density in priority function. Therefore, we use exponential weighting to the urgency while using logarithmic weighting to the value density. The function of priority assignment can be rewritten to

$$pr_i = Ug_i(t) \times \ln[1 + VD_i(t)] \tag{4}$$

Based on the above analysis of value density $VD_i(t)$ and urgency $Ug_i(t)$, we can conclude that

(1) When T_i executes at interval $[a_i, d_i)$, pr_i satisfies

$$pr_i = \frac{1}{e(st_i)} \times \ln\left[1 + \frac{V_i}{rt_i}\right] \tag{5}$$

Further analysis is needed:

(1) According to Theorem 1, when T_i keeps running, $\ln[1+V_i/rt_i]$ increases. Because st_i remain unchanged with time t growing. Therefore, (5) keeps increasing.

(2) Once T_i is preempted, $1/e^{st_i}$ augments as time t grows, while $\ln[1+V_i/rt_i]$ remain unchanged, then (5) keeps increasing. Based on above, pr_i increases as time t at interval $[a_i, d_i)$.

(2) When T_i executes at $[d_i, cr_i)$, its slack time is slt_i and satisfies

$$pr_i = \frac{1}{e^{st_i}} \times \ln\left[1 + \frac{cr_i - t}{cr_i - d} \times \frac{V_i}{rt_i}\right] \tag{6}$$

(1) In the process of running T_i, with the growing of the time t, $((cr_i-t)/(cr_i-d_i))\times(V_i/rt_i)$ increases while slt_i remains unchanged and then (6) keeps increasing.

(2) When T_i has been preempted, with the growing of time t, slt_i decreases and $1/e^{slt_i}$ augments accordingly while $((cr_i - t)/(cr_i - d_i)) \times (V_i/rt_i)$ reduces. Further discussion of Eq. (6) is needed. At moment t, $slt_i = cr_i - t - rt_i < cr_i - d_i$. After task T_i has been executed with Δt, there is $pr_i(\Delta t) = (1/e^{cr_i-t-rt_i-\Delta t}) \times \ln[1 + ((cr_i - t - \Delta t)/(cr_i-d_i))\times(V_i/rt_i)]$, where $cr_i-t-rt_i-\Delta t > 0$. And $1/e^{cr_i-t-rt_i-\Delta t}$ is exponential growth, in which growth rate is faster than $\ln[1+((cr_i-t-\Delta t)/(cr_i - d_i)) \times (V_i/rt_i)]$'s logarithmic descent rate significantly. Therefore, $pr_i(\Delta t)$ increases when Δt grows.

5.2. Preemption Threshold Selection for Task. The selection of preemption thresholds directly affects the scheduling algorithm. We select preemption threshold by analyzing the task's response time.

Definition 3. Task T_i's Effect Job Set is denoted as EJS_i. During the execution of task T_i, all task of the sets can preempt T_i, which are defined as Effect Job Set (EJS_i), satisfying $EJS_i = T_m \mid (pri_m > pri_i \wedge (D_m > A_i) \wedge (A_m < D_i)$.

Response time Rt_i of task T_i contains two parts: estimated execution time et_i and sum of the remaining execution time of the effect job sets EJS_i:

$$Rt_i = et_i + \sum_{T_j \in EJS_i} rt_j \tag{7}$$

By analyzing the response time of T_i, its threshold θ_i can be obtained under the condition which can be scheduled. The procedure of preemption threshold selection is shown in Algorithm 1, where T_{max} is the highest priority task.

Obviously, the computational complexity of Algorithm 1 is $O(n)$.

```
Calculate $\theta_i$ using (7);
$\theta_i = pr_i$;
WHILE ($Rt_i \le cr_i$) DO
  IF ($\theta_i > pr_{max}$)
    return not scheduled;
  ELSE
    IF $T_i$ is executing and $st_i < st_{max}$
      $\theta_i := pr_{max}$;
      return $\theta_i$;
    ELSE
      Calculate $Rt_i$ using (7)
    END IF
  END IF
END WHILE
return $\theta_i$
```

Algorithm 1: Compute θ_i.

6. Task Scheduling Strategy Based on Preemption Threshold

We propose a limited preemption scheduling strategy DSPT based on preemption threshold, which reduces the blocking time of the tasks by restricting preemption.

Definition 4. Sufficient and necessary conditions of $pr_i > \theta_j$ is that task T_i can preempt task T_j.

By Definition 4, if task T_i preempts task T_j, there must be $pr_i > pr_j$.

Theorem 5. *If task T_i and task T_j are not preempted, the following should be satisfied: $(pr_i \le \theta_j) \wedge (pr_j \le \theta_i)$.*

Proof. Supposing that there are two tasks, T_p and T_q, which do not preempt each other, $\urcorner((pr_p \le \theta_q) \wedge (((pr_q \le \theta_p)))) \Longrightarrow (pr_p > \theta_q) \vee (pr_q > \theta_p)$ is satisfied.

There are two possible cases:

(1) If $pr_p > \theta_q$, task T_p can preempt task T_q,

(2) If $pr_q > \theta_p$, task T_p may be preempted by task T_q.

Obviously, tasks T_p and T_q can preempt each other, in contradiction with Definition 4. Therefore, Theorem 5 can be proved. □

6.1. Scheduling Algorithm Based On Preemption Threshold. The soft real-time dynamic task scheduling based on preemption threshold (DSPT) called Algorithm 1 is to calculate the threshold and determines the scheduling queue dynamically as shown in Algorithm 2.

6.2. Algorithm Complexity. There is only one loop in Algorithm 2, that is, the 7th step is the loop of the length that is equal to the number of tasks in the ready queue. Thus, the computational complexity of the Algorithm 2 is $O(n \times m)$, and the computational complexity of the whole scheduling algorithm is $O(n \times m)$.

```
$\theta_i$: preemption threshold of $T_i$;
$pr_i$: priority of $T_i$;
$S_j$: ready tasks queue. $1 \le j \le num$,
$num$ is the number of tasks in the queue;
$T_{arr}$: task just arriving;
$T_{exe}$: task in executing;
$T_{max}$: the highest priority task of $S_j$;
$T_{arr}$ arrive;
IF $num = 0$;
    $T_{exe} := T_{arr}$;
    EXECUTE $T_exe$;
ELSE
    Compute $pr_{arr}$ using (7);
    FOR $i = 1$ to $num$ DO
        Compute $pr_i$ using (5);
    Compute $\theta_i$;
    SORT task in $S_j$ by $pr_j$ Desc;
END IF
IF $pr_{arr} > \theta_{max}$
    $T_{exe} := T_{arr}$;
    IF $T_{in}$ in $S_j$ can't Finish
        ABORT $T_{in}$;
    ELSE
        $T_{exe} := T_{max}$;
        ADD $T_{arr}$ to $S_j$;
        $num++$;
    END IF
    EXECUTE $T_{exe}$;
END IF
```

Algorithm 2: Task scheduling.

7. Simulations and Analysis

7.1. Simulation Experiments. The experiment platform's CPU is Intel dual-core 2.8 GHZ processor, and the memory is 8 GB. The operating system is Ubuntu Linux, and the source code is implemented by C language.

In the experiment, we consider the aperiodic soft real-time task. Any task T_i's arrival time a_i obeys Poisson distribution ($\lambda = 4$) and the estimated execution time et_i is subject to $[3.0, 5.0]$ average distribution. Therefore, its relative deadline d_i is $[1.2, 1.5] \times et_i$ average distribution, its critical time cr_i is $[1.2, 1.5] \times d_i$, and its value V_i is randomly selected between $[10, 50]$.

Performance indicators are included as follows:

(1) Systematic success rate (SSR):

$$SSR = \frac{N_s}{N} \tag{8}$$

where N_s denotes as the number of successful tasks and N represents the number of system overages.

(2) Average task latency rate ATDR:

$$ATDR = avg\left(\sum_{T_i \in TS} \frac{el_i}{et_i}\right) \tag{9}$$

where TS denotes the successful task set and el_i and et_i denote execution latency and execution time of the success task T_i.

(3) Effective utilization Rate (EUR):

$$EUR = \frac{CT_s}{CT} \tag{10}$$

where CT_s denotes as the CPU time for successful tasks and CT represents the total CPU time in scheduling.

(4) Cumulative Value (CV):

$$CV = \sum_{T_i \in TS} V_i \tag{11}$$

where TS denotes the successful task set and represents the value of the task T_i.

7.2. Performance Analysis. Under soft real-time environment, the preemptive threshold scheduling strategy is compared with Earliest Deadline First (EDF) method and Least Slack First (LSF) method, and the simulation results and performance are analyzed as follows.

(1) Figure 2 shows simulation results for SSR scheduled by DSPT, EDF and LSF. The task's arrival interval is represented on the horizontal axis, and the vertical axis is SSR. When the arrival interval of the task is very short, the preemption among the tasks causes the SSR to be low. With the increasing of arrival interval, the preemption among tasks decreases and the SSR arguments. From Figure 2, SSR is sorted in descending order followed by DSPT, LSF, and EDF. This is due to the EDF algorithm chooses task with earliest deadline but not most urgent, which can lead to many tasks fail. While the LSF algorithm always chose the task with the shortest slack time dynamically, its unrestricted preemption causes the thrashing which can make lots of task fail. The DSPT can adapt to the system environment dynamically, and its preemption threshold can avoid the invalid preemption among tasks. (2) The simulation results of ATDR are shown in Figure 3, in which the ATDR is on the vertical axis and arrival interval is on the horizontal axis. It can be seen from Figure 3, the ATDR decreases since the arrival interval grows from 2.5, while ATDR is lower because the intervals among tasks are so short that many of these are dropped to 1.0. Moreover, EDF algorithm leads to the highest ATDR, while the DSPT method produces the lowest ATDR. Because the EDF algorithm always chooses the task with earliest deadline of slack time to execute, many tasks miss the deadline and delay. Relative to the EDF method, the LSF algorithm assigns the most urgency task to execute in time. But its unconstrained preemption may cause serious thrashing among tasks under system overload and this increases the ATDR. The DSPT approach chooses the most urgency task to execute, and limits the preemption among the tasks by the threshold, thereby reduces the delay of the task effectively.

(3) The simulation results for EUR scheduled by EDF, LSF, and DSPT are shown in Figure 4, where the horizontal

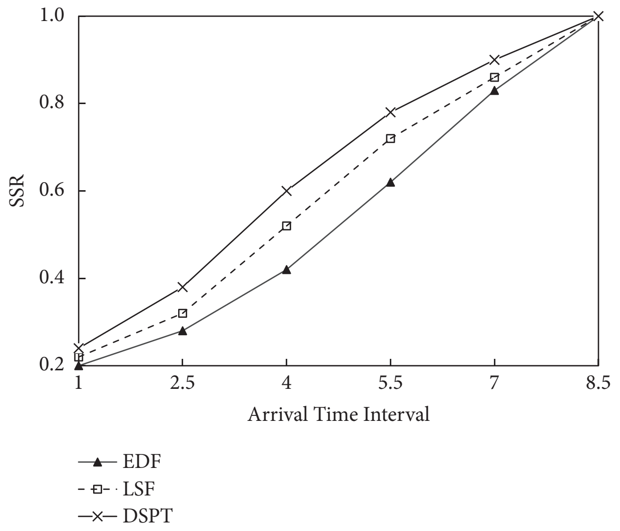


Figure 2: SSR results.

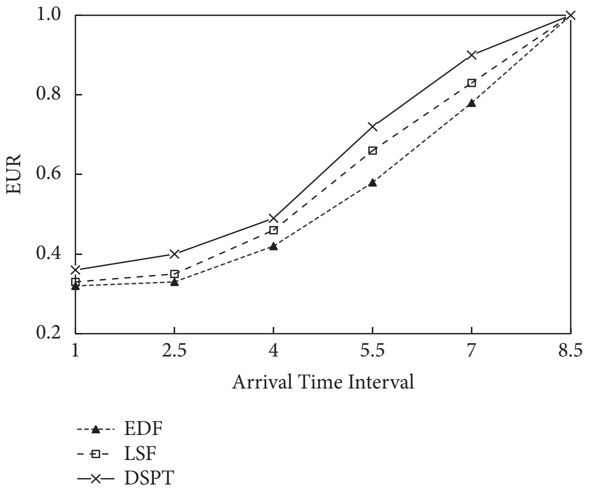


Figure 4: EUR results.

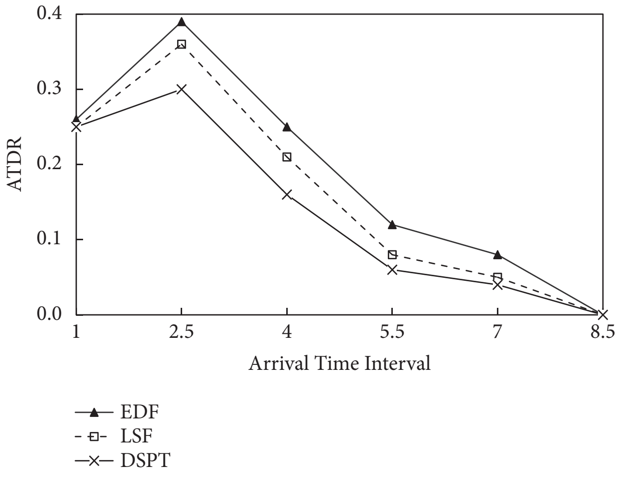


Figure 3: ATDR results.

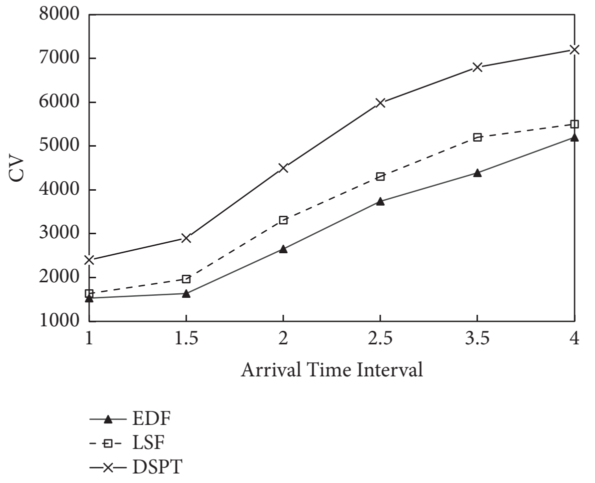


Figure 5: CV results.

axis is task's arrival interval and the vertical axis is EUR. The short arrival interval means intense preemption among tasks, which causes the EUR low. While the arrival interval becomes longer, there are EUR arguments as the tasks' preemption decreases. From Figure 4, DSPT obtains the best EUR and EDF has the least EUR. Because the EDF algorithm always execute the task with earliest deadline, which causes more invalid CPU utilization than LSF along with the task failure. The DSPT using preemption threshold can improve the EUR through reducing invalid preemption of tasks.

(4) Figure 5 plots the system's CV as arrival interval grows. It can be seen from Figure 5, the CV increases when the arrival interval of tasks increases. DSPT obtains the highest CV, because it assigns higher executing priority of the task with high value density. The LSF method produces better CV than EDF method, because under the same system load, that more tasks finish augments CV obtained with the formal approach.

8. Conclusions and Future Works

In this paper, we consider video streaming scheduling in wireless networks as soft real-time scheduling. The soft real-time task model has been constructed to express streaming packet and its quality, measure as value, and urgency metric are explored in scheduling. Response time analysis (RTA) approach has been used to test the task's schedulability, and preemption threshold has been added to the schedule to release the invalid preemption. The scheduling schema DSPT can constrain the delay and improve the success ratio and benefit effectively.

In future works, we want to apply our scheduling method to practical wireless networks, like 5G net, to test its practical effect. Moreover, the stream scheduling of mixed hard real-time and soft real-time characterizes are also a study work in the next step.

Data Availability

The data, including task's properties and performance indicators in the experiments, used to support the findings of this study are available from the corresponding author upon request.

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 (NSFC) (Grant nos. 61562044, 41661083) and the Science and Technology Research Project of Jiangxi Provincial Department of Education (Grant nos. GJJ170234, GJJ160781).

References

[1] L. De Cicco and S. Mascolo, "An adaptive video streaming control system: Modeling, validation, and performance evaluation," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 526–539, 2014.

[2] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, 2012.

[3] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 693–705, 2014.

[4] L. He and F. Li, "Content and buffer status aware packet scheduling and resource management framework for video streaming over LTE system," *Eurasip Journal on Image and Video Processing*, vol. 2017, no. 1, 2017.

[5] A. M. Sheikh, A. Fiandrotti, and E. Magli, "Distributed scheduling for scalable P2P video streaming with network coding," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 11-12, April 2013.

[6] A. M. Sheikh, A. Fiandrotti, and E. Magli, "Distributed scheduling for low-delay and loss-resilient media streaming with network coding," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2294–2306, 2014.

[7] N. Thomos, E. Kurdoglu, P. Frossard, and M. Van Der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 893–906, 2015.

[8] S. Huang, E. Izquierdo, and P. Hao, "Adaptive packet scheduling for scalable video streaming with network coding," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 10–20, 2017.

[9] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, 2015.

[10] D.-R. Chen, "A real-time streaming control for quality-of-service coexisting wireless body area networks," *Applied Soft Computing*, 2017.

[11] Y. Cao, Q. Liu, Y. Zuo, G. Luo, H. Wang, and M. Huang, "Receiver-assisted cellular/wifi handover management for efficient multipath multimedia delivery in heterogeneous wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.

[12] C. Sieber, T. Hosfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC," in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management, IM 2013*, pp. 1318–1323, Belgium, May 2013.

[13] S. A. Hosseini, F. Fund, and S. S. Panwar, "(Not) yet another policy for scalable video delivery to mobile users," in *Proceedings of the 7th ACM Workshop on Mobile Video, MoVid 2015*, pp. 17–22, USA, March 2015.

[14] S. A. Hosseini and S. S. Panwar, "Restless streaming bandits: Scheduling scalable video in wireless networks," in *Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 620–628, Monticello, IL, USA, October 2017.

[15] Y.-H. Yeh, Y.-C. Lai, Y.-H. Chen, and C.-N. Lai, "A bandwidth allocation algorithm with channel quality and QoS aware for IEEE 802.16 base stations," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1601–1615, 2014.

[16] J. Kim and E.-S. Ryu, "QoS optimal real-time video streaming in distributed wireless image-sensing platforms," *Journal of Real-Time Image Processing*, vol. 13, no. 3, pp. 547–556, 2017.

[17] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 795–805, 2014.

[18] H. R. Mendis, N. C. Audsley, and L. S. Indrusiak, "Task allocation for decoding multiple hard real-time video streams on homogeneous NoCs," in *Proceedings of the 13th International Conference on Industrial Informatics, INDIN 2015*, pp. 246–251, UK, July 2015.

[19] Y. Cao, F. Song, Q. Liu, M. Huang, H. Wang, and I. You, "A LDDoS-Aware energy-efficient multipathing scheme for mobile cloud computing systems," *IEEE Access*, vol. 5, pp. 21862–21872, 2017.

[20] P. Dziurzanski, A. K. Singh, and L. S. Indrusiak, "Feedback-based admission control for hard real-time task allocation under dynamic workload on many-core systems," in *Architecture of computing systems—ARCS 2016*, vol. 9637 of *Lecture Notes in Comput. Sci.*, pp. 157–169, Springer, [Cham], 2016.

[21] W. Wang, Y. Cao, J. Gong, and Z. Li, "CP-TPS: A Real-Time Transaction Processing Strategy Supporting Compensatory Task," in *Proceedings of the 20th IEEE International Conference on Computational Science and Engineering and 15th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017*, pp. 475–482, China, July 2017.

[22] H. Mendis, N. Audsley, L. Indrusiak et al., "Dynamic and static task allocation for hard real-time video stream decoding on NoCs," *Leibniz Transactions on Embedded Systems*, vol. 4, no. 2, pp. 01–25, 2017.

[23] J. Wu, C. Yuen, N.-M. Cheung, and J. Chen, "Delay-Constrained High Definition Video Transmission in Heterogeneous Wireless Networks with Multi-Homed Terminals," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 641–655, 2016.

[24] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Adaptive Flow Assignment and Packet Scheduling for Delay-Constrained Traffic over Heterogeneous Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8781–8787, 2016.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 9678694, 9 pages
https://doi.org/10.1155/2018/9678694

Research Article

An Improved Artificial Bee Colony Algorithm in LoRa Wireless Communication System for Efficient Multimedia Transmission

Yan Song,[1] Lidong Huang ,[1] Panfeng Xu,[1] Lili Li,[1] Min Song,[2] and Yue Long[1]

[1] *College of Physics, Liaoning University, Shenyang, China*
[2] *College of Information, Shenyang Institute of Engineering, Shenyang, China*

Correspondence should be addressed to Lidong Huang; 4031731893@smail.lnu.edu.cn

Received 11 September 2018; Revised 28 October 2018; Accepted 18 November 2018; Published 3 December 2018

Guest Editor: Kai Wang

Video streaming communication networks will be a very important way to send multimedia information anytime and anywhere, and the construction of the network base station which transmits signals is crucial in future. However, there is a contradiction between the power consumption of LoRa nodes and the real-timeliness of mesh network. In order to solve the contradiction, this article aims to combine the mesh network of LoRa wireless communication system with an improved artificial bee colony algorithm. Specifically, an artificial bee colony algorithm, which is based on RBF radial basis neural network trained with random gradient method, is designed. Simulation results show that the proposed algorithm solves the contradiction between power consumption and real-timeliness effectively. When using this improved network system structure to send multimedia information, it shows obvious superiority in terms of the high efficiency and real-timeliness of multimedia transmission.

1. Introduction

With the development of wireless communication network technology, the advantages of intelligent productions of wireless communication are widely accepted, such as low cost and good scalability [1–3]. These productions have been paid more and more attention by the society [4, 5]. Moreover, the transmission of multimedia information is getting larger and more global. Video streaming and other high-capacity media transmission technologies require better wireless networks architecture [5, 6].

The LoRa technology solves the problem of long distance and low power consumption successfully. Compared to other wireless transmission technologies, LoRa has the advantages of longer transmission distance and lower consumption [5–7]. However, the existing wireless transmission technology cannot meet the real-time monitoring effect of wireless network under the conditions of low power consumption and long distance. In the existing methods, there is a mesh network architecture that minimizes the number of devices, which greatly reduces the cost of base station establishment [8, 9]. At the same time, it is very convenient to deploy the equipment [10–12]. The mesh network architecture is very stable, and it is not affected by a single node. When a near node fails or is disturbed, the packet will be transferred automatically to the alternate path for transmission. Moreover, it has flexible structure and advantages in terms of overloading and communication load balancing [13–15]. This kind of mesh network reduces the interference of the adjacent user wireless network when transmitting data, and its efficiency of information transmission has been improved greatly.

To a certain extent, the mesh network reduces the power consumption and achieves the coverage of a larger area, but it cannot completely solve the contradiction between the power consumption and real time of LoRa wireless network [16–19]. Some additional interference or space extension may result in loss of signal transmission capability. We consider adopting the basic artificial swarm algorithm. There is still a problem that the wireless network layout may be optimized correspondingly [20, 21], but the effect of optimization is not obvious enough and it converges slowly. When looking for the optimal solution in space, there may be a deviation. Moreover, the traditional method in the algorithm will affect the convergence speed of the optimal solution greatly. Therefore, this existing method cannot deal with the contradiction between power consumption and real time [11, 16, 22], which is an obstacle to the multimedia information transmission. To fill this gap, the paper proposes an improved algorithm to solve the problem.

The contributions of this paper are listed as follows: an artificial bee colony algorithm based on RBF radical basis neural network (RBFABC) is designed to resolve this problem, and this algorithm can solve the contradiction between power consumption and real-timeliness [14, 17, 23]. Furthermore, this design not only improves the traditional way of selecting and updating the honey sources in basic artificial swarm optimization algorithm, but also enables the algorithm to converge towards the optimal solution. Moreover, the convergence speed of the proposed algorithm is much higher. The improved algorithm not only ensures the real-time performance of wireless network [14, 16, 24], but also reduces the power consumption of wireless network. In brief, the improved algorithm overcomes technical obstacles in the field of multimedia wireless network, and it has great significance to the development of wireless transmission technology.

The rest of this paper is organized as follows: Section 2 introduces the basic artificial swarm algorithm. Section 3 presents the system model. Section 4 proposes the RBFABC algorithm. Section 5 performs simulations to verify the effectiveness of the proposed algorithm. Section 6 concludes this paper.

2. Basic Artificial Swarm Algorithm

As everyone knows, bees have special behavioral patterns for finding honey sources [17, 18]. In general, the process of searching for honey sources is divided into three parts: looking for the honey sources; updating the honey sources; selecting the best honey sources [14, 25, 26]. The algorithm flow is shown in Algorithm 1.

```
Init $\boldsymbol{X_{mn}}$, $N$, $C = 1$
input initial population
set the total number $N$
while (iter < = maxcycle)
    $\boldsymbol{V_{ij}} = \boldsymbol{X_{ij}} + \emptyset_{ij}(\boldsymbol{X_{ij}} - \boldsymbol{X_{kj}})$
    apply the greedy selection process
    $P_i = \frac{fit}{\sum_{k=1}^{N} fit_i}$
    if $f_m$ >= 0 then
        $fit_m = \frac{1}{1 + fit_m}$
    else
        $fit_m = 1 + |fit_m|$
    end if
        if $P_{m+1} < P_m$ then
  $\boldsymbol{X_{mn}} = \boldsymbol{X_{min}} + \mathrm{rand}(0,1)(\boldsymbol{X_{mn}} - \boldsymbol{X_{min}})$
        end if
    memorize the best position so far
    $C = C + 1$
    until $C$ = maximum cycle number
end while
output the optimal solution
end
```

Algorithm 1: The program diagram of the artificial swarm optimization algorithm.

In this algorithm, the leader bees have a memory function, which can save the information of the honey sources. Moreover, the leader bees share the honey sources information with orientation bees. According to their dance of the leader bees, the orientation bees determine which direction to go. The observation bees are responsible for searching nectar around the hive, and the leader bees search for honey sources [27, 28]. We define some variables to represent them [16, 17, 23]. Firstly, a series of original honey sources are initialized randomly. $\boldsymbol{X} = (X_1, X_2, X_3, \ldots, X_n)$ represents n honey sources [18, 19, 25]. The position of the m-th honey source is $\boldsymbol{X_m} = [X_{m1}, X_{m2}, X_{m3}, \ldots, X_{mn}]$ $(m = 1, 2, 3, \ldots, n)$, which are potential solutions to the problem [26, 27]. Then the fitness of honey sources are evaluated by the following formula:

$$fit_m = \begin{cases} \dfrac{1}{1 + f_m} & f_m > 0 \\ 1 + |f_m| & f_m < 0 \end{cases} \tag{1}$$

where f_m represents the target function value of m-th solutions. After that, the location of n honey sources updates in a certain way. The process is described as follows.

$$\boldsymbol{X_m^n} = \boldsymbol{X_{min}^n} + \mathrm{rand}\,(0,1)\left(\boldsymbol{X_{max}^n} - \boldsymbol{X_{min}^n}\right) \tag{2}$$

If the honey sources are better, we will select the better one instead of the old one; otherwise keep the old one. If all the leader bees finished searching, the honey sources information will be transmitted to the observation bees [4, 13]. Then the observation bees choose leader bees according to the fitness value [28, 29], and the selection probability is described as follows:

$$P_m = \frac{fit_m}{\sum_{i=1}^{n} fit_m} \tag{3}$$

$$\boldsymbol{U_{ij}} = \boldsymbol{X_{ij}} + \boldsymbol{r_{ij}}\left(\boldsymbol{X_{ij}} - \boldsymbol{X_{kj}}\right) \tag{4}$$

where $\boldsymbol{X_{ij}}$ represents the location of honey sources, $\boldsymbol{U_{ij}}$ represents the nearby location of $\boldsymbol{X_{ij}}$, $\boldsymbol{r_{ij}}$ represents the random number within (-1, 1), P_m represents the selective probability, and fit_m represents the fitness of honey sources. If the searching process is over, the honey sources information will be transmitted to the leader bees. Then the leader bees choose a direction according to fitness of honey sources. If this honey source is not replaced after finite loops, it will be abandoned. Finally the leader bees will look for the new honey sources.

3. System Model

In order to achieve the effect of algorithm optimization, we analyze the parameters of network architecture and establish the mathematical model of wireless network. As shown in Figure 1.

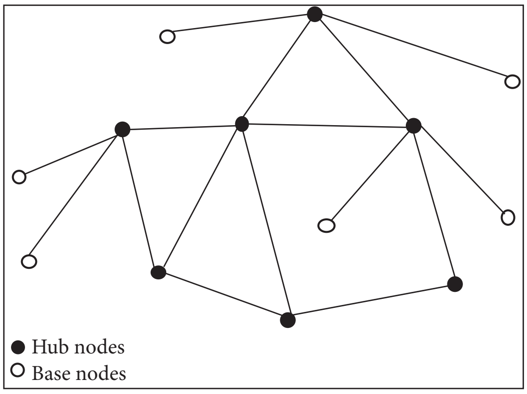


Figure 1: Mesh network model.

Beyond that, this paper combines the central function with the improved algorithm. In wireless mesh networks, a small amount of wireless devices can be required to cover a large range [8, 11, 15]. In other words, it is very convenient to create a mathematical model of wireless network [16–18]. The correlation signal transmission capability function of network nodes created is described as follows.

$$c_m = w_1\alpha_{m1} + w_2\alpha_{m2} + w_3\alpha_{m3} + w_4\alpha_{m4} + w_5\alpha_{m5} \quad (5)$$

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1 \quad (6)$$

The signal coverage range of nodes are related to the performance of the node (α_{m1}), the location of deployment (α_{m2}), the signal coverage (α_{m3}), the band width ($\alpha_{\mathrm{m}4}$), and the anti-interference capability (α_{m5}). The above factors have different effects on the signal coverage capability of nodes, and w is used to reflect the degree of influence of factors [7]. Under ideal conditions, the relationship between the transmitting power consumption and propagation distance of nodes is described as follows:

$$P = s + k\lg d + k\lg f \quad (7)$$

where P represents transmission power, d represents coverage range, f represents transmission frequency, s represents initial effective distance, and k represents effective radiofrequency factor [7, 16, 18]. In order to balance the relationship between power consumption and real time, the mathematical model is established as follows:

$$T_m = \frac{C_m}{C_{max}} \quad (8)$$

$$P = s + k\lg dT_m + k\lg f \quad (9)$$

where T_m represents the signal coverage capability after quantization, C_m represents the signal coverage capability of nodes, and C_{max} represents the maximum ability of nodes signal coverage.

4. RBFABC Algorithm

Although the artificial bee colony algorithm works well, there is still a problem of slow convergence rate in solving the problem of nodes optimization. At the same time, the choice of roulette in the original algorithm may make the algorithm fall into the local optimum. Therefore, in each iteration, there will be an error with the newly generated solution due to each local optimum. Because of the limitation of adaptive value, the speed of the algorithm convergence is affected. The gradient method, which is in the RBF radial basis neural network algorithm, is used to deal with the updating factors [10, 21, 25, 26]. Thus we improve the updating factors of honey sources by the RBF. Firstly, within a specified range, a series of original nodes locations are initialized randomly. $\boldsymbol{X} = (X_1, X_2, X_3, \ldots, X_n)$ represents n honey sources. The position of the m-th honey source is described as follows.

$$\boldsymbol{X_m} = [X_{m1}, X_{m2}, X_{m3}, \ldots, X_{mn}] \quad (10)$$

This is a potential solution to the optimization process [9, 11]. The degree of honey source is estimated by the instantaneous error function [24–27], as follows:

$$\begin{aligned} J(n) &= \frac{1}{2}|\boldsymbol{e}(\boldsymbol{n})|^2 \\ &= \frac{1}{2}\left|y_d - \sum_{k=1}^{N}\boldsymbol{w_k}(\boldsymbol{n})\,\emptyset\left\{\boldsymbol{x}(\boldsymbol{n}), \boldsymbol{c_k}(\boldsymbol{n})\right\}\right|^2 \end{aligned} \quad (11)$$

where $\boldsymbol{w}(\boldsymbol{n})$ represents the weighted value of the impact factors, $\boldsymbol{c_k}(\boldsymbol{n})$ represents the center of the radial basis function, and $\boldsymbol{\delta_k}(\boldsymbol{n})$ represents variance [27, 28].

When the instantaneous error is less than the specified error, the current honey sources are selected as the target honey sources. Moreover, the formula of searching for honey sources is described as follows.

$$\boldsymbol{y_m} = \sum_{k=1}^{N}\boldsymbol{w_{mk}}\left[s + k\lg\left\|\boldsymbol{x}(\boldsymbol{n}) - \boldsymbol{c_k}(\boldsymbol{n})\right\|^2 + k\lg f\right] \quad (12)$$

$$\boldsymbol{w}(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{w}(\boldsymbol{n}) - \mu_w\frac{\partial}{\partial\boldsymbol{w}}J(n) \quad (13)$$

$$\boldsymbol{c_k}(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{c_k}(\boldsymbol{n}) - \mu_c\frac{\partial}{\partial\boldsymbol{c_k}}J(n) \quad (14)$$

$$\boldsymbol{\delta_k}(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{\delta_k}(\boldsymbol{n}) - \mu_\delta\frac{\partial}{\partial\boldsymbol{\delta_k}}J(n) \quad (15)$$

During the updating process [22, 23, 26], $\boldsymbol{w}(\boldsymbol{n}+\boldsymbol{1})$, $\boldsymbol{c_k}(\boldsymbol{n}+\boldsymbol{1})$, and $\boldsymbol{\delta_k}(\boldsymbol{n}+\boldsymbol{1})$ will be updated according to the network correction equation. If these honey sources are not replaced, the position will be reserved. Then the leader bees continue to search for new honey sources. Finally, the new honey sources will be searched as a substitute of position by the improved updating formula. This process is shown in Algorithm 2.

The general steps of the RBFABC algorithm are as follows: initializing a series of original nodes location; searching for the position of some new nodes around the initialized nodes; calculating the instantaneous error value of the position of the new nodes; selecting the better nodes [6, 7, 27]; generating some new nodes around the better nodes by the updating formula; selecting better nodes with instantaneous

```
Init $\boldsymbol{X}_{mn}$, $N$, $C = 1$
input initial population
set the total number $N$
while (iter <= maxcycle)
    produce new solutions
$y_m = \sum_{k=1}^{N} \boldsymbol{w}_{mk} \left[ s + k \lg \left\| \boldsymbol{x}(\boldsymbol{n}) - \boldsymbol{c}_k(\boldsymbol{n}) \right\|^2 + k \lg f \right]$
    apply the greedy selection
    $J_n = \frac{1}{2} |\boldsymbol{e}(\boldsymbol{n})|^2 = \frac{1}{2} |\boldsymbol{y}_d - \boldsymbol{y}_m|^2$
    if ($J_n$ <= fitness) then
     $\boldsymbol{w}(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{w}(\boldsymbol{n}) - \mu_w \frac{\partial}{\partial \boldsymbol{w}} J(n)$
     $\boldsymbol{c}_k(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{c}_k(\boldsymbol{n}) - \mu_c \frac{\partial}{\partial \boldsymbol{c}_k} J(n)$
    $\boldsymbol{\delta}_k(\boldsymbol{n}+\boldsymbol{1}) = \boldsymbol{\delta}_k(\boldsymbol{n}) - \mu_\delta \frac{\partial}{\partial \boldsymbol{\delta}_k} J(n)$
    end if
memorize the best position so far
    $C = C + 1$
    until $C$ = maximum cycle number
end while
output the optimal solution
end
```

ALGORITHM 2: Program diagram of the RBFABC algorithm.

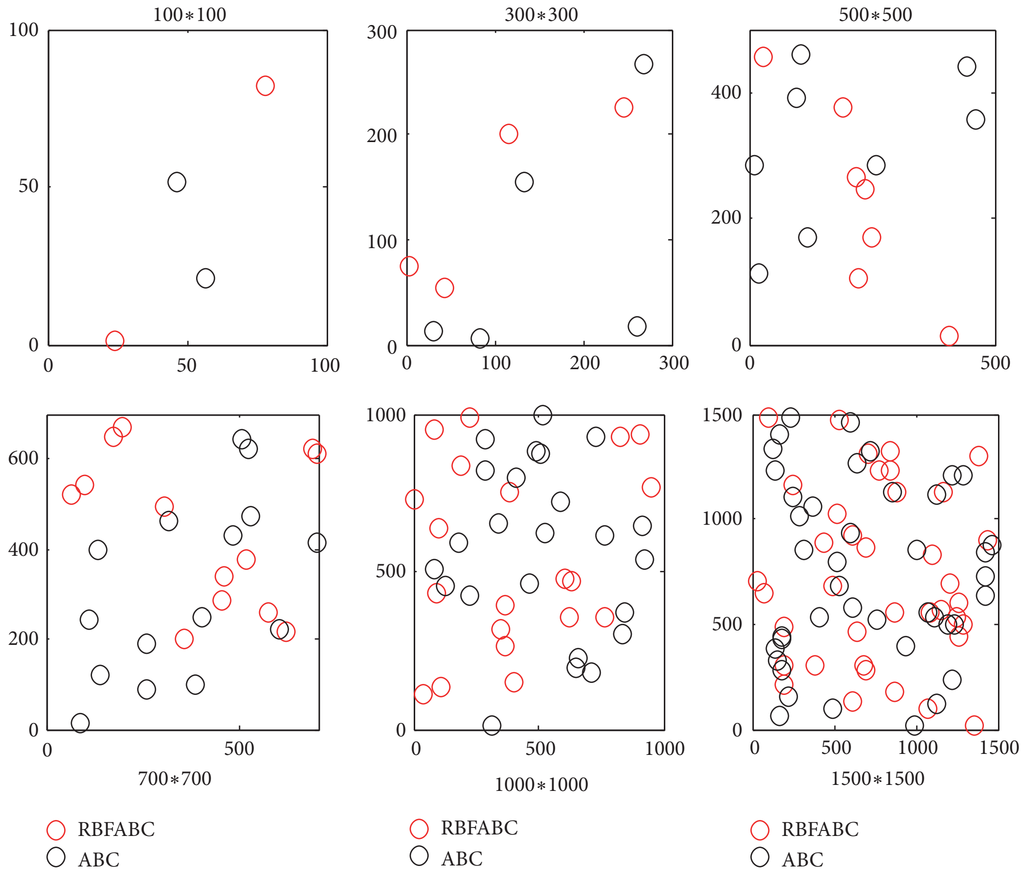


FIGURE 2: Node distribution under the same area.

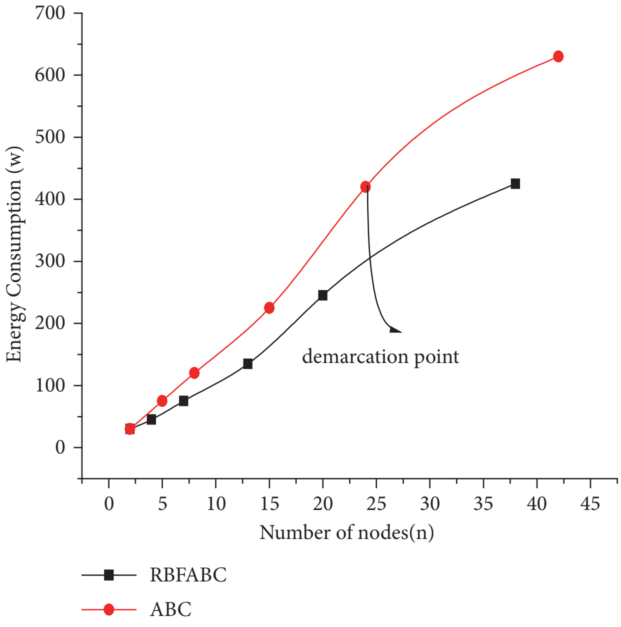


Figure 3: Power consumption diagram corresponding to different nodes.

error values; judging whether there are points which need to be abandoned, and if so, these nodes are converted to the alternate nodes; searching for new nodes which are based on the formula [27, 28, 30]; storing the location of optimal nodes so far; evaluating whether the location of nodes meet the stopping condition of optimization, and if so, outputting the most appropriate node, otherwise going back to the second step [29, 30].

5. Simulation Results

Under LoRa wireless communication system, the contradiction of power consumption and real time is well solved, which is of great significance to the development of wireless networks. When using the improved algorithm to optimize wireless communication network, the effect of real time can be optimized to the best. Moreover, the number of nodes is the least. On the one hand, the LoRa wireless network can achieve good communication in a certain area, and, on the other hand, it can meet the requirements of real time and lower power consumption [8, 20, 23].

In order to show the effect of wireless network nodes optimized by the improved artificial colony algorithm, we select different scene areas to test the two kinds of algorithm. Under the same conditions of external influence, we keep the initial parameters of two kinds of algorithm the same in each scenario. Furthermore, the scene with an area of 100\300\500\700\1000\1500 square meters is tested and the power consumption is not limited. In these different areas, the distribution of nodes will have a very intuitive simulation effect, as shown in Table 1.

Table 1: Optimization results of the two algorithms in certain scenarios.

Scene area ($\mathbf{m}^2$)	ABC (n)	RBFABC (n)
$(100 * 100)$	2	2
$(300 * 300)$	5	4
$(500 * 500)$	8	7
$(700 * 700)$	15	13
$(1000 * 1000)$	24	20
$(1500 * 1500)$	42	32

From Table 1, the improved algorithm is very efficient. Moreover, the proposed algorithm can use fewer nodes to cover the same area. Above all, the improved algorithm distributes the nodes more evenly in the same space. In a certain area, the distribution of nodes corresponding to the two kinds of algorithm is shown in Figure 2.

Figure 2 shows that, under different areas, the improved algorithm uses fewer nodes. At the same time, the real-time effect of the improved algorithm is better. In general, the improved algorithm obviously has fewer iterations, higher updating speed, and higher convergence speed. What is more, the wireless network realizes the full coverage of the network with fewer nodes by the improved algorithm. Not only that, we also find more advantages of the improved algorithm after deeper analysis of the power consumption. In different scenes, the power consumption of different number of nodes is shown in Figure 3.

Figure 3 shows that the RBFABC algorithm is obviously lower than the ABC algorithm in terms of power consumption, which shows the superiority of the improved algorithm in wireless network. In most cases, as the area of the scene increases, both the power consumption and the number of nodes also increase. However, we draw a conclusion from Figure 3 that as the number of nodes increases, the power consumption is controlled to a lower level by the improved algorithm. In summary, the RBFABC algorithm solves the contradiction between power consumption and number of nodes better in wireless network.

Correspondingly, to show the superiority of the improved algorithm, a great deal of experiments are done to test the power consumption. For the convenience of data comparison, we still take the scenario area of 100\300\500\700\1000\1500 square meters to test. In the case of certain power, the optimized spatial nodes distribution is shown in Figure 4.

It can be concluded from Figure 4 that, under the same power, the RBFABC algorithm is obviously better than the ABC algorithm in terms of the distribution of the wireless network. Beyond that, fewer nodes cover a wider range by the RBFABC algorithm. To show the superiority of the improved algorithm, we do more tests. Under the same power consumption, the specific number of nodes is shown in Table 2.

According to the data in Table 2, we can draw a conclusion that the RBFABC algorithm is better than the ABC algorithm in terms of optimization effect. Beyond that, the relationship between iteration number and optimal solution is shown in Figure 5.

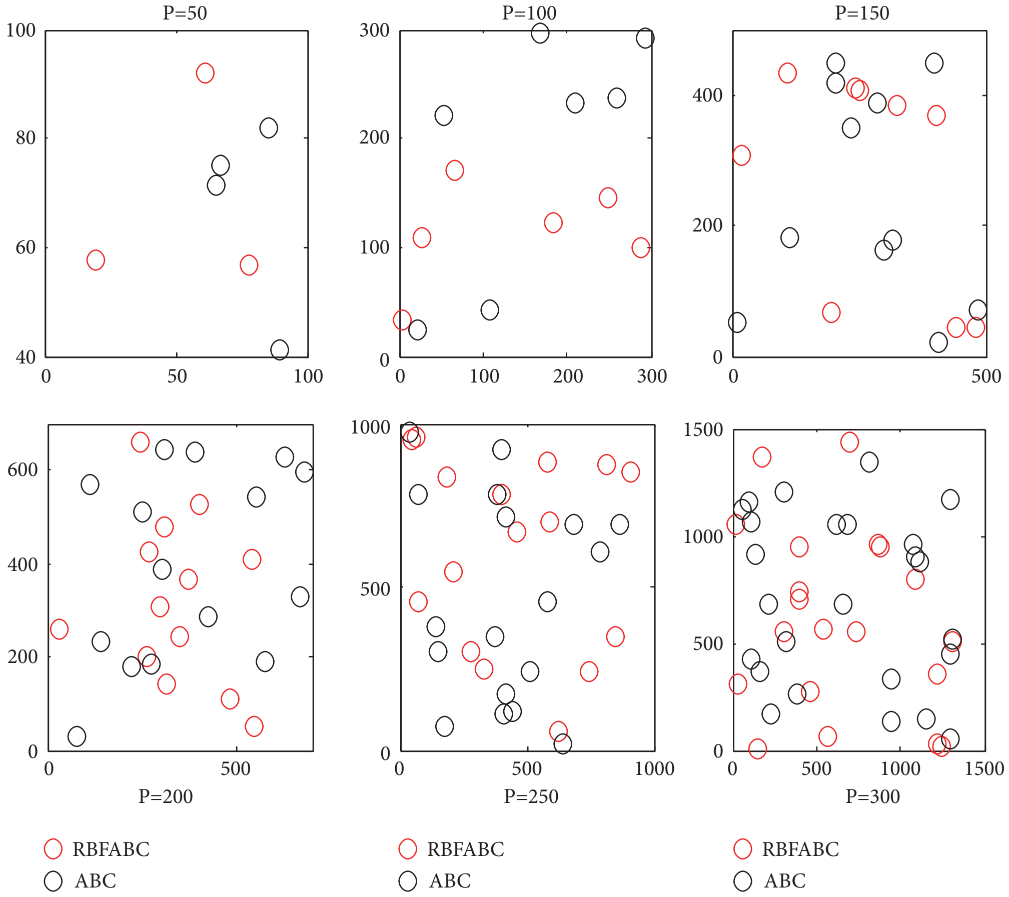


Figure 4: Node distribution under the same power.

Table 2: Optimization results under a certain power.

Power consumption (w)	ABC (n)	RBFABC (n)
(50)	4	3
(100)	7	6
(150)	11	9
(200)	15	13
(250)	18	16
(300)	25	20

Table 3: Comparison of multimedia data transfer speed.

Number of nodes (n)	ABC (kb/s)	RBFABC (kb/s)
(5)	200	180
(10)	400	350
(15)	620	560
(20)	810	780
(25)	980	850
(30)	1200	990

Figure 5 shows that the improved algorithm can find the optimal solution faster. Therefore, the improved algorithm has more advantages for the optimization of wireless network nodes. Moreover, in a certain area, when the number of network nodes is the same and other factors are consistent, the multimedia data transmission speed which is optimized by two kinds of algorithms is shown in Table 3.

From Table 3, the improved algorithm is obviously superior to the original algorithm in terms of transmission speed. As the number of nodes increases, the data transmission speed becomes faster and faster. In order to display the data visually, the comparison of speed trend is shown in Figure 6.

It is seen from Figure 6 that the RBFABC algorithm is faster than the ABC algorithm in terms of multimedia data transmission speed. Moreover, as the number of network nodes increases, the data transmission speed gap is bigger and bigger. In general, the RBFABC algorithm is better than the ABC algorithm. Above all, the real-time performance of the system is greatly improved. In addition to these, we do a lot deeper analysis. Under the same number of network nodes, the comparison of network delay is shown in Figure 7.

It can be concluded from Figure 7 that, under the same network nodes, the RBFABC algorithm is obviously lower than the ABC algorithm in terms of network delay. On the other hand, we compare the maximum data load of the single data transmission in the network system. The analysis results of two kinds of algorithm are shown in Figure 8.

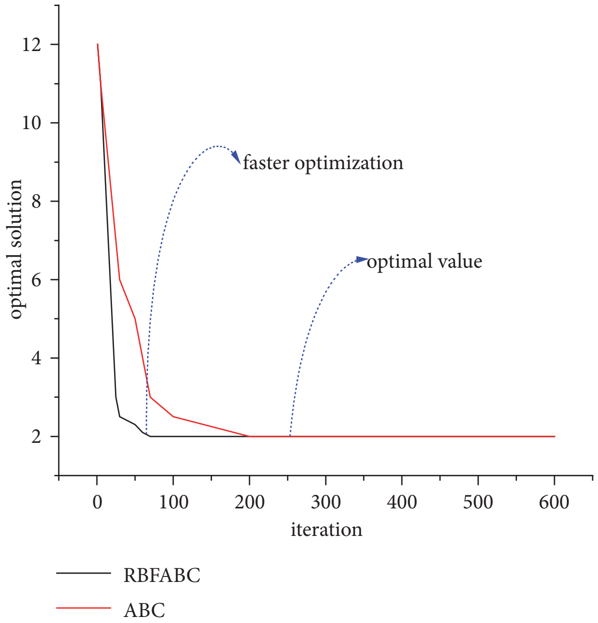


Figure 5: The relationship between iteration number and optimal solution.

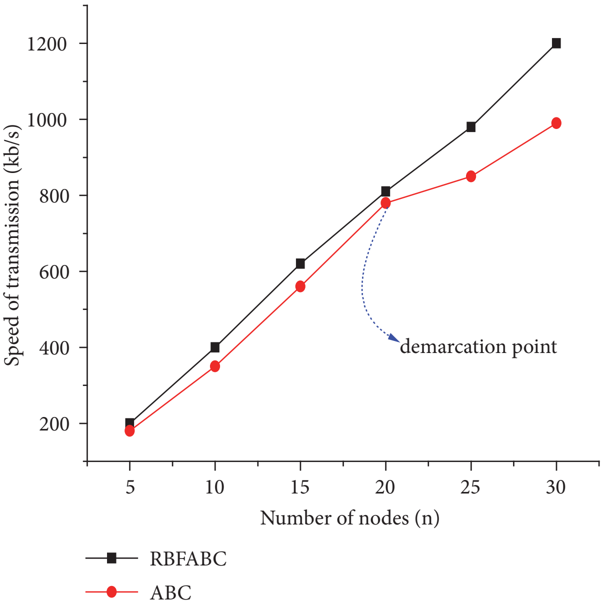


Figure 6: Comparison of speed trend.

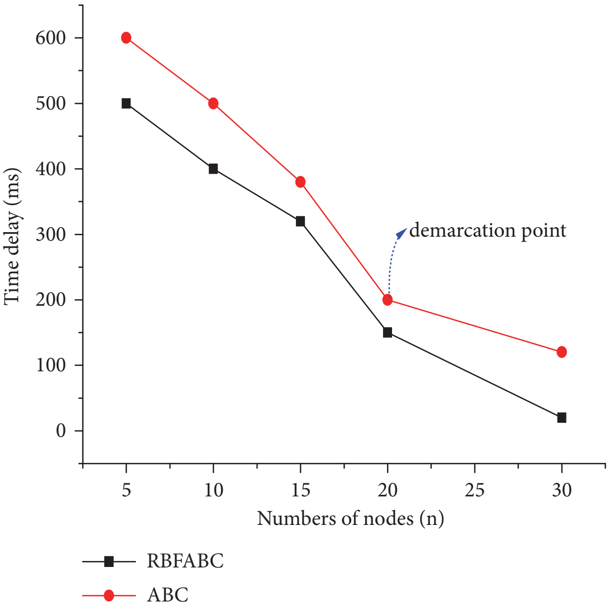


Figure 7: Comparison of network delay.

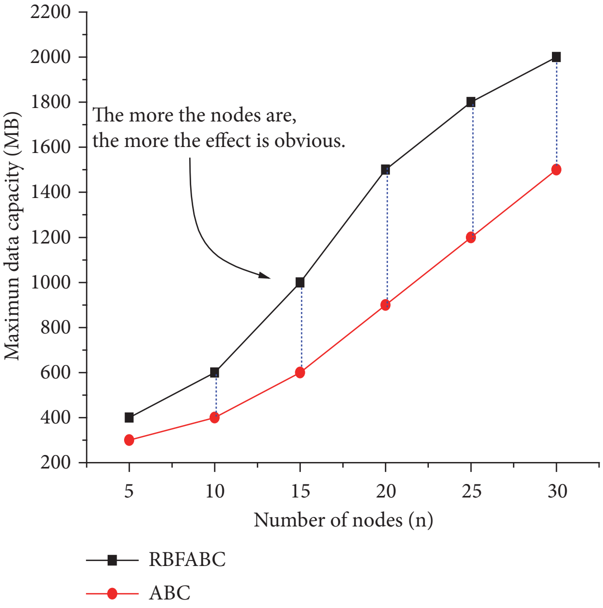


Figure 8: Maximum load comparison.

It is seen from Figure 8 that the RBFABC algorithm is significantly higher than the ABC algorithm in terms of the maximum data load of the single data interaction. In general, the improved algorithm has more advantages.

6. Conclusion

In this paper, to resolve the contradiction of base station establishment in LoRa wireless communication systems between the power consumption of LoRa nodes and real-timeliness of the mesh network, we have proposed a novel RBFABC algorithm which achieves better performance by embedding RBF radial basis neural network training with random gradient method. Extensive simulations have been conducted and the results have shown the obvious superiority of the RBFABC algorithm in terms of the high efficiency and real-timeliness of multimedia transmission.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Intelligent Manufacturing Project of the Ministry of Industry and Information Technology: Industrial Internet Data Mutual Recognition Research–Low-Power Message Distribution, the National Natural Science Foundation of China under grants 71602124 and 61773187, and Liaoning Provincial Natural Science Foundation of China under grant 20170540662.

References

[1] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[2] H. Shah, T. Herawan, R. Naseem, and R. Ghazali, "Hybrid Guided Artificial Bee Colony Algorithm for Numerical Function Optimization," in *Advances in Swarm Intelligence*, vol. 8794 of *Lecture Notes in Computer Science*, pp. 197–206, Springer International Publishing, Cham, 2014.

[3] Q. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

[4] S. kuchlin and P. Jenny, "Automatic mesh refinement and parallel load balancing for Fokker-Planck-DSMC algorithm," *Journal of Computational Physics*, vol. 363, pp. 140–157, 2018.

[5] X. Ma, S. Xu, F. An, and F. Lin, "A Novel Real-Time Image Restoration Algorithm in Edge Computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3610482, pp. 1–13, 2018.

[6] P. Wang and B. Henz, "Efficient approaches to resource allocation in MIMO-based wireless mesh networks," in *Proceedings of the 14th Annual Wireless Telecommunications Symposium, WTS '15*, vol. 39, pp. 1–7, USA, April 2015.

[7] F. Lin and X. Lü, "QoS guaranteed pre-pushing scheme in peer-assisted streaming network," *China Communications*, vol. 11, no. 14, pp. 111–117, 2014.

[8] J. Wellons and Y. Xue, "The robust joint solution for channel assignment and routing for wireless mesh networks with time partitioning," *Ad Hoc Networks*, vol. 13, pp. 210–221, 2014.

[9] H. Chaoui, M. Khayamy, and O. O. Okoye, "Adaptive RBF Network Based Direct Voltage Control for Interior PMSM Based Vehicles," *IEEE Transactions on Vehicular Technology*, 2018.

[10] W. Zhuang and M. Ismail, "Cooperation in wireless communication networks," *IEEE Wireless Communications Magazine*, vol. 19, no. 2, pp. 10–20, 2012.

[11] J. Si, H. Huang, Z. Li, B. Hao, and R. Gao, "Performance analysis of adaptive modulation in cognitive relay network with cooperative spectrum sensing," *IEEE Transactions on Communications*, vol. 62, no. 7, pp. 2198–2211, 2014.

[12] M. Benedetto, K. Johansson, and M. Johansson, "Industrial control over wireless networks," *International Journal of Robust & Nonlinear Control*, vol. 20, no. 2, pp. 119–127, 2018.

[13] C. M. Stefanovic, "LCR of amplify and forward wireless relay systems in general alpha-Mu fading environment," in *Proceedings of the 2017 25th Telecommunication Forum (TELFOR)*, pp. 1–6, Belgrade, November 2017.

[14] D. Prasad, V. Koneri, and K. Shivakumar, "Mitigating wireless network interface card energy consumption in mobile devices," *Research & Technology in the Coming Decades*, vol. 2014, no. 1, pp. 103–114, 2014.

[15] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks," *Wireless Networks*, vol. 16, no. 5, pp. 1493–1510, 2010.

[16] J. Wu, X. Qiao, Y. Xia, C. Yuen, and J. Chen, "A low-latency scheduling approach for high-definition video streaming in a heterogeneous wireless network with multihomed clients," *Multimedia Systems*, vol. 21, no. 4, pp. 411–425, 2015.

[17] M. Ghaderi, D. Goeckel, A. Orda, and M. Dehghan, "Minimum Energy Routing and Jamming to Thwart Wireless Network Eavesdroppers," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1433–1448, 2015.

[18] S. Dasgupta, G. Mao, and B. Anderson, "A New Measure of Wireless Network Connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1765–1779, 2015.

[19] X. Cheng, Q. Wang, Q. Wang, and D. Wang, "A high-reliability relay algorithm based on network coding in multi-hop wireless networks," *Wireless Networks*, pp. 1–10, 2017.

[20] Y. E. Sagduyu, Y. Shi, A. Fanous, and J. H. Li, "Wireless network inference and optimization: Algorithm design and implementation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 257–267, 2017.

[21] J. Liu and X. Wu, "Supervision and control based on rbf neural network," *Computer Knowledge Technology*, vol. 2018, pp. 1–13, 2018.

[22] H. Qin, L. Shen, C. Sima, and Q. Ma, "RBF Networks with Dynamic Barycenter Averaging Kernel for Time Series Classification," in *Artificial Intelligence*, vol. 888 of *Communications in Computer and Information Science*, pp. 139–152, Springer Singapore, Singapore, 2018.

[23] P. Wang, "The Application of Radial Basis Function (RBF) Neural Network for Mechanical Fault Diagnosis of Gearbox," *IOP Conference Series: Materials Science and Engineering*, vol. 269, pp. 012–056, 2017.

[24] J. Dong and L. Chen, "The nonlinear integral sliding mode of RBF neural network algorithm is used to control the motion trajectory error of the manipulator," *Chinese Journal of Construction Machinery*, vol. 2018, no. 17, pp. 11–22, 2018.

[25] Y. Ma and C. Na, "Switch fault diagnosis system based on cost sensitive RBF neural network," *Railway Computer Application*, vol. 2018, no. 13, pp. 45–53, 2018.

[26] H. Du, E. Zhao, K. Guo, and H. Zhing, "Safety monitoring model of dam service based on genetic algorithm and RBF neural network," *Journal of China Three Gorges University*, vol. 2018, no. 18, pp. 15–27, 2018.

[27] D. Zaborski, W. Grzesiak, and M. Szewczuk, "An application of radial basis function (RBF) networks to daily body weight gains prediction in the indigenous Harnai Sheep of Pakistan," *Publication Preview Source Biotechnologia Problemy Wyzwania Sczecin Poland*, vol. 14, no. 18, pp. 67–87, 2018.

[28] D. Karaboga and B. Akay, "A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3021–3031, 2011.

[29] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[30] J.-J. Forneron and S. Ng, "The ABC of simulation estimation with auxiliary statistics," *Journal of Econometrics*, vol. 205, no. 1, pp. 112–139, 2018.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 1413026, 9 pages
https://doi.org/10.1155/2018/1413026

Research Article

PO-MPTCP: Priorities-Oriented Data Scheduler for Multimedia Multipathing Services

Wei Lu,[1] Dandan Yu,[1] Minghe Huang,[1] and Bin Guo[2]

[1] *School of Software, Jiangxi Normal University, Nanchang 330022, China*
[2] *Information Office, Jiangxi Normal University, Nanchang 330022, China*

Correspondence should be addressed to Wei Lu; weilu@jxnu.edu.cn

Received 9 August 2018; Revised 18 October 2018; Accepted 29 October 2018; Published 2 December 2018

Guest Editor: Jiyan Wu

With the diversified wireless network access technology and large-scale equipment of multinetwork interface devices expanding, network transmission performance for multi-homed terminals has been widely concerned by academic circles. More and more scholars have paid attention to Multipath Transmission Control Protocol (MPTCP) as one of the representative methods for studying path transmission performance. However, their studies ignore the impact of dynamic network environments on data transmission performance and seldom consider the priority of data transmission. Undoubtedly, not prioritizing packets will have a dramatic effect on the users experience in the heterogeneous networks. In this paper, we propose a novel priority-aware streaming media multipath data scheduler mechanism (PO-MPTCP) to achieve the following goals: (1) detecting the priority of all streaming media data; (2) achieving multiattribute-aware path evaluation and switching mechanism; (3) introducing a path-quality priority-driven data distribution mechanism to improve streaming multipath transmission performance. The simulation experiment shows that PO-MPTCP proposed by this paper improves the transmission performance of streaming media and reduces the transmission delay. For the result of simulation experiment, it is easy for us to find that PO-MPTCP is more efficient in data delivery than the standard MPTCP mechanism.

1. Introduction

Under the background of establishment of various wireless access technologies (i.e., WiFi, 4G, Bluetooth, etc.), the number of mobile terminals equipped with several different standard network interfaces is increasing year by year. Multi-homed devices can use multiple network interfaces to transmit data in parallel, effectively improving network transmission throughput [1]. Due to the growing demands for bandwidth requirement and transmission rate requirement of real-time streaming applications (i.e., Facebook [2], WeChat [3], YouTube [4], etc.), traditional single-path Transmission Control Protocol (TCP) [5] cannot utilize the characteristics of the terminals multinetwork interfaces; thus the Internet Engineering Task Force (IETF) proposes Multipath TCP (MPTCP) to be compatible with the current application layer based on traditional TCP [6, 7].

As an extension of TCP, MPTCP is compatible with applications, and it never needs to make too many modifications for traditional TCP sockets to achieve application layer transparency. MPTCP's main idea is that data streams would be parallel distributed to multiple links, and it will improve the performance of data transmission (i.e., improving network throughput, reducing transmission delay, maximizing the utilization of the network, etc.) [8]. From Figure 1, we can see the MPTCP-based multimedia multipath transmission in wireless network. It simply illustrates that mobile devices use three paths (*Paths A, B, and C*) to correspond to the MPTCP-based media server.

By adopting a multi-homing structure, MPTCP is considered to be a promising transmission technology, which can meet the requirements for specific Quality of Services (QoS) and balance real-time task utilization rate in network resources [9, 10]. However, most MPTCP data scheduling algorithms and congestion control algorithms only consider transmission efficiency and fairness [7–9] and ignored the priority difference of data transmission caused by the complexity of data types [11].

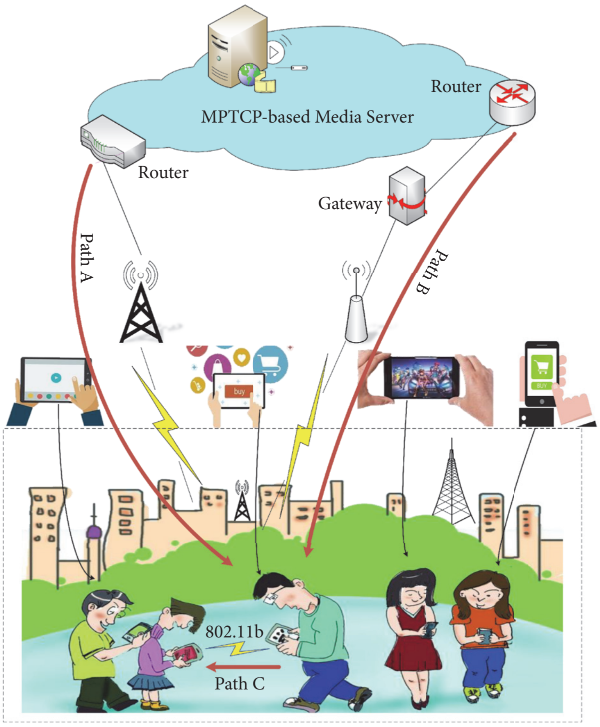


Figure 1: MPTCP-based multimedia multipath transmission in wireless network.

Inspired by the above considerations, this paper presents a priorities-oriented data scheduler for multimedia multipathing mechanism that allows applications to distinguish the relatively prioritized data, ensure higher-priority data can be transmitted via a good quality path, improve the transmission performance of higher-priority data, reduce the transmission delay, and enhance the overall transmission performance for real-time streaming media services. The simulation results confirm that this mechanism can effectively improve the transmission efficiency. PO-MPTCP's fundamental contributions are as follows:

(1) It gives MPTCP the ability to perceive the priority of streaming media data and the ability to know when and how to call priority scheduling services.

(2) It designs a multiattribute perceptual mechanism to estimate path qualities and design a multiattribute cooperative path switching algorithm.

(3) It introduces a data-priority-driven distribution strategy based on path quality.

The remainder of this paper is structured as follows. In Section 2, we briefly introduce the relevant works of our PO-MPTCP. Section 3 introduces the details of PO-MPTCP. In Section 4, we evaluate and analyze the performance of PO-MPTCP based on the experimental results. Section 5 gives summaries of this paper and our future work.

2. Related Work

MPTCP is defined as a development of the TCP that can be compatible with existing Internet devices and application interfaces and which still receives great attention from academic circles. Many scholars are devoted to optimizing the data scheduling algorithm of MPTCP. Cao et al. [12] proposed a receiver-driven data scheduling strategy based on MPTCP, and the purpose is to transfer some operations (e.g., data scheduling and path selection) from the sending end to the receiving end and balance the load of sender and receiver. Marikoshi et al. [13] introduced an improved method for rapidly received ACK based on MPTCP which applies the available increase of ACK to enhance throughput. Kim et al. [14] proposed a method for managing MPTCP (RBPM) based on the size of the receive buffer. The main operation is to estimate the disorder packets by using the available receiving buffer size and multipath dissimilar characteristics; that is, if congestion occurs, RBPM would immediately stop the bad transmission path. Due to capturing the quality of the path correctly, Chung et al. [15] described a new path management scheme on the basis of Machine Learning on MPTCP (MPTCP-ML), which used signal strength, throughput, and interfering APs number to estimate path quality. Kimura et al. [16] discussed three data scheduling mechanisms: (1) data scheduling based on the highest sending rate, (2) data scheduling based on the largest sending window, and (3) data scheduling based on the lowest sending delay.

At the same time, the optimization problem of streaming media data distribution with MPTCP has caused widespread exploration in the academic community. Hayes B et al. [17] used HyperText Transfer Protocol 2.0 adaptive bitrate video transmission based on the MPTCP architecture to eliminate network bandwidth and estimate buffer status by using three modular push strategies for MPTCP. The goal is to improve the transmission of video content in a loss environment, enabling the delivery of adaptive bit rate video. Xu et al. [18] extended the partial reliability protocol of MPTCP to provide a flexible QoS trade-off for multimedia applications in real-time and reliability. Cao et al. [19] introduced the concepts of partial reliability and real-time constraints in MPTCP and discussed the real-time constraints of multimedia streams and the partial reliability of TCP.

In many current MPTCP researches, we found that the focus of scholars research has begun to shift from improving data schedules and congestion control algorithms [20–23] to optimizing real-time streaming media services. But these optimizations do not take into account the priority of real-time streaming applications for different types of transmission data, and the existing literature on data-priority scheduling is almost limited to the network layer and wireless sensor network. As described in [24], a Markov model is used to control the flow of different priorities on sensor nodes. Therefore, we propose a priority-oriented streaming media multipath data transmission mechanism for developing future mobile streaming media applications and providing attractive benefits including transfer performance improvement and high-quality users experience provisioning.

3. PO-MPTCP Detail Design

As an extension of TCP, MPTCP has many attractive benefits for content-rich real-time multi-stream media in terms

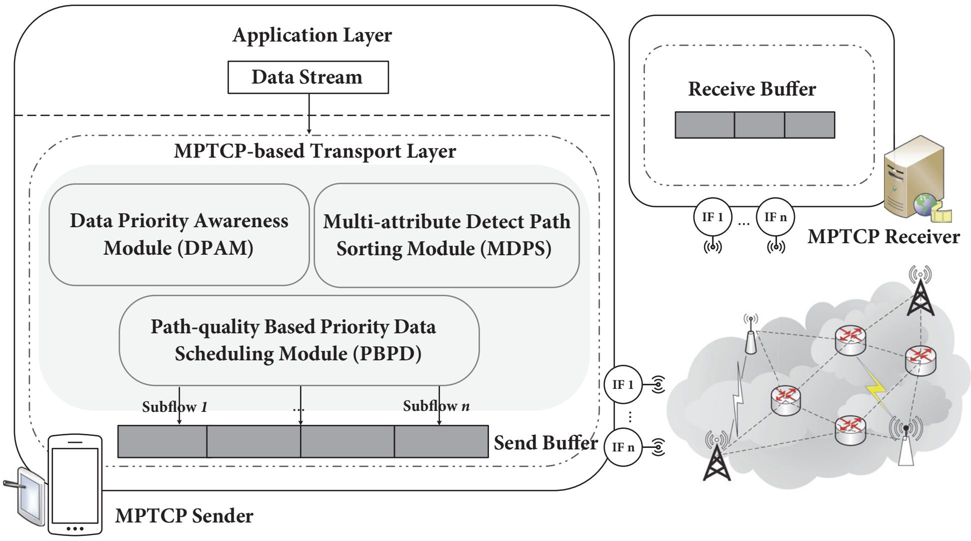

Application Layer
Data Stream
MPTCP-based Transport Layer
Data Priority Awareness Module (DPAM)
Multi-attribute Detect Path Sorting Module (MDPS)
Path-quality Based Priority Data Scheduling Module (PBPD)
Subflow 1
...
Subflow n
Send Buffer
IF 1
IF n
MPTCP Sender
Receive Buffer
IF 1
IF n
MPTCP Receiver

FIGURE 2: Architecture of the priority-oriented MPTCP data transmission strategy.

of performance improvement, bandwidth aggregation, and users experience. But these performances might be affected by many factors, such as path characteristics and data scheduling strategies, when the limited receive buffer with a strict sorting. Every wrong grievance is assumed to result in buffer congestion or serious MPTCP performance decline. In order to meet bandwidth requirements and transmit rate for the services of bandwidth-intensive and real-time-content-rich streaming, we discuss the theory about adding priority to MPTCP data scheduling which offers good probabilities including data scheduling dispatch, throughput improvement, and important data transmission-quality assurance.

In this paper, we define the priority of the data subflow as an additional service for the application, and the scheduling policy takes into account real needs of the network environment. According to the data submission order and currently available network bandwidth, the scheduling policy can allow application to determine the data priority and deliver other data sequentially; that is, the vital data should be prioritized and transmitted using the best quality link path. In this data scheduling strategy, it still guarantees relatively important data to be given priority for submitting, reduces transmission delays, and increases the efficiency of whole application when the network service quality is poor.

Figure 2 shows the system architecture for the priority-oriented MPTCP data transmission strategy, which mainly includes MPTCP's sender, MPTCP's receiver, and multipath heterogeneous wireless network. The sender includes three modules: (1) Data Priority Awareness Module (DPAM), (2) Multiattribute Detect Path Sorting Module (MDPS), and (3) Path-quality Based Priority Data scheduling Module (PBPD). If application layer has data to send, a module of sender is similar to SCTP multiflow buffer module which transfers data from application layer to transport layer, and it uses DPAM to detect and record the priority information for the delivery data, utilize MDPS to determine the status of each subpath based on the RTT and CWND values, and sort the paths according to the quality of each path. Priorities-oriented data scheduler for multimedia multipathing services will dispatch data packets according to data priorities and path qualities; it also transmits information to network layer and finally gets to the receiver. In this way, when the receiver gets the data packets, it will restructure them in sequence, transfer them to the upper layer of the receiver, and feedback the SACK to the sender.

3.1. Data Priority Awareness Module (DPAM). In mobile Internet, data is developing to large scale and diversity which leads to multimedia applications demands for network bandwidth and data specialization is increasing. While the traditional MPTCP transmission method neglects the application's priority for messages, it just treats information equally as byte streams and results in data transmission blindness. Therefore, we have to establish a module to detect the data priorities.

In order to differentiate the data priorities, we will mark the different priority missions in the network. When the sender starts to transmit data, DPAM will be through the cross-layer cooperative communication [25], retrieve priority values corresponding to each stream, and store them in the status list (denoted by PO_{list}). Each element of PO_{list} is composed of a doublet (i.e., index of subflow i, priority of subflow PO_{D_i}). Algorithm 1 shows pseudocode of DPAM.

3.2. Multiattribute Detect Path Sorting Module (MDPS). The data stream transmission efficiency is proportional to the path quality in the ideal case, but the current environment of the heterogeneous network is very complex, so we hope all of data can choose a relatively good quality subpath for transmitting. In the priority-oriented data transmission protocol introduced by this paper, we assume that relatively important data will have a higher priority that can be transmitted through a relatively reliable path, which can reduce the loss probability of important packet. In addition, we propose a Multiattribute Detect Path Sorting Module (MDPS) which can determine the status of each subpath based on the RTT

```
Definition:
    $D_i$: the $i$th data stream within the MPTCP session.
    $PO_{D_i}$: the priority value of data stream $D_i$
    $PO_{list}$: the status list for each stream
When there are some data stream need to send,
for all data stream $D_i$ within a MPTCP session do
        $PO_{list} = (i, PO_{D_i})$
end for
```

Algorithm 1: DPAM-based priority information collection algorithm.

and CWND values and sort the path according to the quality of the path.

To deliver the prepared data, we assume that there are n available paths in an MPTCP, MDPS can record the qualities of each available path (RTT and CWND) according to the MPTCP connection quality, and uniformly storage is estimated in the set list (denoted by Q_{list}). According to the estimates of RTT and CWND in the Q_{list}, MDPS will give priority to supporting the lowest RTT and sort paths in ascending order based on the RTT values (except that the path's RTT values are the same, it will be followed by the descending order of the CWND values). MDPS will sort all available paths by quality which is used for MPTCP data scheduler. The pseudocode of MDPS is shown in Algorithm 2.

3.3. Path-Quality Based Priority Data Scheduling Module (PBPD). On the basis of DPAM and MDPS, PBPD will try to support that packets arrived in order. When the sending window is idle and n packets would be sent, we prefer to use the subpath with a good path quality to schedule important data. That is, as long as there is important data with prior in queue, it would be quickly removed from queue and utilize good priority subpath for transmission. It ensures that high-priority data in the queue can be transmitted by the best-quality path and achieve the lowest delay at the same time. When the sending window is idle and n data packets would be sent, PDPB will implement steps as follows:

(a) According to PO_{list} (priority collection of data) produced by DPAM, PDPB will determine whether the current data packet needs priority scheduling. If PO_{S_i} is NULL, the standardized MPTCP will be used for data distribution. Otherwise, it indicates that the application has priority packet, then PBPD would be dispatched.

(b) According to the sort path formed by the MDPS module, PBPD will use the first path of P_{list} set as the candidate path to be given priority to important data.

(c) PDPB will take the next path of P_{list} as the candidate path when the path *cwnd* is full.

(d) Repeat all above three steps until there is no available sending window on the sender; that is, n data have already been sent. Algorithm 3 shows the pseudocode for PBPD.

Table 1: Path configuration used in the simulation.

Network Parameters	Path *A*	Path *B*
Wireless technology	WiFi/IEEE 802.11b	WiMax/IEEE 802.16
Access link bandwidth	11Mbps	10Mbps
Access link queue type	Droptail	Droptail
Uniform loss rate	0-5%	0-5%

4. Performance Evaluation

4.1. Simulation Topology. In this section, we utilize the popular network simulator NS-2 [26, 27] to verify and evaluate the performance for our proposed priorities-oriented data scheduler algorithm and compare with classical MPTCP. In order to build a real and reliable MPTCP simulation environment, we embed MPTCP patch into NS-2 and then apply PO-MPTCP algorithm to the current MPTCP's simulator.

According to our previous work [6], we can find that the total Internet traffic is made up by 80%-83% of TCP traffic and 17%-20% of UDP, and the suddenness of the network background traffic increases the difficulty for deployment of MPTCP in the network. For simulating a more real network environment we will configure four TCP traffic generators and one UDP traffic generator in every router to get the 80% of TCP traffic and 20% of UDP traffic. The details are shown in Figure 3.

In Figure 3, the simulation topology forms MPTCP's sender and MPTCP's receiver, and it simultaneously fits two links together (denoted by Paths A and B) that have their own corresponding network parameters, as shown in Table 1. For simulating loss-of-frame at the network link layer, a random unified model is used for each path to set the packet loss rate which is caused by random connection competition or radio interference. At present, the bandwidth of Paths A and B is set to 11 Mbps and 10 Mbps, respectively, and Path A adopts the standard interface WiFi/IEEE802.11b, while Path B uses WiMax/IEEE 802.16 interface. In addition, the network simulation time of each path is set to 60s, and other experimental parameters are default as the values of NS-2 network simulator.

4.2. Simulation Analysis. For convenience, we denote the data result of the standard MPTCP transmission mechanism as "MPTCP" and define priority-oriented data multipath streaming media transmission mechanism proposed by us as "PO-MPTCP". In this section, we use a random unified model to uniform loss errors and take 0%-5% as the lossy rate of each path for analyzing the average throughput, average delay, jitter, and peak signal-to-noise ratio (PSNR) changes in the packet loss rate experiment.

(1) Average Throughput. In Figure 4, we can see the average throughput of standard MPTCP and PO-MPTCP at different packet loss rates. In lossy heterogeneous wireless environment, four simulation results show the influence of different packet loss rates on multimedia stream throughput, and PO-MPTCP can achieve a more reliable data transmission. When setting the same packet loss rate, PO-MPTCP achieves

```
Definition:
        $PA_i$: the $i$th path within the MPTCP session.
        $RRT_{PA_i}$: the RTT estimation value of path $PA_i$
        $CWND_{PA_i}$: the CWND estimation value of path $PA_i$
        $S_i$: the status information for $i$th path.
        $Q_{list}$: the quality list for each path
        $P_{list}$: a preferred path list selected by MDPS
for all path $PA_i$ within a MPTCP session do
        $S_i = (i, RTT_{PA_i}, CWND_{PA_i})$
        put $S_i$ into $Q_{list}$
end for
for all paths $S_i$ within the $Q_{list}$ do
    all paths are sorted by $RRT_{PA_i}$ values in an ascending order
    if their $RRT_{PA_i}$ values are same then
        sort the paths in a descending order basing on their $CWND_{PA_i}$ values
    end if
    set $k = S_i \longrightarrow$ subflow index
    put $P_k$ into $P_{list}$
    $k = k + 1$
end for
```

Algorithm 2: MDPS-based path information collection and sorting algorithm.

```
Definition:
        $PO_{list}$: the priority of data stream list perceived by DPAM
        $PO_{D_i}$: the priority information for $i$th data stream
        $P_{list}$: a preferred path list obtained by MDPS
        $P_{list}(0)$: the first path of $P_{list}$
        $P_{send}$: a candidate path used for prioritized data delivery
When the send buffer is idle and can send packets,
for all $PO_{D_i}$ within the $PO_{list}$ do
    if $PO_{D_i}$==1 then
        there is a prioritized packet to send
        set $P_{send}$=$P_{list}(0)$
        while the cwnd of $P_{list}(0)$ is full do
            set $P_{send}$=$P_{list}(0) \longrightarrow$ next
        end while
    else then
        using standard MPTCP data schedule to distribute
    end if
    schedule the packet over $P_{send}$
end for
```

Algorithm 3: PBPD-based data scheduling algorithm.

better average throughput than standard MPTCP with 2.28%, 8.18%, 2.69%, and 8.04% improvements, respectively. The standard MPTCP does not consider two factors: (1) the requirements of specific QoS on multimedia application; (2) the priorities of the application data and the qualities of different transmission paths in the heterogeneous network. These factors cause earlier packets which are in the MPTCP receiver buffer out of order. With the increase in the number of packets, it can cause buffer congestion and network transmission throughput reduction. Our proposed PO-MPTCP considers the data priorities and the path qualities and selects the best link to transmission; it effectively improves the transmission performance of MPTCP.

(2) Average Delay. As an important parameter for measuring network performance, average delay is often used by researchers to determine the transmission rate of a path when it comes to researching related network protocols. Aiming to highlight the advantage for the PO-MPTCP, Figure 5 compares the average delay of PO-MPTCP and standard MPTCP with 0-5% packet loss rate. For the purpose of conveniently displaying the end-to-end transmission delay, we take the packet loss rate as an example of 0.01, as shown

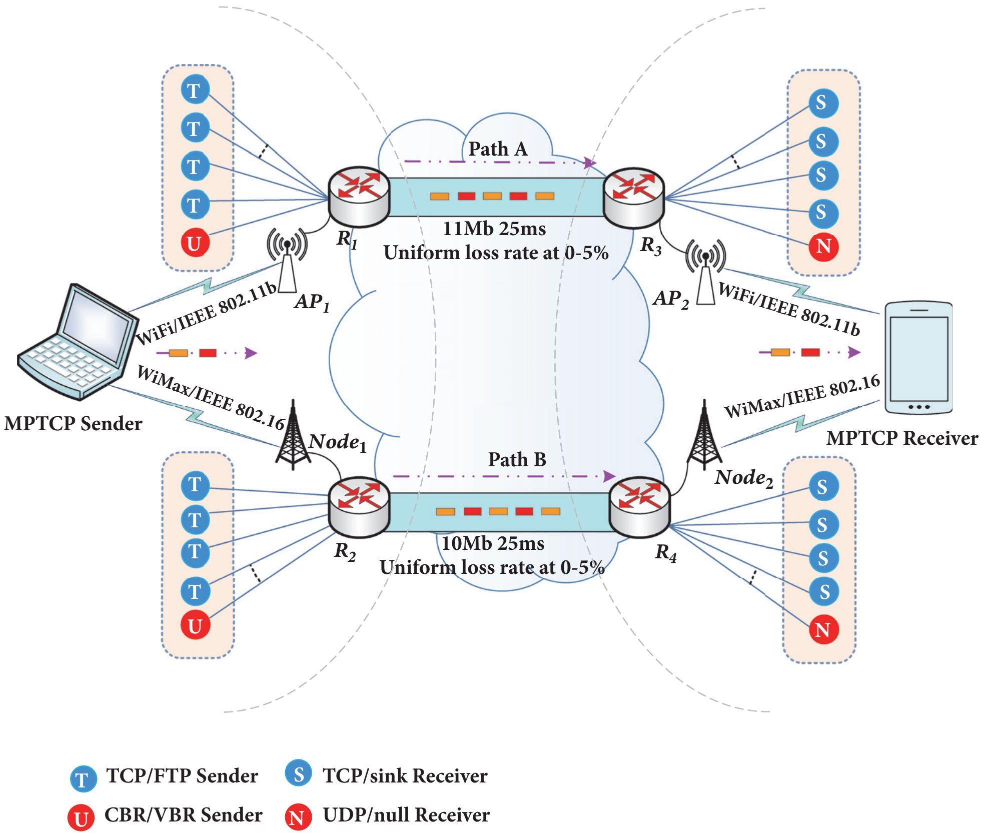


FIGURE 3: MPTCP-based simulation topology.

in Figure 6, which shows PO-MPTCP is better than the standard MPTCP. Therefore, PO-MPTCP can improve the transmission performance and reduce the transmission delay through priority-aware data scheduling policies and link sequencing.

(3) Jitter. As far as we know, jitter is an important indicator for predicting the stability of streaming media packets, and it is usually achieved by measuring the last value of delay and the function value of packet length. We take the loss rate of 0.02 to explore the change of jitter within 60 seconds. Figure 7 illustrates that the standard MPTCP generates out-of-order packets more frequently than PO-MPTCP. PO-MPTCP can dynamically be aware of data's priority and adaptive switching path, which can better improve the transmission performance of streaming media and ensure the stability of multimedia streaming.

(4) Peak Signal-to-Noise Ratio. Streaming media services will become the core business of the future mobile Internet. So we studied the transmission performance of multimedia in the heterogeneous streaming media simulation. The performance is measured by PSNR [28] value which translates network transmission throughput and loss rate into user-perceived quality through the following equation.

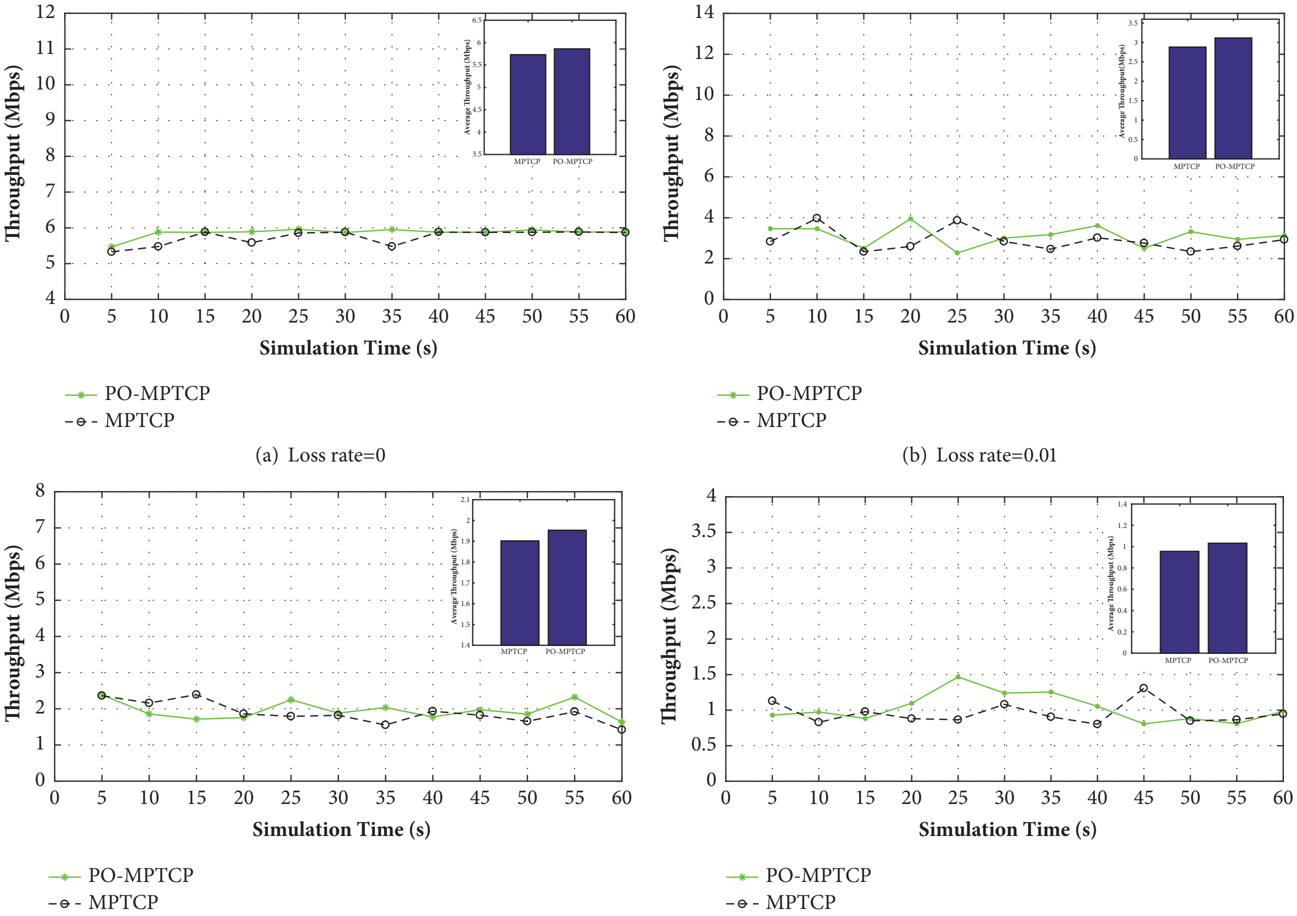
$$PNSR = 20\log_{10}\left(\frac{MAX\_Rbit}{\sqrt{(E\_Th - A\_Th)^2}}\right), \qquad (1)$$

where *MAX_Rbit* represents the rate of the multimedia subflow transmission. *E_Th* denotes expected average throughput in network while *A_Th* is actual average throughput.

Figure 8 demonstrates that the PNSR of PO-MPTCP is 3.7% higher than the standard MPTCP in the heterogeneous wireless network. By analyzing the user quality experience with PSNR, PO-MPTCP can achieve a better transmission quality of streaming media and the PSNR fluctuation is more stable. Hence, compared to the existing mechanisms, PO-MPTCP can improve the user experience.

5. Conclusion

This paper presents PO-MPTCP which is a priorities-aware packet scheduler solution for multimedia multipathing services. It makes full use of its ability to identify packet priority and estimate and switch the path based on the multiattribute perception. In addition, PO-MPTCP provides a priority-driven data distribution strategy to distinguish data priority and ensure that high priority data is transmitted on the best quality path. PO-MPTCP improves the efficiency data delivery by utilizing the path-based priority-aware method. Simulation results show that PO-MPTCP provides attractive benefits in comparison with classical MPTCP including multimedia average throughput improvement, end-to-end delay reduction, and higher-stability multimedia transmission. In particular, we utilize PSNR to analyze the users service experience of PO-MPTCP, which shows that PO-MPTCP can better meet the transmission quality requirements of



(a) Loss rate=0

(b) Loss rate=0.01

(c) Loss rate=0.02

(d) Loss rate=0.05

Figure 4: Comparison of throughput when (a) loss rate=0, (b) loss rate=0.01, (c) loss rate=0.02, and (d) loss rate=0.05.

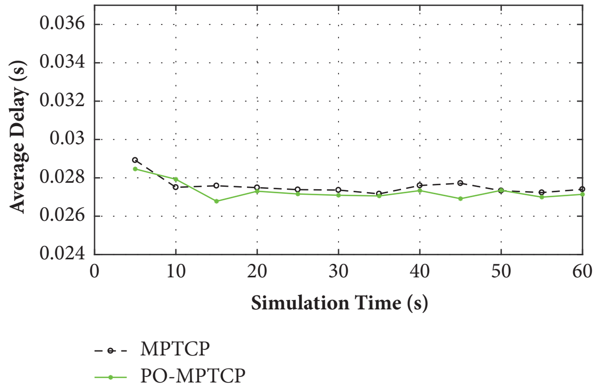


Figure 5: Comparison of delay.

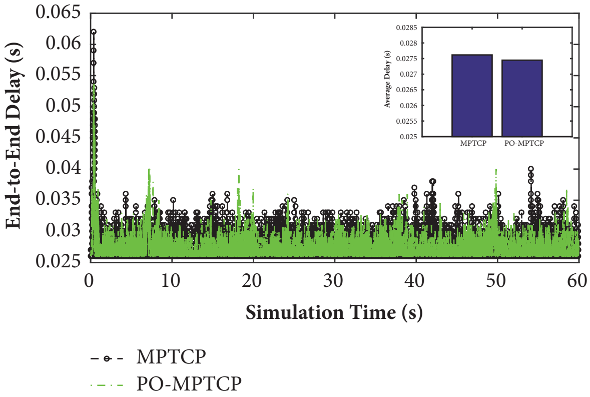


Figure 6: Comparison of end-to-end delay.

streaming media. Our future work will concentrate on deploying PO-MPTCP in actual systems, enabling PO-MPTCP to satisfy high bandwidth requirements for multimedia applications and delay intolerance. In addition, we have studied the practical significance of priority-based queue management, and it provided a theoretical basis for the deployment of MPTCP protocol in future wireless networks.

Data Availability

The research data used to support the findings of this study was simulated by NS-2. The data which the other researchers want to get are available from the corresponding author upon request.

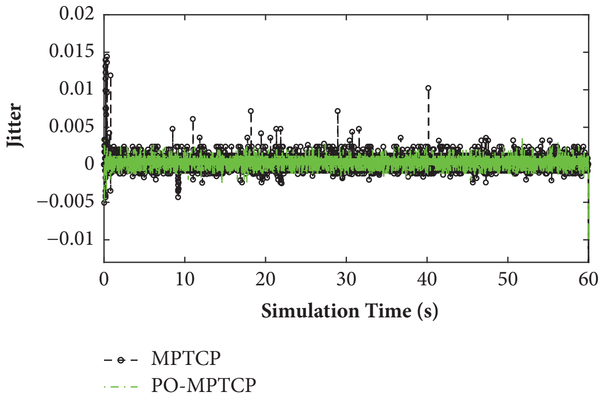


Figure 7: Comparison of jitter.

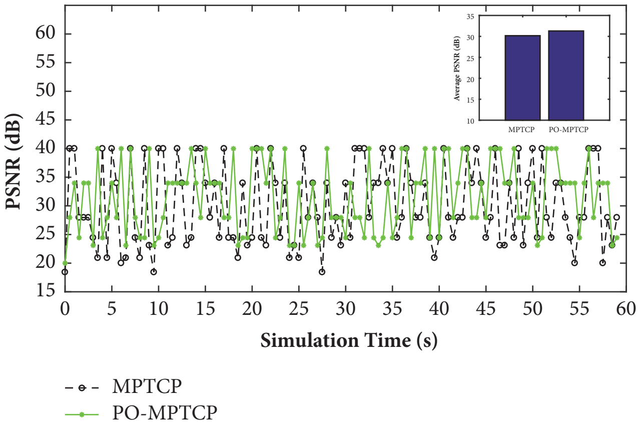


Figure 8: Comparison of streaming media PSNR.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no. 61562044; the Natural Science Foundation of Jiangxi Province under grant no. 20171BAB212014; the Project of Soft Science Research Plan of Jiangxi Province under grant no. 20161BBA10010; and the Jiangxi Province Graduate Innovation Project under grant no. YC2017-S144.

References

[1] J. Wu, B. Cheng, C. Yuen, Y. Shang, and J. Chen, "Distortion-aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 688–701, 2015.

[2] "Facebook," https://www.facebook.com/.

[3] "WeChat," https://www.wechat.com/.

[4] "YouTube," https://www.youtube.com/.

[5] X. Liu, D. Shan, R. Shu, and T. Zhang, "MPTCP tunnel: an architecture for aggregating bandwidth of heterogeneous access networks," *Wireless Communications Mobile Computing*, vol. 2018, Article ID 2045760, 11 pages, 2018.

[6] J. Zeng, Y. Cao, F. Ke, M. Huang, G. Zhang, and W. Lu, "Performance evaluation of secure multipath retransmission mechanism in next generation heterogeneous communication systems," *IET Networks*, vol. 7, no. 2, pp. 61–67, 2018.

[7] Y. Cao, F. Song, G. Luo et al., "(PU)2M2: A potentially underperforming-aware path usage management mechanism for secure MPTCP-based multipathing services," *Concurrency & Computation Practice & Experience*, vol. 30, no. 3, 2017.

[8] J. Wu, B. Cheng, M. Wang, and J. Chen, "Quality-Aware Energy Optimization in Wireless Video Communication with Multipath TCP," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2701–2718, 2017.

[9] R. Matsufuji, D. Cavendish, K. Kumazoe, D. Nobayashi, and T. Ikenaga, "Multipath TCP path schedulers for streaming video," in *Proceedings of the 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PACRIM 2017*, pp. 1–6, Victoria, Canada, August 2017.

[10] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2345–2361, 2016.

[11] C. Xu, W. Quan, A. V. Vasilakos, H. Zhang, and G.-M. Muntean, "Information-centric cost-efficient optimization for multimedia content delivery in mobile vehicular networks," *Computer Communications*, vol. 99, pp. 93–106, 2017.

[12] Y. Cao, Q. Liu, Y. Zuo, F. Ke, H. Wang, and M. Huang, "Receiver-centric buffer blocking-aware multipath data distribution in MPTCP-based heterogeneous wireless networks," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 10, pp. 4642–4660, 2016.

[13] Y. Morikoshi, H. Abe, and K. Kato, "HayACK: Exploiting characteristically diverse paths to achieve quick ACKing in MPTCP," in *Proceedings of the 9th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2017*, pp. 383–390, Hong Kong, China, December 2017.

[14] J. Kim, B.-H. Oh, and J. Lee, "Receive Buffer based Path Management for MPTCP in heterogeneous networks," in *Proceedings of the 15th IFIP/IEEE International Symposium on Integrated Network and Service Management, IM 2017*, pp. 648–651, Lisbon, Portugal, May 2017.

[15] C. Jonghwan, D. Han, J. Kim, and C.-K. Kim, "Machine learning based path management for mobile devices over MPTCP," in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, pp. 206–209, Jeju, Republic of Korea, February 2017.

[16] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, "Alternative scheduling decisions for multipath TCP," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2412–2415, 2017.

[17] B. Hayes, Y. Chang, and G. Riley, "Adaptive bitrate video delivery using HTTP/2 over MPTCP architecture," in *Proceedings of the 13th IEEE International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, pp. 68–73, Spain, June 2017.

[18] C. Xu, P. Zhang, S. Jia, M. Wang, and G.-M. Muntean, "Video streaming in content-centric mobile networks: challenges and solutions," *IEEE Wireless Communications Magazine*, vol. 24, no. 5, pp. 157–165, 2017.

[19] Y. Cao, Q. Liu, G. Luo, Y. Yi, and M. Huang, "PR-MPTCP+: Context-aware QoE-oriented multipath TCP partial reliability extension for real-time multimedia applications," in *Proceedings of the 2016 IEEE Visual Communication and Image Processing, VCIP 2016*, pp. 1–4, Chengdu, China, November 2016.

[20] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "PASCCC: priority-based application-specific congestion control clustering protocol," *Computer Networks*, vol. 74, pp. 92–102, 2014.

[21] S. Iniya Shree, M. Karthiga, and C. Mariyammal, "Improving congestion control in WSN by multipath routing with priority based scheduling," in *Proceedings of the 2017 International Conference on Inventive Systems and Control, ICISC 2017*, Coimbatore, India, January 2017.

[22] P. Sabbagh, M. Alaei, and F. Yazdanpanah, "A priority based method for congestion control in Wireless Multimedia Sensor Networks," in *Proceedings of the 8th International Conference on Information and Knowledge Technology, IKT 2016*, pp. 177–182, Hamedan, Iran, September 2016.

[23] A. Skrastins and J. Jelinskis, "Priority-based Session Admission Control method for next generation Internet," in *Proceedings of the 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017*, pp. 153–158, Valencia, Spain, May 2017.

[24] N. V. Listova, V. V. Fedorenko, V. V. Samoylenko, and I. V. Samoylenko, "Markov's Model of the Different-Priority Traffic Routing Control in a Sensor Node," in *Proceedings of the 2017 IVth International Conference on Engineering and Telecommunication (EnT)*, pp. 155–159, Moscow, Russia, November 2017.

[25] E. Exposito, M. Gineste, L. Dairaine, and C. Chassot, "Building self-optimized communication systems based on applicative cross-layer information," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 354–361, 2009.

[26] U. C. Berkeley, LBL, USC/ISI and Xerox Parc, NS-2 documentation and software, version 2.35.

[27] Google Code Project, "Multipath-TCP: Implement multipath TCP on NS-2," http://code.google.com/p/multipath-tcp/.

[28] C.-H. Lin, C.-K. Shieh, C.-H. Ke, N. K. Chilamkurti, and S. Zeadally, "An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11e WLAN," *Telecommunication Systems*, vol. 42, no. 3-4, pp. 223–234, 2009.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 7624768, 9 pages
https://doi.org/10.1155/2018/7624768

Research Article

A Novel Method of Complexity Metric for Object-Oriented Software

Tong Yi and Chun Fang

School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China

Correspondence should be addressed to Tong Yi; 1337742168@qq.com

Received 8 September 2018; Accepted 16 October 2018; Published 1 November 2018

Guest Editor: Yuanlong Cao

With the rapid development and wide application of multimedia technology, the demand for the actual development of multimedia software in many industries is increasing. How to measure and improve the quality of multimedia software is an important problem to be solved urgently. In order to calculate the complicated situation and fuzziness of software quality, this paper introduced a software quality evaluation model based on the fuzzy matter element by using a method known as the fuzzy matter element analysis, combined with the TOPSIS method and the close degree. Compared with the existing typical software measurement methods, the results are basically consistent with the typical software measurement results. Then, Pearson simple correlation coefficient was used to analyse the correlation between the existing four measurement methods and the metric of practical experience, whose results show that the results of software quality measures based on fuzzy matter element are more in accordance with practical experience. Meanwhile, the results of this method are much more precise than the results of the other measurement methods.

1. Introduction

At present, with the rise and application of multimedia technology, it is a great challenge to provide more reliable technical support and strong technical support for the development of multimedia software. At the same time, object-oriented technology has become the mainstream of current software development, which is suitable for developing multimedia software, for example, using the image processing software Adobe Photoshop developed by C++, using Action Script to develop animation processing software Flash, and using C++ for the Jedi survival and heroic alliance games.

We must point out that multimedia software is a typical complex system; therefore, how to scientifically measure the complexity of multimedia software plays a vital role in developing high-quality multimedia software. Software metrics has become the important and long-term focused research field of software engineering and also became an important and effective method in assessing and predicting software development activities. The purpose of software metrics research is to provide guidance for developing high-quality software [1].

Since the concept of software measurement was first proposed by Rubey R. J. and Hartwick R. D. in 1968[2], the researches, development, and applications have been carried out for more than fifty years. Through literature review, this paper found that previous researches mainly from internal attributes, external attributes, and other aspects of the research of software quality metric. Over these years, many scholars have made a broad and deep research on the software quality metric and prefer to find the key or the important software quality measurement factors from the inner elements of software itself. The factors were measured or counted directly or indirectly to construct the corresponding metric model. Early metrics on structured programs were primarily focused on Lines of Code (LOC) [3], McCabe coloring graph method [4], Function Point Analysis (FPA) [5], etc.

In 1994, Chidamber S. and Kemerer C. proposed a CK metrics set for object-oriented software quality metrics research. The Weighted Methods per Class (WMC), Number of Children (NOC), Depth of Inheritance (DIT), Coupling Between Objects (CBO), Lack of Cohesion (LCOM), and Response for a Class (RFC) are included in set, which are the fundamental of object-oriented software quality metrics.

Padhy N. et al. proposed the three metrics based on CK metrics set and combined WMC, RFC, CBO, DIT. and NOC together [6]. In addition, Misra S. and Adewumi A. et al. proposed a cognitive complexity metrics set for evaluating object-oriented software projects [7], including method complexity, message complexity, attribute complexity, weighted class complexity, and code complexity. According to software measurement experience, Gupta D. L. et al. proposed some possible exist the hypothetical situation in measurement validation and design 14 measurement elements, including WMC, CBO, and RFC. Furthermore, Gupta D. L. et al. took open source software code as the data source and used SPSS software make logistic regression analysis. The results of the study showed that these methods can predict design flaw of class in software quality metric, and software defects prediction methods based on object-oriented metrics are developed [8]. Wang J. and Wang Q. found that dependency relationship is an important reason of software complexity. The dependency relationship can reflect cohesion and coupling between software elements. Meanwhile, cohesion and coupling are recognized as a measure of software quality of the important indicators. Besides, the dependency relationship of software is proved to be an important factor of software defects prediction through the experimental study. It can predict software integration errors and provide help for software quality metric in early stage [9]. The above methods of object-oriented software metrics are all belong to research of software quality metric based on software internal attributes.

However, developers and researchers paid attention to broad software quality characteristics in the process of software quality metric research based on external attributes of software quality. These characteristics include software quality characteristics of ISO/IEC 25010 software quality model in narrow sense and other software quality characteristics associated with software development and application. Gosain A. and Sharma G. defined the dynamic software quality characteristics, including robust, unambiguous, dynamic, discriminating, and machine independent. Then they evaluated cases with Java software and found that the dynamic software quality characteristic has significant positive correlation with maintainability by Pearson correlation analysis and principal component analysis [10]. Similarly, Hu X and Zuo J. et al. choose 6 software quality characteristics from GB/T16260 series of standards. The 6 software quality characteristics include capability, reliability, usability, efficiency, maintainability, and portability. Then the hierarchical model of evaluation is established for research and analysis external attributes of software quality [11].

Class diagram, a very important software model diagram, describes the classes and their relationships among the systems. They can be scientifically constructed whether or not it has a significant impact on the complexity of software. At present, the class complexity measure method is still rare. Marchesi M. [12] uses 7 indicators to measure the complexity of the class diagram from different angles. However, the method only considers the relationship between classes and inheritance, without considering other relationships, such as the association relationship and aggregation relationship. On the basis of Marchesi M. research, Genero M. [13] uses 14 indicators to further distinguish the relationship between classes and classes, that is, the combination of relative complexity measure and absolute complexity measure. The theories of Dr. Zhang Y. [14], In P. [15], Gosain A. [10], Gupta D. L. [8], and Padhy N. [6] are similar with Genero M's, which use a set of indicators to evaluate the complexity of class diagrams. The advantage of it could analyse the complexity of a class diagram from different perspectives, but its disadvantage is that it is difficult to compare two or two class diagrams. Dr. Zhou Y. transforms UML class diagrams into weighted dependencies. And then he uses the information entropy to define the complexity of UML class diagram [16], which has achieved good measure results. Dr. Yi T. has made improvements on the basis of Dr. Zhou Y. making a comprehensive consideration of interclass relationships, class attributes, and class complexity of the method. He proposed a UML class diagram complexity measurement method based on dependency analysis [17, 18].

In this paper, the research work mentioned above is a part of existing domestic and international research work, but there is no doubt that the results of researches in the UML class diagram model are not enough. One of the important reasons is that UML standard issued by the object management group (OMG) only gives the description of the semantic conceptual level in various modelling elements, which leads to the fact that the researchers often use different weighting indicators for the class diagram model. It means that researchers do not have a uniform standard, resulting in different metrics for the same class diagram. Meanwhile, because of the comprehensiveness, fuzziness, and complexity of the software quality measurement system, the software quality measurement is a process of multiple indicator decision making; the fuzzy matter element theory is introduced in this paper. In order to overcome the limitation of weight precision of the class relationship between two classes in the literature [16–18], this paper proceeds from fuzzy matter element theory, introducing the concept of close degree, and used entropy method to calculate the weight of every indicator; software quality measurement model was established in fuzzy matter element that based on entropy weight and TOPSIS method applied to UML class diagram metric. Firstly, element indicators of UML class diagram constitute the compound fuzzy matrix of matter elements and then fuzzy matrix of matter elements of the optimal subordinate degree obtained with the dimensionless, calculating the weight of each element indicators by entropy method, finally, through TOPSIS method and the concept of Euclid approach degree got comprehensive attribute values of each UML class diagrams. This paper hopes to only use a comprehensive complexity value to evaluate the complexity of UML class diagram and enough really predicts the complexity of software quality.

2. A Novel Method of Complexity Metric for Software Quality

2.1. Building Evaluate Compound Fuzzy Matter Element of Software Quality. Matter element analysis [19] is a new discipline that studies laws and methods for solving incompatible problems. It is an intersecting edge discipline of thinking science, systems science, and mathematics. Matter element analysis itself is not a branch of mathematics. It is a new discipline that develops on the basis of classical mathematics and fuzzy mathematics and is different from them. The new subject, Matter Element Analysis, which was created in 1994 by Chinese scholar Cai Wen, was specifically designed to solve incompatible problems. The fuzzy matter element combines fuzzy set theory and matter element analysis theory, which can not only solve the ambiguity of measurement indicators, but also solve the incompatibility of measurement results. Because of its simple calculation method, reliable evaluation results, and strong practicality, this theory is widely used in logistics science and technology [20], electromechanical [21], architecture [22], and other fields.

The matter element $R = (T, C, X)$ for evaluating the software quality was constructed in this paper, where T denotes the software class diagram to be evaluated, C denotes the evaluation indicator, and X denotes the corresponding magnitude of the evaluation indicator. If X has ambiguity, R is called a fuzzy matter element. If T has n evaluation indicators $C_1, C_2, \ldots, C_n$ whose corresponding magnitudes are $X_1, X_2, \ldots, X_\mathrm{n}$, R is said to be n-dimensional fuzzy matter elements[23]. The n-dimensional matter elements of m the software diagrams to be evaluated are combined to form the n-dimensional compound fuzzy matter elements of m the software diagram to be evaluated. R_{mn} is defined as follows:

$$R_{mn} = \left(\begin{array}{ccccc} & T_1 & T_2 & \cdots & T_m \\ C_1 & X_{11} & X_{21} & \cdots & X_{m1} \\ C_2 & X_{12} & X_{22} & \cdots & X_{m2} \\ \vdots & \vdots & \vdots & & \vdots \\ C_n & X_{1n} & X_{2n} & \cdots & X_{mn} \end{array} \right) \tag{1}$$

In formula (1), T_i represents the i (i=1,2,... m) software class diagram, C_j is the j (j=1,2,...,n) evaluation indicator of the software class diagram, and X_ij represents the corresponding magnitude of the j evaluation indicator of the i software class diagram.

2.2. Dimensionless of Evaluation Indicators. In the evaluation of software class diagrams, there are many evaluation indicators involved. If there are no uniform metrics among the indicators, the evaluation process will be difficult to carry out. In order to compare the different dimension indicators together for comparison, the magnitude of these evaluation indicators must be dimensionless [23]. The dimensionless process is to remove the dimension's influence on the physical value through mathematical methods. There are generally two types of indicators for quantification processing results, some of which are larger and better indicators, that is, positive indicators; others are smaller, better indicators, that is, negative indicators. According to the actual situation, this paper selects the smaller and better indicators in the software quality evaluation.

$$\max_j = \max\left(X_{1j}, X_{2j}, \ldots, X_{mj}\right) \tag{2}$$

$$\min_j = \min\left(X_{1j}, X_{2j}, \ldots, X_{mj}\right) \tag{3}$$

$$u_{ij} = \frac{\max_j - X_{ij}}{\max_j - \min_j} \tag{4}$$

In formula (4), u_{ij} is the dimensionless result of the j-th evaluation indicator of the i-th software class diagram. $\max_j$ is the maximum value of the j-th evaluation indicator of the software class diagram, and $\min_j$ is the minimum value of the j-th evaluation indicator of the software class diagram.

After the dimensionless treatment of formula (1) through formula (4), formula (5) is obtained, that is, the fuzzy matter element weight matrix of optimal membership degree R'_{mn}.

$$R'_{mn} = \left(\begin{array}{ccccc} & T_1 & T_2 & \cdots & T_m \\ C_1 & u_{11} & u_{21} & \cdots & u_{m1} \\ C_2 & u_{12} & u_{22} & \cdots & u_{m2} \\ \vdots & \vdots & \vdots & & \vdots \\ C_n & u_{1n} & u_{2n} & \cdots & u_{mn} \end{array} \right) \tag{5}$$

2.3. Evaluation Indicator Weight Determining Based on Entropy Method. In the process of software quality evaluation, the weight of an indicator reflects the relative importance of the indicator in the overall evaluation process. Therefore, the determination of weight is very important. Common weight determination methods include entropy method, expert scoring method, and analytic hierarchy process. This paper uses entropy method to calculate weights to achieve the subjective and objective unity of weights. The entropy method is based on the difference in the degree of information contained in each indicator, that is, the utility value of the information to determine the weight of the indicator. It is an objective weighting method.

The formula for calculating the information entropy and weight function in the comprehensive evaluation is as follows: For the software quality evaluation model in question, if there are initial data matrix R_{mn} of the n evaluation indicators of the m software class diagram to be evaluated, each indicator is significantly different in the dimension, order of magnitude, and merits of indicators. Therefore, the initial data must be standardized:

$$y_{ij} = \frac{X_{ij}}{\sum_{i=1}^{m} X_{ij}}, \quad i = 1, 2, \ldots, m; \; j = 1, 2, \ldots, n \tag{6}$$

Get information entropy of the j-th evaluation indicator according to formula (7):

$$e_j = -k \sum_{i=1}^{m} y_{ij} \ln y_{ij}, \quad j = 1, 2, \ldots, n \tag{7}$$

The constant k in formula (7) is related to the number of samples, m, and $k = 1/\ln m$ is often taken. Because of information entropy e_{j} can be used to measure the information utility value of the j-th evaluation indicator. When the sample is completely disordered, e_{j} =1; meanwhile, the information value of e_{j} is zero for the utility value of the comprehensive evaluation. Therefore, the information utility value of an evaluation indicator is determined by the difference between 1 and the information entropy e_{j} of the evaluation indicator; that is,

$$h_j = 1 - e_j, \quad j = 1, 2, \ldots, n \tag{8}$$

The entropy method is used to estimate the weight of the evaluation indicator. Its essence is to use the information utility value of the evaluation indicator to measure. When the difference h_j is higher, the importance of the evaluation is bigger, so the weight of the j-th evaluation indicator is

$$w_j = \frac{h_j}{\sum_{j=1}^{n} h_j}, \quad j = 1, 2, \ldots, n \tag{9}$$

Fuzzy matter element weight matrix of optimal membership degree is

$$R_w = \begin{pmatrix} & C_1 & C_2 & \cdots & C_n \\ w_j & w_1 & w_2 & \cdots & w_n \end{pmatrix} \tag{10}$$

2.4. Fuzzy Compound Matter Element for Evaluating the Quality Characteristics. R_s is a weighted fuzzy compound matter element for evaluating the quality characteristics, and then there are

$$R_s = \begin{pmatrix} & T_1 & T_2 & \cdots & T_m \\ C_1 & C_{11} = w_1 u_{11} & C_{21} = w_1 u_{21} & & C_{m1} = w_1 u_{m1} \\ C_2 & C_{12} = w_2 u_{12} & C_{22} = w_2 u_{22} & & C_{m1} = w_2 u_{m2} \\ \vdots & & & & \\ C_n & C_{1n} = w_n u_{1n} & C_{2n} = w_n u_{2n} & & C_{mn} = w_n u_{mn} \end{pmatrix} \tag{11}$$

In formula (11), $C_{ij} (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ the calculation value of the j-th evaluation indicator of the i-th software class diagram is represented.

2.5. Calculating Comprehensive Evaluation of Software Quality. TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [24] is a multiobjective decision-making method. The basic idea is to define the ideal solution and negative ideal solution of the decision problem. It is assumed that the ideal solution is the optimal program and the negative ideal solution is the worst program. If there is an evaluation plan in the feasible evaluation plan, the evaluation plan is the closest to the ideal, while far away from the negative ideal solution, we call this program the optimal program.

Further determine the ideal solution vector C^+ and negative ideal solution vector C^- of matrix R_s:

$$\begin{aligned} C^+ &= \left(C_1^+, C_2^+, \ldots, C_n^+\right), \\ C_j^+ &= \max\left\{C_{1j}, C_{2j}, \ldots, C_{mj}\right\}, \quad j = 1, 2, \ldots, n \end{aligned} \tag{12}$$

$$\begin{aligned} C^- &= \left(C_1^-, C_2^-, \ldots, C_n^-\right), \\ C_j^- &= \min\left\{C_{1j}, C_{2j}, \ldots, C_{mj}\right\}, \quad j = 1, 2, \ldots, n \end{aligned} \tag{13}$$

There are several ways to calculate the distance between ideal solutions and negative ideal solutions, such as Euclidean distance, Manhattan distance, Chebyshev distance, and so on. Among them, Euclidean distance is an easy-to-understand distance calculation method, which is derived from the distance formula between two points in Euclidean geometry. In this paper, the Euclidean distance is used, and its calculation formula is as follows [23]:

$$S_i^+ = \left\| C^+ - C_i \right\| = \sqrt{\sum_{j=1}^{n} \left(C^+ - C_{ij}\right)^2}, \quad i = 1, 2, \ldots, m \tag{14}$$

$$S_i^- = \left\| C_i - C^- \right\| = \sqrt{\sum_{j=1}^{n} \left(C_{ij} - C^-\right)^2}, \quad i = 1, 2, \ldots, m \tag{15}$$

In formula (14) and formula (15), $C_i = (C_{i1}, C_{i2}, \cdots, C_{in})^T$; C_i is the i-th column vector of matrix R_s.

In this paper, the comprehensive evaluation of software quality adopts entropy method for consideration. The source has S_i^+ and S_i^-. The binary entropy function can be used to calculate the weights of the Euclidean distance between each class diagram to be evaluated and the ideal solution.

$$\begin{aligned} H^+ = H\left(S_i^+\right) = -S_i^+ \log_2 S_i^+ - \left(1 - S_i^+\right) \log_2\left(1 - S_i^+\right), \\ i = 1, 2, \ldots, m \end{aligned} \tag{16}$$

Similarly, the binary entropy function is used to calculate the weights of the Euclidean distances of each class diagram to be evaluated and the negative ideal solution; namely,

$$\begin{aligned} H^- = H\left(S_i^-\right) = -S_i^- \log_2 S_i^- - \left(1 - S_i^-\right) \log_2\left(1 - S_i^-\right), \\ i = 1, 2, \ldots, m \end{aligned} \tag{17}$$

According to the concept of close degree [23], combined with the uncertainty of the ideal solution and the negative ideal solution, the fuzzy matter element software quality metric measures the software quality by the following uncertainty-weighted fusion method. The calculation formula is as follows:

$$Z_i = \frac{H^+ S_i^+}{H^+ S_i^+ + H^- S_i^-}, \quad i = 1, 2, \ldots, m \tag{18}$$

In formula (18), the value is between 0 and 1. The closer the value is to 0, the evaluation object complexity is smaller and the closer to the optimal ideal level.

3. Case Analysis

3.1. Data Sources. In order to validate the measurement method proposed in this paper, we will do an experiment to estimate the metric values. With the permission of Genero M., we selected twenty-six UML class diagrams [13] related to the bank information systems as the object of the experiment. For better representation, NDep represents dependency, NAssoc represents normal association, NAgg represents aggregation, NGen represents generalization, NM represents class method, NA represents class attribute, and NC represents the number of classes. For specific indicators and data for details, see Table 1.

3.2. Model Establishment. According to the above theory and evaluation indicator system, the steps for establishing a fuzzy matter element evaluation model are as follows.

Step 1. Construct the composite fuzzy matrix of matter elements according to Table 1.

Step 2. Calculate the degree of optimal membership. According to the compound fuzzy matter element matrix determined in the first step, the degree of optimal membership is calculated using formula (4), and the fuzzy matter element matrix of optimal membership degree is obtained.

$$R'_{26\times 7} = \left(\begin{array}{cccccccc} & T_1 & T_2 & T_3 & T_4 & \cdots & T_{25} & T_{26} \\ NDep & 1 & 1 & 1 & 1 & \cdots & 0 & 1 \\ NAssoc & 0.928571 & 0.928571 & 0.928571 & 0.785714 & \cdots & 0 & 0.642857 \\ NAgg & 1 & 0.888889 & 0.777778 & 1 & \cdots & 0.555556 & 0 \\ NGen & 1 & 1 & 1 & 1 & \cdots & 0.333333 & 0.708333 \\ NM & 1 & 0.955556 & 0.922222 & 0.955556 & \cdots & 0.155556 & 0.233333 \\ NA & 1 & 0.961538 & 0.903846 & 0.942308 & \cdots & 0.269231 & 0.423077 \\ NC & 1 & 0.967742 & 0.935484 & 0.967742 & \cdots & 0.354839 & 0.612903 \end{array}\right) \quad (19)$$

Step 3. Based on the fuzzy matter element matrix of optimal membership degree $R'_{27\times 7}$, according to formula (6), formula (7), formula (8), formula (9), formula (10), and formula (11) by entropy method to obtain each indicator weights that composes the fuzzy matter element weight matrix of optimal membership degree R_w,

$$R_w = \left(\begin{array}{cccccccc} & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \\ w_j & 0.17124 & 0.143127 & 0.150199 & 0.124097 & 0.189334 & 0.191128 & 0.184991 \end{array}\right) \quad (20)$$

Step 4. Get R_s by formula (11).

$$R_s = \left(\begin{array}{cccccccc} & T_1 & T_2 & T_3 & T_4 & \cdots & T_{25} & T_{26} \\ NDep & 0.017124 & 0.017124 & 0.017124 & 0.017124 & \cdots & 0 & 0.017124 \\ NAssoc & 0.132904 & 0.132904 & 0.132904 & 0.112457 & \cdots & 0 & 0.09201 \\ NAgg & 0.150199 & 0.13351 & 0.116821 & 0.150199 & \cdots & 0.083444 & 0 \\ NGen & 0.124097 & 0.124097 & 0.124097 & 0.124097 & \cdots & 0.041366 & 0.087902 \\ NM & 0.189334 & 0.18092 & 0.174608 & 0.18092 & \cdots & 0.029452 & 0.044178 \\ NA & 0.191128 & 0.183777 & 0.17275 & 0.180101 & \cdots & 0.051458 & 0.080862 \\ NC & 0.184991 & 0.179024 & 0.173056 & 0.179024 & \cdots & 0.065642 & 0.113382 \end{array}\right) \quad (21)$$

Table 1: Twenty-six UML class diagrams evaluation indicators.

Class diagrams	NDep	NAssoc	NAgg	NGen	NM	NA	NC
1	0	1	0	0	8	4	2
2	0	1	1	0	12	6	3
3	0	1	2	0	15	9	4
4	0	3	0	0	12	7	3
5	0	1	3	0	21	14	5
6	0	2	0	0	12	6	3
7	1	3	0	0	13	8	4
8	0	2	2	2	14	10	6
9	1	1	0	0	12	9	3
10	0	2	3	2	22	14	7
11	0	2	3	4	30	18	9
12	0	3	3	2	39	19	7
13	1	3	2	2	35	22	8
14	0	0	0	4	30	11	5
15	0	0	0	10	30	12	8
16	0	0	0	18	38	17	11
17	2	11	6	10	76	42	20
18	1	11	6	16	88	41	23
19	1	7	6	20	94	45	21
20	3	13	7	24	98	56	33
21	0	1	5	2	47	28	9
22	0	3	5	20	65	31	18
23	0	11	6	21	79	44	26
24	0	1	5	19	69	32	17
25	4	14	4	16	84	42	22
26	0	5	9	7	77	34	14

Note: Genero metric values from Genero M.'s experiment in Table 1.

Step 5. Software quality measurement values of fuzzy matter element in this paper are calculated by formula (12), formula (13), formula (14), formula (15), formula (16), formula (17), and formula (18).

$$\begin{aligned} Z_i = {} & (0.02089, 0.101997, 0.339518, 0.205419, \\ & 0.856791, 0.106658, 0.251853, 0.510079, \\ & 0.105591, 1.075057, 1.733703, 2.00285, \\ & 1.906329, 0.606743, 1.143489, 2.588069, \\ & 8.394743, 9.021374, 8.872664, 9.91465, \\ & 3.522168, 6.674977, 9.271023, 6.446144, \\ & 8.760155, 7.146074) \end{aligned} \tag{22}$$

3.3. Data Analysis

3.3.1. Comparing the Experiment Results of Four Metrics. To verify the effectiveness and practicability of the proposed measurement method, this paper plans to compare with the method proposed by Dr. Zhou Y. [16] and the method proposed by Dr. Yi T. [17, 18] in the three aspects, understandability, analysability, and maintainability. For convenient discussion, the method of Dr. Zhou Y. and the method of Dr. Yi T. are called Zhou metric [16], Yi15 metric [17], and Yi18 metric [18]; the measurement method proposed in this paper is called Z_i metric, as shown in Table 2.

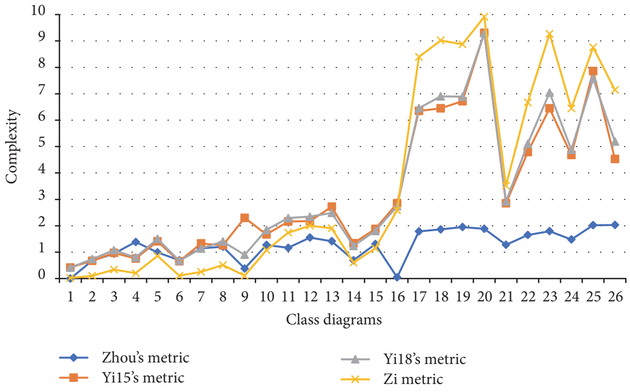


Figure 1: Comparing the experiment results.

Comparing the experimental results of the above four software quality measurement models, as shown in Figure 1,

Table 2: Comparing the experiment results of measurement methods.

Class diagrams	Zhou metric	Yi15 metric	Yi18 metric	Z_i metric	Understandability	Analysability	Maintainability
1	0	0.42	0.41176	0.02089	1	1	1
2	0.673012	0.67	0.73657	0.101997	2	2	2
3	0.940493	0.97	1.07097	0.339518	2	2	2
4	1.386294	0.76	0.7987	0.205419	2	2	2
5	0.989909	1.41	1.50423	0.856791	2	2	2
6	0.693147	0.65	0.66949	0.106658	2	2	2
7	1.14688	1.33	1.13783	0.251853	2	3	3
8	1.206376	1.26	1.40369	0.510079	3	3	3
9	0.381909	2.3	0.9008	0.105591	2	2	2
10	1.271002	1.67	1.8433	1.075057	3	3	3
11	1.16503	2.16	2.29233	1.733703	3	3	3
12	1.553338	2.18	2.34555	2.00285	3	3	3
13	1.414547	2.72	2.48526	1.906329	3	3	3
14	0.693147	1.34	1.237	0.606743	2	2	2
15	1.303487	1.88	1.80312	1.143489	2	3	3
16	0.04308	2.85	2.7398	2.588069	4	4	4
17	1.787461	6.35	6.45495	8.394743	6	6	6
18	1.8612	6.45	6.90285	9.021374	6	6	6
19	1.949444	6.72	6.88925	8.872664	6	5	6
20	1.883662	9.31	9.29632	9.91465	6	6	7
21	1.277816	2.85	2.91541	3.522168	3	3	3
22	1.649751	4.79	5.10804	6.674977	5	5	5
23	1.794866	6.45	7.04766	9.271023	6	6	6
24	1.480208	4.68	4.88021	6.446144	5	5	5
25	2.020782	7.86	7.5702	8.760155	6	5	6
26	2.030221	4.53	5.19316	7.146074	4	5	5

and comparing them with the understandability, analysability, and maintainability of the class diagrams obtained by the practical experience, it is found that the four metric results are similar. But also some interesting results are found.

(1) For the class diagram 4, the Zhou metric has higher values for the computational class diagram 4 complexity, the Yi15 metric and the Yi18 metric have lower values for the complexity of the class diagram 4, and the complexity of the class diagram 4 that the practical experience has obtained has lower values. The complexity of class diagram 4 calculated using the fuzzy matter element model in this paper is low, which is consistent with the actual experience.

(2) For the class diagram 9, the Zhou metric and the Yi18 metric have lower values for the computational class diagram 9 complexity, the Yi15 metric has higher values for the complexity of the class diagram 9, and the complexity of the class diagram 9 that the practical experience has obtained has lower values. The complexity of class diagram 9 calculated using the fuzzy matter element model in this paper is low, which is consistent with the actual experience.

(3) For the class diagram 16, the Zhou metric has lower values for the computational class diagram 16 complexity, the Yi15 metric and the Yi18 metric have higher values for the complexity of the class diagram 16, and the complexity of the class diagram 16 that the practical experience has obtained has higher values. The complexity of class diagram 16 calculated using the fuzzy matter element model in this paper is high, which is consistent with the actual experience.

(4) For the class diagram 19, the Yi18 metric for the complexity of the class diagram 19 has lower values than the class diagram 18, the Zhou metric and the Yi15 metric have higher values for the computational class diagram 19 complexity, and the complexity of the class diagram 19 that the practical experience obtained has lower values than the class diagram 18. The complexity of class diagram 19 calculated using the fuzzy matter element model in this paper is consistent with actual experience.

(5) For the class diagram 25 and the class diagram 26, the Zhou metric shows that the 26th class diagram complexity is higher than the 25th class diagram complexity and the Yi15 metric and Yi18 metric methods show the 26th class diagram complexity lower than it. The complexity of the class diagrams 25 and 26 obtained by the practical experience is opposite with the Zhou metric, which is in consistent with

Table 3: Correlation coefficient and correlation intensity.

Correlation coefficient absolute value	Correlation intensity
$\lvert r\rvert = 0$	Zero correlation
$0 < \lvert r\rvert \leq 0.3$	Weak correlation
$0.3 < \lvert r\rvert \leq 0.5$	Low correlation
$0.5 < \lvert r\rvert \leq 0.8$	Significant correlation
$0.8 < \lvert r\rvert \leq 1$	High correlation
$\lvert r\rvert = 1$	Completely correlation

the complexity of the class diagram calculated by using the fuzzy matter element model in this paper.

3.3.2. Pearson Simple Correlation Coefficient Test. In order to further discuss the existing correlation between the results of complexity metric and the value of understandability, the value of analysability, and the value of maintainability, we propose the Pearson simple correlation coefficient to test whether or not the complexity measure method is consistent with the practical experience. Pearson simple correlation coefficient is calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum\left(X_i - \overline{X}\right)^2}\sqrt{\sum\left(Y_i - \overline{Y}\right)^2}} \quad (23)$$

The correlation intensity between the two variables refers to Table 3.

Using the well-known statistical software SPSS for correlation analysis and the results of the correlation analysis are shown in Table 4.

Through the comparison and data analysis in Table 4, we can find that this paper's UML class diagram metric is consistent with practical experience in the understandability, analysability, and maintainability. The value of the UML class diagrams complexity measure Z_i is calculated by fuzzy matter element model, which is compared with practical experience. The Pearson simple correlation coefficient between Z_i and the value of the understandability of practical experience is 0.959. The Pearson simple correlation coefficient between Z_i and the value of the analysability of practical experience is 0.956. The Pearson simple correlation coefficient between Z_i and the value of the maintainability of practical experience is 0.962. Zhou metric is significantly correlated with the class diagram of practical experience. But the fuzzy matter element model metrics, Yi15 metric, and Yi18 metric are highly correlated with the class diagram of practical experience. Therefore, the complexity measure method of this paper is consistent with the practical experience from the view of average.

3.3.3. The Analysis of the Visualization of Measurement Results. In order to compare the abovementioned four metrics methods more intuitive, the classification results of this paper are shown in Figure 2. For better representation, U represents the understandability, A represents the analysability, and M represents the maintainability.

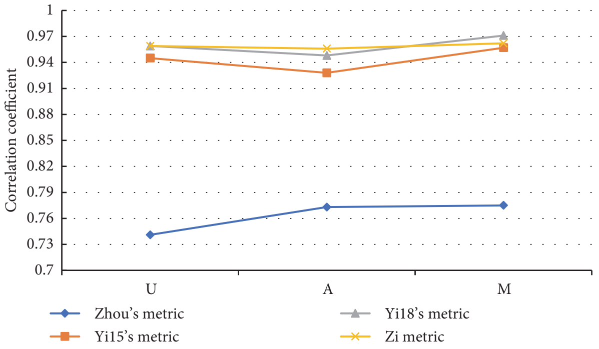


Figure 2: Pearson simple correlation analysis.

From Figure 2, we can find that this paper's results are closer to 1. It suggests that the class diagram complexity calculated by the fuzzy matter element model is consistent with the value of practical experience by comparing with other metrics. This method can quickly calculate the comprehensive attribute value of the software class diagrams. Meanwhile, the results of this study can be more accurately to reflect the software complexity. So the measurement model proposed in this paper is relatively better.

4. Conclusions

This paper uses the basic theory and method of matter element analysis, combined with fuzzy set theory and TOPSIS method to establish a fuzzy matter element model based on entropy weight and TOPSIS method. It is applied to the evaluation of software class diagram, and at the same time the difference between the entropy values as a weight, making full use of the information in the original data, to a certain extent reduces the subjectivity of weight determination; the evaluation results are in good agreement with the actual situation, indicating that the method is reasonable and feasible.

Data Availability

The data used to support the findings of this study were supplied by M. Genero under license. M. Genero at the Department of Computer Science at the University of Castilla-La Mancha, Ciduad Real, Spain, has allowed the author to quote the twenty-seven UML class diagrams related to bank information systems and the corresponding metric values. Reference: M. Genero. Defining and validating metrics for conceptual models [D], University of Castilla-La Mancha, 2002.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Table 4: The correlation analysis of the complexity measurement results.

Metric methods	Understandability	Analysability	Maintainability	Average
Zhou metric	0.741∗∗	0.773∗∗	0.775∗∗	0.763
Yi15 metric	0.945∗∗	0.928∗∗	0.957∗∗	0.949667
Yi18 metric	0.959∗∗	0.948∗∗	0.971∗∗	0.959333
Z_i metric	0.959∗∗	0.956∗∗	0.962∗∗	0.959

Note: ∗∗ indicates significant correlations at the 0.01 level in bilateral test.

Acknowledgments

This research has been supported by the Science and Technology Foundation of Jiangxi Provincial Department of Education (Project Name: Research on Software Complexity Measurement Based on Multiple Attribute Decision Making).

References

[1] X. Zhou, X.-K. Chen, J.-S. Sun, and F.-Q. Yang, "Software measurement based reusable component extraction in object-oriented system," *Acta Electronica Sinica*, vol. 31, no. 5, pp. 649–653, 2003.

[2] R. J. Rubey and R. D. Hartwick, "Quantitative measurement of program quality," *ACM National Computer Conference*, vol. 23, pp. 671–677, 1968.

[3] B. Hardekopf and C. Lin, "The ant and the grasshopper: Fast and accurate pointer analysis for millions of lines of code," *ACM SIGPLAN Notices*, vol. 42, no. 6, pp. 290–299, 2007.

[4] T. J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, 1976.

[5] N. Choursiya and R. Yadav, "An enhanced function point analysis (FPA) method for software size estimation," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 2797–2799, 2015.

[6] N. Padhy, S. Satapathy, and R. P. Singh, "Utility of an object oriented reusability metrics and estimation complexity," *Indian Journal of Science and Technology*, vol. 10, no. 3, pp. 1–9, 2017.

[7] S. Misra, A. Adewumi, L. Fernandez-Sanz, and R. Damasevicius, "A suite of object oriented cognitive complexity metrics," *IEEE Access*, vol. 6, pp. 8782–8796, 2018.

[8] D. L. Gupta and K. Saxena, "Software bug prediction using object-oriented metrics," *Sādhanā*, vol. 42, no. 5, pp. 655–669, 2017.

[9] J. Wang and Q. Wang, "Analyzing and predicting software integration bugs using network analysis on requirements dependency network," *Requirements Engineering*, vol. 21, no. 2, pp. 161–184, 2016.

[10] A. Gosain and G. Sharma, "A dynamic size measure for object oriented software," *International Journal of Systems Assurance Engineering and Management*, vol. 8, pp. 1209–1221, 2017.

[11] X. Hu, J. Zuo, and K. Wang, "Study on AHP-based quantification of software quality," *Computer Application and Software*, vol. 30, no. 11, pp. 138–141, 2013.

[12] M. Marchesi, "OOA metrics for the Unified Modeling Language," in *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering, CSMR 1998*, pp. 67–73, Italy, March 1998.

[13] M. Genero, M. Piattini, and M. Chaudron, "Quality of UML models," *Information and Software Technology*, vol. 51, no. 12, pp. 1629-1630, 2009.

[14] Y. Zhang, J. Tao, and L. Qian, "A metrics suite for class complexity based-on UML," *Computer Science*, vol. 29, no. 10, pp. 128–132, 2002.

[15] P. In, S. Kim, and M. Barry, "UML-based object-oriented metrics for architecture complexity analysis," in *Proceedings of the Ground System Architectures Workshop the Aerospace Corporation*, March 2003.

[16] H. Lu, Y. Zhou, B. Xu, H. Leung, and L. Chen, "The ability of object-oriented metrics to predict change-proneness: A meta-analysis," *Empirical Software Engineering*, vol. 17, no. 3, pp. 200–242, 2012.

[17] T. Yi, "On the application of information entropy-based multi-attribute decision in UML class diagram metrics," *International JoUrnal of u- and e-Service, Science and Technology*, vol. 8, no. 6, pp. 105–116, 2015.

[18] T. Yi and C. Fang, "A complexity metric for object-oriented software," *International Journal of Computers and Applications*, pp. 1–6, 2018.

[19] F. Geng and X. Ruan, "Campus network information security risk assessment based on FAHP and matter element model," in *Intelligent Computing Methodologies*, vol. 10363, pp. 298–306, 2017.

[20] Y. Hu, "Comprehensive evaluation of multi index panel data based on fuzzy matter element analysis," *Statistics & Decision*, no. 14, pp. 32–35, 2016.

[21] H. Jiang, Q. Zhang, and J. Peng, "An improved cloud matter element model based wind farm power quality evaluation," *Power System Technology*, vol. 38, no. 1, pp. 205–210, 2014.

[22] W. J. You, Z. S. Xu, and D. L. Liu, "On the fire risk assessment for the ancient buildings based on the matter element analysis," *Journal of Safety & Environment*, vol. 17, no. 3, pp. 873–878, 2017.

[23] Q. Pang, H. Wang, and Z. Xu, "Probabilistic linguistic term sets in multi-attribute group decision making," *Information Sciences*, vol. 369, pp. 128–143, 2016.

[24] R. N. Sun, B. Zhang, and T. T. Liu, "Service ranking method based on improved entropy TOPSIS," *Journal of Chinese Computer Systems*, vol. 38, no. 6, pp. 1221–1226, 2017.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 7543875, 9 pages
https://doi.org/10.1155/2018/7543875

Research Article

Video Data Integrity Verification Method Based on Full Homomorphic Encryption in Cloud System

Ruoshui Liu, Jianghui Liu, Jingjie Zhang, and Moli Zhang

Information Engineering College, Henan University of Science and Technology, Luoyang 471003, China

Correspondence should be addressed to Jianghui Liu; jihua@haust.edu.cn

Received 1 August 2018; Accepted 16 September 2018; Published 22 October 2018

Guest Editor: Yuanlong Cao

Cloud computing is a new way of data storage, where users tend to upload video data to cloud servers without redundantly local copies. However, it keeps the data out of users' hands which would conventionally control and manage the data. Therefore, it becomes the key issue on how to ensure the integrity and reliability of the video data stored in the cloud for the provision of video streaming services to end users. This paper details the verification methods for the integrity of video data encrypted using the fully homomorphic crytosystems in the context of cloud computing. Specifically, we apply dynamic operation to video data stored in the cloud with the method of block tags, so that the integrity of the data can be successfully verified. The whole process is based on the analysis of present Remote Data Integrity Checking (RDIC) methods.

1. Introduction

In the current era of rapid development of the Internet and big data technologies [1–5], the emergence of cloud computing becomes inevitable. Cloud computing provides large enterprises with an on-demand solution that enables companies to lease cloud service in the form of infrastructure or software to conduct tasks, e.g., data management, business expansion and service provision [6]. Cloud computing also provides individuals with a variety of cloud services. Typically, cloud provisions of video services have greatly improved the user experience [7]. Video data stored in the cloud share some common characteristics, e.g., large volume, high redundancy, and fast real-time requirement. The compressed video data requires functions such as data location indexing and controllable coding rate. However, cloud computing has been controversial regarding its security since its inception, and users cannot be guaranteed the security of video data in the cloud. In other words, tenants cannot fully trust cloud service providers [8]. Firstly, in multitenant resource sharing environment, tenants normally express concern about their video data which could be leaked, falsified, and unauthorizedly spread by cloud service providers or other tenants. Secondly, there is a risk of illegal access because virtual machines cannot be effectively and securely isolated. Thirdly, data and processes in cloud computing often exist in a distributed manner; data belonging to multiple parties needs to be shared with assurance of leakage free and verified integrity [9]. These characteristics of video data determine that video data encryption should generally meet the following requirements.

(i) Security. Security is the primary requirement for data encryption. It is generally accepted that when the cost of deciphering the password is greater than that of directly purchasing the video, the cryptosystem is secure. Since the video data can also be regarded as ordinary binary data, conventional passwords can be used in video encryption. In addition, the large amount of video data gives rise to the increased level of difficulty when code-breakers inevitably perform a large number of decoding operations on the encrypted data. Therefore, some typical and fast encryption algorithms can be applied to ensuring security.

(ii) Compression Ratio. Generally speaking, the amount of data before and after encryption and decryption remains unchanged, so the compression ratio keeps unaltered. This feature is called compression rate invariability. Data encryption using the algorithm with the compression rate invariability does not change the physical space in

storage. The transmission rate of encrypted data remains the same.

(iii) Real-Time. As it is required for real-time transmission and access of video data, the use of encryption and decryption algorithms cannot insert too much delay. Therefore, the encryption and decryption algorithms need to be fast.

(iv) Data Format Invariability. The invariability of the data format defined here means that the format of the video data before encryption and after decryption remains unchanged. This feature brings a number of advantages. The important one is to make the time positioning of video data possible. This enables the support of the addition, deletion, cut, and paste operations to the video data.

(v) Data Operability. In some cases, it is required to directly operate on the encrypted data without having to perform the cumbersome process of decrypting and then encrypting. These operations include rate control, image block clipping, addition, and deletion. The algorithms with which some operations become still operable after data is encrypted is said to have data operability.

In the past ten years, there have been many encryption algorithms applied to MPEG video streams [10]. All algorithms can meet different levels of security requirements. Most of the algorithms ensure the real-time nature of video streaming and display processing. Some of them guarantee that the compression ratio is unchanged. In addition, compatibility, operability, abnormality [11], and routing [12, 13] have been also addressed in other algorithms. Based on the difference between the encryption algorithm and the compression coding process, we divide the existing algorithms into the following categories [14]. The first is the direct encryption algorithm in which video data is considered as ordinary data to be directly encrypted. Therefore, the algorithms in this category do not have compatibility. The second is called the selective encryption algorithm in which video data is partially and selectively encrypted, and those algorithms are compatible. The third is called the encryption algorithm with compression function. The algorithms in this categories combine encryption process, the compression and encoding process together, so that they embraces the features of being compressive, compatible, and operable.

This paper proposes a video data integrity checking method based on homomorphic encryption. The user can verify the integrity of the data and support public verification and data dynamics. Using homomorphic tags can greatly reduce the bandwidth requirement for video data integrity checking solution. The proposed method and implemented services are deployed on the cloud system, which reduces the cloud user's communication and computational overhead. It is proved to be feasible through security analysis and performance analysis with experimental results.

2. Related Work

Resource monitoring is an important part for resource management of cloud platform. It provides the basis for resource allocation, task scheduling and load balancing [15]. Since the cloud computing environment has the characteristics of transparent virtualization and resource flexibility, it is infeasible to apply conventional methods to protect the data security in the cloud platform. Additionally, the collection, transmission, storage, and analysis of a large number of monitored data will bring much cost. Therefore, it is of critical importance to develop new tools suitable for monitoring data in the cloud.

The cryptographic protocol is an essential part of most security modules [16]. In a broad sense, all cryptographic protocols are a special case of secure multiparty computation. They are widely used in many fields, such as financial trading, social networking, real-time monitoring, and information management. Conventional cryptographic protocols often include multiple participants, who may be trusted parties (e.g., the user and authenticated participants) or untrusted parties (unauthenticated participants). Theoretically, all protocols with untrustworthy parties have the potential for the adoption of full homomorphic encryption. Therefore, most applications of all-homomorphic encryption can be considered as a secure multiparty computation. Fully homomorphic encryption allows various operations to be carried out to encrypted data without a private key. This enables computing of sensitive data with encryption to be outsourced, so that data security and privacy problems in the current development of cloud computing can be effectively solved [17]. The general application framework of all-homomorphic encryption is shown in Figure 1.

The homomorphic encryption algorithm is the data obfuscation algorithm in code obfuscation [18]. The data in the program not only contains numbers but also characters. It is insufficient to use homomorphic encryption to numbers only. Moreover, the execution efficiency of the program will be slowed down after the code is obfuscated. The Fourier transform can reduce the amount of calculation and the length of the ciphertext. It can also improve the operational efficiency of the program while ensuring security. The data obfuscation in code obfuscation includes polynomial obfuscation, data conversion obfuscation, etc. Their disadvantage is that the data is easily exposed during encryption and decryption. The relationship between reverse engineering and obfuscation algorithms is shown in Figure 2.

The homomorphic encryption algorithm operates internally, and it can be processed without decryption. As the increase in demand for information security becomes apparent, especially in the applications of cloud computing and e-commerce, research on homomorphic encryption algorithms is constantly deepening [18].

It has been found that not only homomorphic encryption can be applied to cloud computing, a number of computing functions that satisfy multiple additions and few multiplications are also useful for privacy-preserving cloud services. For example, averaging does not require multiplication. Standard deviation requires only one multiplication, and some predictive analysis such as logistic regression requires very few multiplications. In the homomorphic encryption schemes [19], schemes like RSA satisfy the multiplicative homomorphism [20] and others like Pailer satisfy the

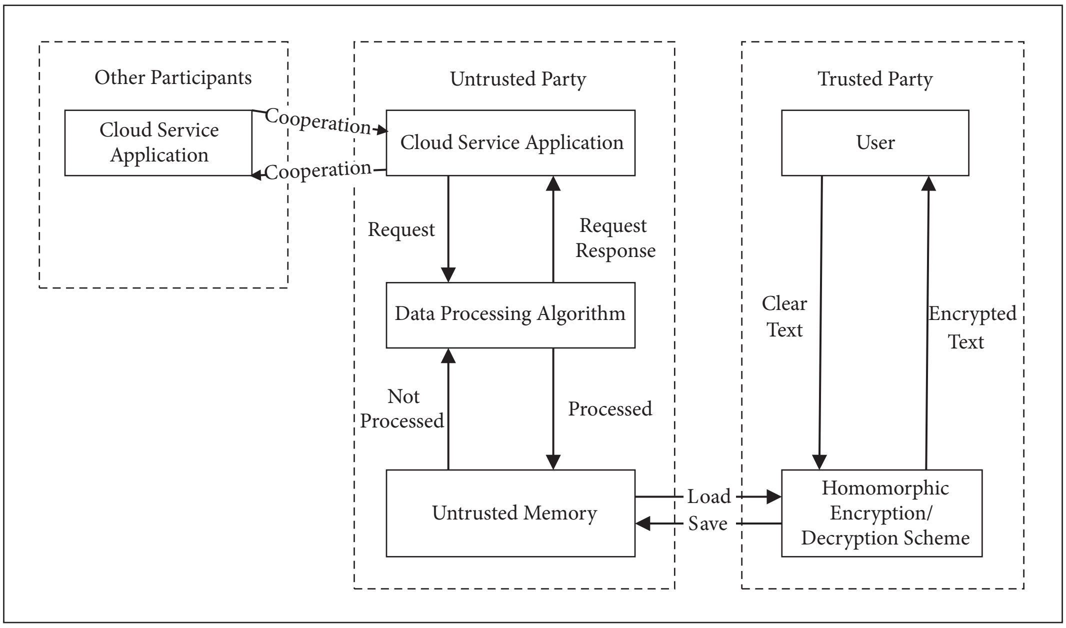


Figure 1: General application framework of fully homomorphic scheme.

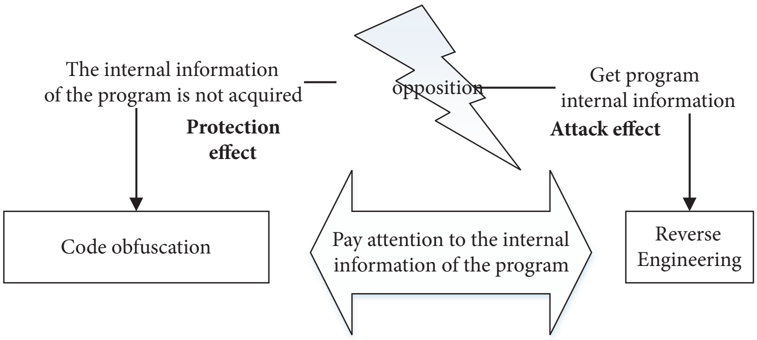

The internal information of the program is not acquired
Protection effect
opposition
Get program internal information
Attack effect
Code obfuscation
Pay attention to the internal information of the program
Reverse Engineering

Figure 2: The relationship between obfuscation algorithm and reverse engineering.

additive homomorphism [21]. FHE has the property of finite homomorphic operations and is more efficient. In addition, it has a shorter ciphertext size.

When using the protocol based on the homomorphic algorithm to check the integrity of the cloud video files, the network bandwidth resources are consumed much less during the execution process. This is because the servers only need to transfer the integrity evidence to users without returning actual video files. Therefore, it enables users to timely detect whether the video files stored in the cloud are corrupted or lost. It saves users more time for data recovery [20]. However, the data integrity verification protocol based on the homomorphic algorithm usually involves multiple large integer exponentiation operations or multiplication operations on the elliptic curve. This fact gives rise to a larger amount of computation. Specifically, for users with limited computing power [22], it takes a long time for homomorphic tags to be generated for video file blocks before uploading video files to the cloud. The computation of the validity of the integrity evidence also requires more time. Although the cloud servers have powerful computing capability, they will consume many resources while performing integrity verification for a number of users.

3. Video Data Integrity Verification Scheme

3.1. Security Model. In the process of checking the integrity of cloud video files using a protocol based on a homomorphic algorithm, the cloud storage server sends users the integrity proof without including a subset of video files or video files after calculation. After receiving the integrity proof, users perform verification locally to determine whether the target data block is intact in the cloud. The Diffie-Hellman system [23], RSA system [24], and bilinear pairings [25] are common homomorphic algorithms in this type of protocol. The execution process of these protocols can be mainly divided into the following 7 steps:

Step 1 (initialize parameters). The user and the cloud server negotiate a set of parameters that are shared by both parties.

Step 2 (initialize keys). Keys are usually asymmetric in the algorithm. The public key is disclosed after the user initializes the key, but the private key is kept by the user.

Step 3 (generate homomorphic tags). The user firstly breaks the video file into blocks with the certain size before uploading the video file to the cloud server. Then the user generates a homomorphic tag locally for each video file block. The video file block and the user's private key are taken as input, and the homomorphic tag is the output.

Step 4 (store video files and tags). The user will store and manage the video file and the tag. Then the user uploads the video file to the cloud for online storage. The local copy is deleted to release the local storage space after the transfer is completed. The homomorphic tag can be stored locally, or it cannot be uploaded to the cloud server until it is encrypted using a symmetric encryption algorithm.

Step 5 (the user initiates a verification challenge). The user generates random numbers locally and constructs a challenge message. Then the user transmits the message to the server.

Step 6 (produce evidence of integrity). The server parses the challenge message and reads the corresponding video file block. The algorithm of producing the integrity evidence consists of three inputs, i.e., the video file block, the challenge message, and the parameter obtained in Step 1. The output is the integrity evidence of the video file block. The server returns the resulting integrity evidence to the challenge initiator.

Step 7 (verify the integrity evidence). The user verifies the legitimacy of the integrity evidence after receipt. The algorithm used in this step usually consists of three inputs, i.e., integrity evidence, homomorphic tag and user public key. The output is a Boolean value, representing whether the integrity evidence is valid.

The formal definition and security definition of data integrity verification are based on full homomorphic encryption. The security model used in this article is shown as follows:

Step 1 (initialize). Challenger runs initialization algorithm and enters related security parameters $k, \lambda_p, \lambda_q, m$, and s. He will obtain the homomorphic key K and private key sk, pass the public key to the opponent. The expression is $\mathrm{KeyGen}(1^k, l_p, l_q, m, s) \longrightarrow (K, sk)$, where m is the number of message sectors and s is a random seed.

Step 2 (generate). This stage is performed by the data owner to generate the tag for the video file. The user inputs the homomorphic key K, the private key sk, and the video file F to output tag set T, which is the sequential set of tags for each block. The expression is $\mathrm{TagGen}(K, sk, F) \longrightarrow T$.

Step 3 (challenge). The data owner executes the algorithm to generate challenge information by blocks as input.

Step 4 (guess). Cloud Storage Service (CSS) executes the algorithm to generate integrity verification by taking inputs of video file, tag set, and challenge.

Step 5 (prove). The data owner executes the algorithm, using the validation p returned by CSS to check the integrity of the video file. The owner takes inputs of the homomorphic key K, the private key sk, challenge $chall$, and verification p. He obtains the output 1 if p is correct, and 0 otherwise. Its expression is $\mathrm{Verify}(K, sk, chall, p) \longrightarrow \{1, 0\}$.

3.2. Video Integrity Verification Method Based on Fully Homomorphic Encryption. The video file is stored in blocks, and the data block is used as the minimum unit in the later stages of label generation and evidence verification. In the initialization phase, a series of initialization parameters are generated for the establishment of the hash function. The encryption is performed using the fully homomorphic encryption function. The algorithm $\mathrm{KeyGen}(\lambda_{\mathrm{p}}, \lambda_{\mathrm{q}}, m, s) \longrightarrow k$ is applied to obtain the homomorphic key $k = (p, q, \overrightarrow{g})$. In the tag generation phase, the client uses a pseudorandom number generator to generate a series of pseudorandom numbers and then multiplies the video file blocks with pseudorandom numbers to obtain the *tag*. The client sends the video file blocks b_i, tag, p, and q to the server but saves the generator $\overrightarrow{g}$, the hash parameter G, and the *seed* used by the pseudorandom number generator. In the challenge phase, the client uses a pseudorandom number generator to generate n random challenge blocks and then sends it to the server. During the evidence generation phase, the server computes evidences b_c and t_c for the data block and label, respectively; they are later returned to client. In the evidence verification phase, the client uses *seed* to regenerate the corresponding pseudorandom number and verifies that the t_c returned by the server is same as the client-specified t_c. It also verifies whether t_c corresponds to the correct b_c. Finally, it is required to conduct security analysis to this verification scheme. In the challenge phase, the challenger randomly generates k challenge blocks and sends them to A. A generates the integrity verification P of the challenge block. If P passes the verification, then A is considered to have completed a successful deception. Suppose A deletes the challenger's data block and it returns any data block and its corresponding label to the challenger. It can be verified that the returned b_c and t_c are the correct counterparts, but A does not know the random number used to construct tag. What the challenger does is to homomorphically hash the received data block and then generate the pseudorandom number with the same seed as the one used to generate tag. The tag is reconstructed and compared with the tag returned by A. The data blocks and tags returned by A are specified by the challenger.

The video file F is represented as a matrix of $m \times n$, and each cell in the matrix is an element in Z_p. The choice of

```
Function KeyGen($\lambda_p, \lambda_q, m, s$).
do
    $q \longrightarrow q \cdot \mathrm{Gen}(\lambda_q)$.
    $p \longrightarrow p \cdot \mathrm{Gen}(q, \lambda_p)$.
while $p = 0$ done
do
    $x \longleftarrow f(p-1)+1$.
    $g_i \longleftarrow x^{(p-1)/q} \bmod p$.
while $g_i = 1$ done
return $(p, q, \overrightarrow{g})$
end
```

Algorithm 1: Initialization parameter generation algorithm.

```
Function $q \cdot \mathrm{Gen}(\lambda_q)$.
do
    $q \longleftarrow f(2^{\lambda_q})$.
while $q$ is not prime done
return $q$.
Function $p \cdot \mathrm{Gen}(q, \lambda_p)$
for $i = 1$ to $4\lambda_p$
    $x \longleftarrow f(2^{\lambda_q})$.
    $c \longleftarrow X \bmod 2q$.
    $p \longleftarrow X - c + 1$.
    if $p$ is prime then return done
        return $p$.
    esle
        return 0.
end.
```

Algorithm 2: Fully homomorphic tag generation algorithm.

guarantees that each element is less than m and therefore less than $2^{\lambda_q - 1}$.It is shown in

$$F = (b_1 b_2 \cdots b_n) = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix}. \qquad (1)$$

The column j of F is only related to the j-th message block of the video file F and is written as $b_j = (b_{1,j}, \ldots, b_{m,j})$. The addition of the 2 video file blocks is to add the corresponding column vectors directly.

$$b_i + b_j = (b_{i,i}, +b_{i,j}, \ldots, b_{m,i} + b_{m,j}) \bmod q. \qquad (2)$$

Algorithm 1 shows the algorithm of initialization parameter generation, while Algorithm 2 shows the algorithm of fully homomorphic tag generation.

3.3. Security Analysis. In order to verify the security of this scheme, a data-holding game is created. If the opponent A wins the game, A can get all ciphertext data blocks and signature label information correctly. The security of this scheme is based on collision resistance of hash function [26] and the difficulty of Diffie-Hellman problem [27].

Theorem 1. *If the hash function and the homomorphic hash function are nonconflicting, the data integrity checking method in the paper is safe.*

Proof. Given the challenged video file F, the file F is divided into n blocks marked as $F = (F_1, F_2, \ldots, F_n)$. Then F_i is divided into m sectors marked as $F_i = (f_{1i}, f_{2i}, \ldots, f_{mi})$. The game between challenger C and opponent A is described as follows.

Step 1 (generate key). The user executes the algorithm KeyGen to obtain the homomorphic key K and the private key sk, both of them are kept in secret by C.

Step 2 (tag query). At any time, the opponent A can query the label of any block F_i $(1 \le i \le n)$. C maintains a list of groups with a value of (i, F_i, T_i), named $Tab1$. When A sends a query label (i, F_i), C will check whether the column of $(i, F_i, *)$ exists in $Tab1$. If $(i, F_i, *) \in Tab1$, then C indexes $(i, F_i, *)$ and returns T_i to A. Otherwise, C computes T_i using TagGen algorithm and adds (i, F_i, T_i) to $Tab1$ and returns T_i to A.

Step 3 (proof verification query). At any time, A can start a certification verification query to C. A adaptively select several blocks. The labels of the blocks are queried from C. A certificate is generated for the selected block. A sends the certificate to C and requests C to response. C calls the Verify algorithm to check the proof and returns the verification result to A.

Step 4 (challenge). C randomly selects two values $k_1, k_2 \in Z_q^*$ and challenge block number C. It is required that each pair (l, F_l) should exist in $Tab1$, where $\mathrm{l} \in \{\pi k_1, \mathrm{i} \mid 1 \le \mathrm{i} \le \mathrm{c}\}$. Then C sends the challenge $chall = \{c, k_1, k_2\}$ to A, and asks A to have proof P of the data of the challenged block.

Step 5 (forgery). A generates a proof $P' = (\overline{F}', \overline{T}')$ based on challenge $chall = \{c, k_1, k_2\}$ and sends it to C, where $\overline{F}' = (\overline{F}'_1, \ldots, \overline{F}'_m)$. A wins if $P' = (\overline{F}', \overline{T}')$ can pass verification. A cannot obtain valid proof if it does not have a challenge block. Then we will prove that if A does not maintain the entire video document, then chances of A winning a data-holding game are negligible.

Step 6 (output). Assuming the opponent A wins, this means that $P' = (\overline{F}', \overline{T}')$ can be proved correct by (3).

If both CSS and the data owner actually perform this scheme, its correctness can be demonstrated as follows:

$$\left(\prod_{i=1}^{c} h_{v_i}^{a_i} \cdot \prod_{t=1}^{m} g_t^{\overline{F_t}}\right)^{sk} \bmod p$$

$$= \left(\prod_{i=1}^{c} h_{v_i}^{a_i} \cdot \prod_{t=1}^{m} g_t^{\sum_{i=1}^{c} a_i f_{tv_i}}\right)^{sk} \bmod p$$

$$= \left(\prod_{i=1}^{c} h_{v_i}^{a_i} \cdot \prod_{i=1}^{c} \prod_{t=1}^{m} g_t^{a_i \cdot f_{tv_i}}\right)^{sk} \bmod p$$

$$
\begin{aligned}
&= \left(\prod_{i=1}^{c}\left(h_{v_i}\cdot\prod_{t=1}^{m} g_t^{f_{tv_i}}\right)^{a_i}\right)^{sk} \bmod p \\
&= \prod_{i=1}^{c}\left(\left(h_{v_i}\cdot\prod_{t=1}^{m} g_t^{f_{tv_i}}\right)^{sk}\right)^{a_i} \bmod p \\
&= \prod_{i=1}^{c}\left(T_{v_i}\right)^{a_i} \bmod p = \overline{T}
\end{aligned}
\tag{3}
$$

□

3.4. Computation Complexity Analysis. Four stages contribute to the computation overhead mainly: tag generation, checking request generation, verification information generation, and integrity verification.

(1) In the tag generation phase, tag information is generated for n blocks of data. The computational complexity is $O(n)$. According to Euler's theorem, $\gcd(e, N)$, then $e^{\phi(N)} \bmod N = 1$. Since modulo operations are much more efficient than exponential operations, only the overhead of the exponentiation operation is considered. Therefore, the computation cost of the tag generation stage is $(n+1)k \times \mathrm{T}_{\exp}(|\mathrm{N}|, \mathrm{N})$, where n is the number of data blocks, $n \times k$ denotes the basic block number, and $\mathrm{T}_{\exp}(\mathrm{len}, \mathrm{num})$ represents the computational time cost of a modulo operation with an exponent of *len* bits and a module of *num* for an integer.

(2) In the checking request generation phase, the computational complexity of two random numbers (r, e) is $O(1)$, and the computational overhead is $T_{prng}(|N| + T_{prng}(k))$. $T_{prng}(len)$ indicates the computational overhead time of generating *len* bits pseudorandom number.

(3) In the verification information generation phase, the computational complexity is $O(n)$. The cloud server first computes $e_r = e^r \bmod N$, which performs a modulo operation with a computing time of $\mathrm{T}_{\exp}(|N|, N)$. Then it is necessary to generate multiple pseudorandom numbers, where $n \times k$ times large multiplication calculations are required in $\sum_{i=1,j=1}^{i=n,j=k} m_{i,j}h(m_{i,j})f_i(j)f(i)$. The length of $f_i(j)$ and m_i is d bits, while the length of $h(m_{i,j})$ is h bits. Each $m_{i,j}h(m_{i,j})f_i(j)f(i)$ and $\sum_{i=1,j=1}^{i=n,j=k} m_{i,j}h(m_{i,j})f_i(j)f(i)$ are calculated. The total computational overhead of the verification information generation phase is $T_{\exp}(|N|, N) + (n \times k + n)T_{prng}(d) + n \times k \times T_{mul}(2d+l+h) + n \times k \times T_{add}(2d+l+h)$, where the computational overhead of $T_{mul}(len)$ represents the multiplication of several *len* bits, and $T_{add}(len)$ represents the computational overhead of the addition of several *len* bits.

(4) The computational complexity of the verification integrity phase is $O(n)$. The cloud storage server requires $n+1$ times of modulo operation and $n-1$ times of modular multiplication operation. The calculated overhead for the entire phase is $(n+1)T_{\exp}(d, N) + (n-1)T_{mul}(|N|, N)$, where $sum \times T_{mul}(len, num)$ denotes the time cost of modularization *num* for *num* integers of *len* bits.

Table 1: The experimental environment.

parameters	values
CPU	2.4GHz Intel(R) Core i7-4712MQ
Internal Memory Storage	8 GB
Operating System (OS)	Windows 7
Exploitation Environment	VMware WorkStation 10

Table 2: File integrity check results.

document names	checking results
A	fail
B	fail
C	success

4. Experimental Results and Analysis

The experiment was run on a PC computer with the configuration shown in Table 1.

This paper uses MIRACL library to implement the prototype of the proposed RDPC scheme in C language, and the implementation is based on Pairing-Based Cryptography (PBC) library and GNU multiarithmetic precision (GMP) library. The homomorphic encryption algorithm is implemented under the framework of VMware WorkStation 10. The scheme in [24] was implemented in simulation for efficiency comparison. Four experiments are performed in the followings according different requirement setup of the proposed scheme.

In order to check the performance of data integrity verification, four metrics are considered, which are security, storage overhead, communication overhead, and computational cost. Security means that each scheme has different security level with different technology. The storage overhead refers to the data block size occupied by metadata in the scheme, and the communication overhead refers to the overhead caused by the communication between the user and the cloud storage server. This mainly exists in the challenging-response link between CSS and TPA. These three conditions determine the computational overhead in data preprocessing stage, integrity proof generation stage, and verification stage.

Experiment 1, three 10MB files (files A, B, and C) are processed, signed, and stored. Then, 10% of the file A is deleted, 10% of the file B is modified, and the file C remains unmodified. Finally, the integrity of the three files is verified. The results are shown in Table 2.

The experimental results show that when the file is deleted or tampered, the integrity of the data cannot be passed by the proposed scheme, but the unmodified file can be verified. It proves the feasibility of this method.

Experiment 2, firstly, a 10MB file is processed and stored in blocks. Then it is tampered with the proportion of 0.1%, 0.2%, 0.4%, 0.6%, 0.8%, 1%, and 1.5%, respectively. Finally, the integrity of the file was verified, and the experiments with each setting were carried out 10, 20, 40, 60, 80, and 100, respectively. The result is shown in Table 3.

The experimental results show that the 10MB file with tampering rate above 0.6% has the success rate of file

Table 3: The relationship between the proportion of tampered files and the efficiency of the algorithm.

Number of experiments	Tampering rate (%)						
	0.1	0.2	0.4	0.6	0.8	1.0	1.5
10	8	9	0	10	10	10	10
20	15	17	19	20	20	20	20
40	32	36	37	38	40	40	40
60	55	57	58	60	60	60	60
80	72	73	75	77	79	80	80
100	91	93	94	97	99	100	100

Table 4: Relationship between data block and efficiency of algorithm.

Number of experiments	Data blocks						
	50	100	150	200	250	300	400
10	7	8	9	10	10	10	10
20	16	17	19	20	20	20	20
40	33	36	37	38	40	40	40
60	52	54	57	59	60	60	60
80	72	73	76	77	79	80	80
100	93	95	96	97	99	100	100

integrity checking which is 100, and the algorithm checking becomes highly efficient with the increase in the number of experiments runs. This suggests that the integrity of the file can be accurately detected.

Experiment 3, firstly, a 100MB file is processed and stored in chunks. Then data blocks of 50, 100, 150, 200, 250, 300, and 400 for multiple integrity verification are conducted. The experimental results are shown in Table 4.

The experimental results show that the data integrity verification which has the highest data integrity verification has the higher number of successful rate with the increase of in number of data blocks and number of experiment runs. This becomes obvious when the number of data blocks is 200 and above.

Experiment 4, firstly, the time cost of establishing the algorithm is evaluated. This is determined mainly by the parameters p and q. The size of the block is set to 16 kb, the size of the sector is set to 256 bits, $|q| = 257$, and $|p| = 512$. Then we mainly consider the installation time cost of different sizes of p. In order to improve accuracy, we run simulations of different cycles from 1 to 100. The results are shown in Figure 3.

Experimental results show that when $|p| = 512$ and $|p| = 1024$, the installation time costs are relatively stable at 0.16s and 1.28s, respectively. The cost is acceptably low. An experimental evaluation of the computational cost of tag generation is performed. In the experiment, $|p| = 1024$, $|q| = 257$. In both scenarios, each block has the same number of sectors. The result is shown in Figure 4.

Experimental results show that the relationship between the computational cost and number of sectors for two schemes is approximately linear. For example, comparison scheme results in the time cost of about 1.4s to generate a block label with 512 sectors, and the proposed scheme only needs 0.42s. In addition, the computational cost of the label generation in the comparison program is significantly higher than the proposed program with the increase of number of sectors. Therefore, it can be seen that our proposed scheme is more computational cost effective and feasible.

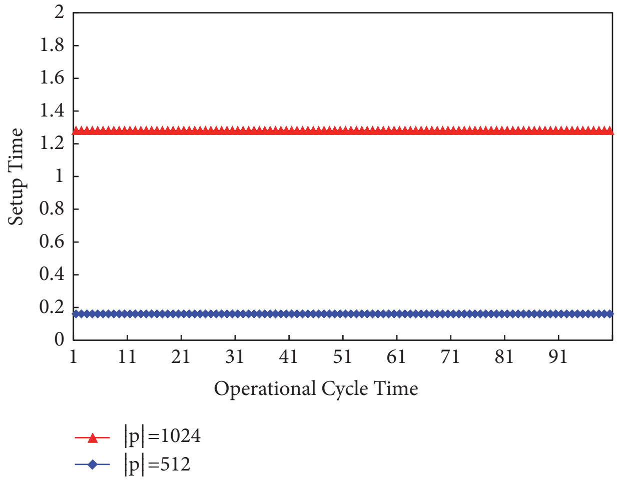


Figure 3: The setup time cost.

Finally, the number of sectors per block is set to 512. Since the computational cost of proof generation and verification is mainly determined by the number of challenged blocks, comparison experiments are conducted for different number of challenged blocks. Figures 5 and 6 show the computational cost of proof generation and verification when the parameters are set to $|p| = 1024$, $|q| = 257$.

The experimental results show that the cost of verification and proof generation rises with the increase of the number of challenged blocks. Our proposed scheme has a greater

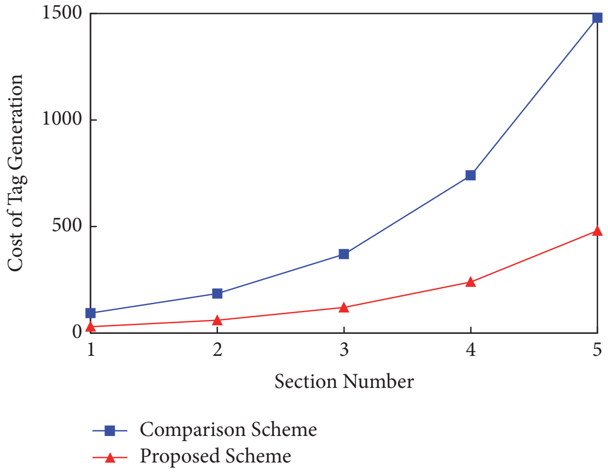


Figure 4: Tag generation computational cost.

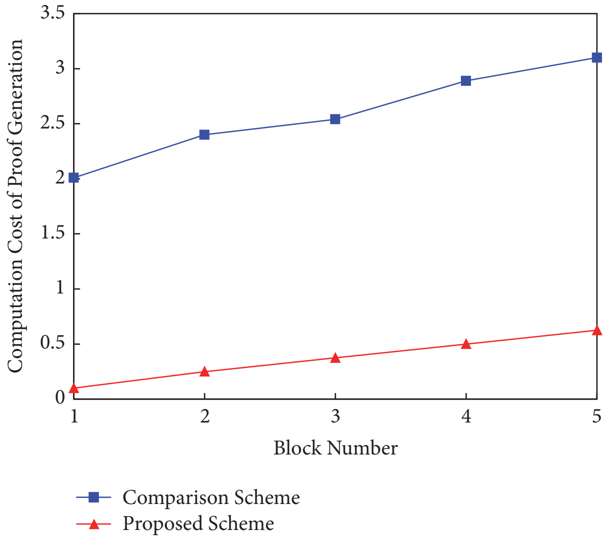


Figure 5: Computation cost of proof generation.

advantage to the comparison scheme in terms of computation cost, especially with the increase in the number of blocks. When the number of blocks challenged is less than approximately 220, the cost of our scheme is slightly greater than the comparison scheme in Figure 6. However, with the increase of the number of challenge blocks, the overhead of the comparison scheme has grown rapidly, exceeding the proposed scheme. It greatly exceeds the proposed scheme. According to studies, 1% of the errors per 460 blocks occurs for a 1GB video file. This gives rise to a confidence level of 99%. In the comparison scheme, in order to challenge 460 blocks, the proof generation takes 3.1s and the verification takes 1.2s, respectively. In our scheme, they only take 0.52s and 0.8s, respectively. Therefore, our proposed scheme is more feasible.

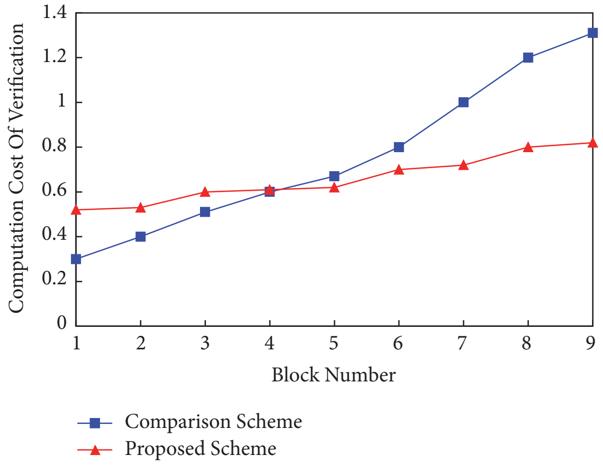


Figure 6: Computation cost of verification.

5. Conclusion

Cloud services have exploded in the era of cloud computing, and various intrusion activities have put information security at risk. This paper studies the integrity of video data in cloud systems, and we propose a method for verification of video data integrity based on full homomorphic encryption. Firstly, the homomorphic encryption technology is used to initialize the video data, which reduces the time complexity. Secondly, the feasibility of the method was verified through security analysis and performance analysis. The final simulation results show that the proposed scheme is superior to comparison schemes in all aspects, and it suggests that the proposed scheme is serving better for the video data integrity verification purpose in the cloud environment.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61602155 and no. 61370221 and in part by the Industry University Research Project of Henan Province under Grant no. 172107000005.

References

[1] W. Quan, Y. Liu, H. Zhang, and S. Yu, “Enhancing crowd collaborations for software defined vehicular networks,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 80–86, 2017.

[2] B. Feng, H. Zhang, H. Zhou, and S. Yu, “Locator/Identifier Split Networking: A Promising Future Internet Architecture,” *IEEE*

Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2927–2948, 2017.

[3] H. Zhang, W. Quan, H.-C. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future internet," *IEEE Network*, vol. 30, no. 3, pp. 46–51, 2016.

[4] C. Yuan, Z. Xia, and X. Sun, "Coverless image steganography based on SIFT and BOF," *Journal of Internet Technology*, vol. 18, no. 2, pp. 209–216, 2017.

[5] F. Song, Z. Ai, J. Li et al., "Smart Collaborative Caching for Information-Centric IoT in Fog Computing," *Sensors*, vol. 17, no. 11, p. 2512, 2017.

[6] Q. Wu, M. Zhang, R. Zheng, Y. Lou, and W. Wei, "A QoS-Satisfied Prediction Model for Cloud-Service Composition Based on a Hidden Markov Model," *Mathematical Problems in Engineering*, vol. 2013, Article ID 387083, 7 pages, 2013.

[7] J. Li, W. Yao, Y. Zhang, H. L. Qian, and J. G. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[8] Q. Wu, X. Zhang, M. Zhang, Y. Lou, R. Zheng, and W. Wei, "Reputation Revision Method for Selecting Cloud Services Based on Prior Knowledge and a Market Mechanism," *The Scientific World Journal*, vol. 2014, Article ID 617087, 9 pages, 2014.

[9] Z. Fu, K. Ren, and J. Shu, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[10] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 97–106, Palm Springs, Calif, USA, October 2011.

[11] R. Zheng, J. Chen, M. Zhang, Q. Wu, J. Zhu, and H. Wang, "A collaborative analysis method of user abnormal behavior based on reputation voting in cloud environment," *Future Generation Computer Systems*, vol. 83, pp. 60–74, 2018.

[12] M. Zhang, M. Yang, Q. Wu, R. Zheng, and J. Zhu, "Smart perception and autonomic optimization: A novel bio-inspired hybrid routing protocol for MANETs," *Future Generation Computer Systems*, vol. 81, pp. 505–513, 2018.

[13] M. Zhang, C. Xu, J. Guan, R. Zheng, Q. Wu, and H. Zhang, "A Novel *Physarum*-Inspired Routing Protocol for Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 483581, 12 pages, 2013.

[14] R. L. Rivest, L. Adleman, and M. L. Dertouzos, *On Data Banks And Privacy Homomorphism Proc of Foundations of Secure Computation*, Academic Press, New York, NY, USA, 1978.

[15] M. Liu and W. An, "Fully Homomorphic Encryption and Its Application," *Journal of Computer Research & Development*, vol. 51, no. 12, pp. 2593–2603, 2014.

[16] H. Yan, G. Chen, and T. Han, "Scope of application of homomorphic encryption algorithm and improvement of efficiency and application," *Computer Engineering and Design*, vol. 38, no. 2, pp. 318–322, 2017.

[17] H. Demin and Y. Xing, "Dynamic cloud storage data integrity verifying method based on homomorphic tags," *Application Research of Computers*, vol. no. 5, pp. 1362–1365, May 2014.

[18] Y. Zhu, H. Wang, Z. HU et al., *Cooperative Provable Data Possession*, Peking University and Arizona University, Beijing, China, 2010.

[19] X. Cao, C. Moore, M. O'Neill, E. O'Sullivan, and N. Hanley, "Optimised multiplication architectures for accelerating fully homomorphic encryption," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 9, pp. 2794–2806, 2016.

[20] J. Chen, H. Ma, and D. Zhao, "Private data aggregation with integrity assurance and fault tolerance for mobile crowd-sensing," *Wireless Networks*, vol. 23, no. 1, pp. 131–144, 2017.

[21] S. Wang, J. Zhou, and J. Liu, "An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing," *IEEE Transactions on Information Forensics Security*, vol. 11, no. 6, pp. 1265–1277, 2016.

[22] A. Li, S. Tan, and Y. Jia, "A method for achieving provable data integrity in cloud computing," *The Journal of Supercomputing*, pp. 1–17, 2016.

[23] Y. Yu, M. H. Au, G. Ateniese et al., "Identity-Based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, 2017.

[24] Q. Li, J. Ma, R. Li et al., "Secure, efficient and revocable multi-authority access control system in cloud storage," *Computers & Security*, vol. 59, no. C, pp. 45–59, 2016.

[25] L. Ferretti, M. Marchetti, M. Andreolini, and M. Colajanni, "A symmetric cryptographic scheme for data integrity verification in cloud databases," *Information Sciences*, vol. 422, pp. 497–515, 2018.

[26] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Computing Surveys*, vol. 47, no. 4, article 63, 2015.

[27] K. Xue, Y. Xue, J. Hong et al., "RAAC: Robust and Auditable Access Control with Multiple Attribute Authorities for Public Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 953–967, 2017.

Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2018, Article ID 4546896, 10 pages
https://doi.org/10.1155/2018/4546896

Research Article

Small Object Detection with Multiscale Features

Guo X. Hu,[1,2] Zhong Yang ,[1] Lei Hu,[3] Li Huang,[4] and Jia M. Han[1]

[1] *College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2] *School of Software, Jiangxi Normal University, Nanchang 330022, China*
[3] *School of Computer Information Engineering, Jiangxi Normal University, Nanchang 330022, China*
[4] *Elementary Education College, Jiangxi Normal University, Nanchang 330022, China*

Correspondence should be addressed to Zhong Yang; yz.nuaa@163.com

Received 15 July 2018; Accepted 13 September 2018; Published 30 September 2018

Guest Editor: Wei Quan

The existing object detection algorithm based on the deep convolution neural network needs to carry out multilevel convolution and pooling operations to the entire image in order to extract a deep semantic features of the image. The detection models can get better results for big object. However, those models fail to detect small objects that have low resolution and are greatly influenced by noise because the features after repeated convolution operations of existing models do not fully represent the essential characteristics of the small objects. In this paper, we can achieve good detection accuracy by extracting the features at different convolution levels of the object and using the multiscale features to detect small objects. For our detection model, we extract the features of the image from their third, fourth, and 5th convolutions, respectively, and then these three scales features are concatenated into a one-dimensional vector. The vector is used to classify objects by classifiers and locate position information of objects by regression of bounding box. Through testing, the detection accuracy of our model for small objects is 11% higher than the state-of-the-art models. In addition, we also used the model to detect aircraft in remote sensing images and achieved good results.

1. Introduction

Object detection, which not only requires accurate classification of objects in images but also needs accurate location of objects is an automatic image detection process based on statistical and geometric features. The accuracy of object classification and object location is important indicators to measure the effectiveness of model detection. Object detection is widely used in intelligent monitoring, military object detection, UAV navigation, unmanned vehicle, and intelligent transportation. However, because of the diversity of the detected objects, the current model fails to detect objects. The changeable light and the complex background increase the difficulty of the object detection especially for the objects that are in the complex environment.

The traditional method of image classification and location by multiscale pyramid method needs to extract the statistical features of the image in multiscale and then classify the image by a classifier [1–3]. Because different types of images are characterized by different features, it is difficult to use one or more features to represent objects, which do not achieve a robust classification model. Those models failed to detect the objects especially that there are more detected objects in an image.

Since deep learning has been a great success in the field of object detection, it has become the mainstream method for object detection. These methods (e.g., RCNN [4], Fast-RCNN [5], Faster-RCNN [6], SPP-Net [7], and R-FCN [8]) have achieved good results in multiobject detection in images. But most of these object detection algorithms are based on PASCAL VOC dataset [9] for training and testing. PASCAL VOC dataset, which provides a standard evaluation system for detection algorithms and learning performance, is the most widely used standard dataset in the field of object classification and detection. The dataset consists of 20 catalogues closely related to human life, including human and animal (bird, cat, cattle, dog, horse, and sheep), vehicle (aircraft, bicycle, ship, bus, car, motorcycle, and train), and indoor item (bottle, chair, table, potted plants, sofa, and television). From the above object category, we can find that the actual size of most objects in the dataset is large object. Even if there are some small objects, such as bottles, these small objects

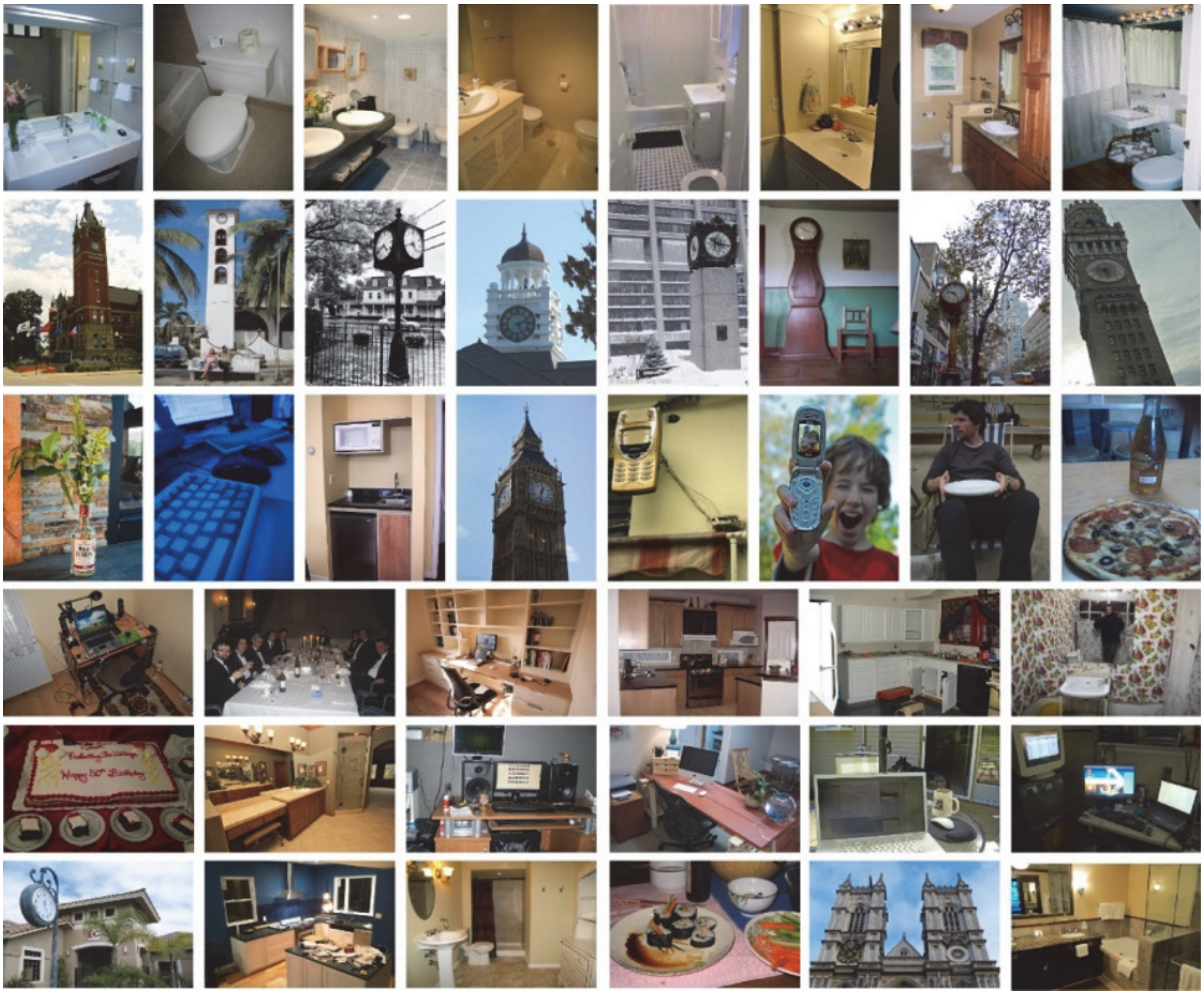


Figure 1: Small object dataset.

display very large objects in the image because of the focal length. Therefore, the detection model based on the dataset composed of large objects will not be effectively detected for the small objects in reality [10].

Based on this problem, we mainly study automatic detection of small object. For small object, we define it as two types: one is a small object in the real world, such as mouse and telephone. And the other is small objects; those are large objects in the real world, but they are shown in the image as small objects because of the camera angle and focal length, such as objects detection in aerial images or in remote sensing images. The small object dataset is shown in Figure 1.

Usually, since small objects have low resolution and are near large objects, small objects are often disturbed by the large objects and it leads to failure in being detected in the automatic detection process. As the mouse in Figure 1 is often placed next to the monitor, the common saliency detection model [11, 12] usually focuses on more significant monitor and ignores the mouse. In addition, we not only find the detected objects in the image but also need to accurately mark object location for object detection. Because the big detected objects have many pixels in the image, they can accurately locate their location. But, it is just the opposite for the small objects that have low resolution and few pixels. Even more, because the small objects have fewer pixels and the finite pixels contain few object features, it is difficult to detect the small objects by the conventional detection model. In addition, there are few studies, references, and also no standard dataset on automatic detection of small objects.

In order to solve these problems, we propose a multiscale deep convolution detection network to detect small objects. The network is based on the Faster-RCNN detection model. We firstly combine the features of the 3*th*, 4*th,* and 5*th* convolution layers for the small objects to a multiscale feature vector. Then, we use the vector to detect the small objects and locate the bounding box of objects. In order to train small objects, the paper also uses the method [13] to build a dataset focusing on small objects. Finally, by comparing the proposed detection model with the state-of-the-art detection model, we find that the accuracy of our method is much better than that of Faster-RCNN.

The paper is organized as follows. In Section 2, we introduce related works. Thereafter in the Section 3, we demonstrate the detection model. Experiments are presented in Section 4. We conclude with a discussion in Section 5.

2. Related Works

Object detection is always a hot topic in the field of machine vision. The conventional detection method based on sliding window needs to decompose images in multiscale images. Usually, one image is decomposed into lots of subwindows of several million different locations and different scales. The model then uses a classifier to determine whether the detected object is contained in each window. The method is very inefficient because it needs exhaustive search. In addition, different classifiers also affect the detection accuracy of objects. In order to obtain robust classifier, the classifiers are designed according to the different kinds of detected objects. For example, the Harr feature combined with Adaboosting classifier [14] is availability for face detection. For pedestrian detection, we use the HOG feature (Histogram of Gradients) combined with support vector machine [15] and the HOG feature combined with DPM (Deformable Part Model) [16, 17] is often used in the field of the general object detection. However, if there are many different kinds of detected objects in an image, those classifiers will fail to detect the objects.

Since 2014, Hinton used deep learning to achieve the best classification accuracy in the year's ImageNet competition, and then the deep learning has become a hot direction to detect the objects. The model of object detection based on the deep learning is divided into two categories: the first that is widely used is based on the region proposals [18–20], such as RCNN [4], SPP-Net [7], Fast-RCNN [5], Faster-RCNN [6], and R-FCN [8]. The other method does not use region proposals but directly detects the objects, such as YOLO [21] and SSD [22].

For the first method, the model firstly performs RoIs selection during the detection; i.e., multiple RoI*s* are generated by selective search [23], edge box [24], or RPN [25]. Then the model extracts features for each RoI*s* by CNN, classifies objects by classifiers, and finally obtains the location of detected objects. RCNN [4] uses selective search [23] to produce about 2000 RoI*s* for each picture and then extracts and classifies the convolution features of the 2000 RoI*s*, respectively. Because these RoI*s* have a large number of overlapped parts, the large number of repeated calculations results in the inefficient detection. SSP-net [7] and Fast-RCNN [5] propose a shared RoI*s* feature for this problem. The methods extract only a CNN feature from the whole original image, and then the feature of each RoI is extracted from the CNN feature by RoI pooling operation independently. So the amount of computing of extracting feature of each RoI is shared. This method reduces the CNN operation that needs 2000 times in RCNN to one CNN operation, which greatly improves the computation speed.

However, whether it is SSP-net or Fast-RCNN, although they reduce the number of CNN operations, its time consumption is far greater than the time of the CNN feature extraction on GPU because the selection of the bounding box of each object requires about 2 seconds/image on CPU. Therefore, the bottleneck of the object detection lies in region proposal operation. Faster-RCNN inputs the features extracted by CNN to the RPN (Region Proposal Network) network and obtains region proposal by the RPN network, so it can share the image features extracted by CNN and thereby it reduces the time of selective search operation. After RPN, Faster-RCNN classifies the obtained region proposal through two fully connected layers and the regression operation of the bounding box. Experiments prove that not only is this speed faster, but also the quality of proposal is better. R-FCN thinks that the full connection classification for each RoI by Faster-RCNN is also a very time-consuming process, so R-FCN also integrates the classification process into the forward computing process of the network. Since this process is shared for different RoI, it is much faster than a separate classifier.

The other type is without using region proposal for the object detection. YOLO divides the entire original image into the S$*$S cell. If the center of an object falls within a cell, the corresponding cell is responsible for detecting the object and setting the confidence score for each cell. The score reflects the probability of the existence of the object in the bounding box and the accuracy of IoU. YOLO does not use region proposal, but directly convolution operations on the whole image, so it is faster than Faster-RCNN in speed, but the accuracy is less than Faster-RCNN. SSD also uses a single convolution neural network to convolution the image and predicts a series of boundary box with different sizes and ratio of length and width at each object. In the test phase, the network predicts the possibility of each class of objects in each bounding box and adjusts the boundary box to adapt to the shape of the object. G-CNN [25] regards object detection as a problem of changing the detection box from a fixed grid to a real box. The model firstly divides the entire image with different scale to obtain the initial bounding box and extracts the features from the whole image by the convolution operation. Then the feature image encircled by an initial bounding box is adjusted to a fixed size feature image by the method Fast-RCNN mentioned. Finally we can obtain a more accurate bounding box by regression operation. The bounding box will be the final output after several iterations.

In short, for the current mainstream there are two types of object detection methods, the first will have better accuracy, but the speed is slower. The accuracy of the second one is slightly worse, but faster. No matter which way to carry out the object detection, the feature extraction uses multilayer convolution method, which can obtain the rich abstract object feature for the target object. But this method leads to a decrease in detection accuracy for small target objects because the features extracted by the method are few and can not fully represent the characteristics of the object.

In addition, the PASCAL VOC dataset is the main dataset for object detection and it is composed of 20 categories of object, e.g., cattle, buses, and pedestrians. But all of these objects in the image are large objects. Even in the PASCAL VOC, there are also some small objects, e.g., cup, but these small objects display very large objects in the image because of the focal length. So, the PASCAL VOC is not suitable for the detection of small objects.

Microsoft COCO dataset [26] is a standard dataset built by Microsoft team for object detection, image segmentation, and other fields. The dataset includes various types of small objects with the complexity of the background, so it is suitable for small objects detection. The SUN dataset [27] consists of 908 scene categories and 4479 object categories and a total of 131067 images that also contain a large number of small objects.

In order to get the rich small object dataset, the paper [13] adopted two standards to build the dataset. The first is that the actual size of the objects is not more than 30 centimeters. Another criterion is that the area occupied of the objects is not more than 0.58% in the image. The author also gives the mAP of RCNN based on the dataset and it has only 23.5% detection rate.

3. Model Introduction

3.1. Faster-RCNN. The RCNN model proposed by Girshick in 2014 is divided into four processes during the object detection. First, 2000 proposal regions in the image are obtained by region proposal algorithm. Second, it extracts the CNN features of the two thousand proposal regions separately and outputs the fixed dimension features. Third, the objects are classified according to the features. Finally, in order to get the precise object bounding box, RCNN accurately locate and merge the foreground objects by regression operation. The algorithm has achieved the best accuracy of the year. But it requires an additional expense on storage space and time because RCNN needs to extract the features of 2000 proposal regions in each image. Later, Fast-RCNN is proposed by Girshick based on RCNN, the model, which maps all proposal regions into one image and has only one feature extraction. So Fast-RCNN greatly improves the speed of detection and training. However, Fast-RCNN still needs to extract the proposal regions which is the same as RCNN. The proposal regions extracted lead to inefficiency. Faster-RCNN integrates the generation of proposal region, extracting feature of proposal region, detection of bounding box, and classification of object into a CNN framework by the RPN network (region proposal network). So it greatly improves the detection efficiency. The RPN network structure diagram is shown in Figure 2. The core idea of Faster-RCNN is to use the RPN network to generate the proposal regions directly and to use the anchor mechanism and the regression method to output an objectness score and regressed bounds for each proposal region; i.e., the classification score and the boundary of the 3 different scales and 3 length-width ratio for each proposal region are outputted. Experiments show that the VGG-16 model takes only 0.2 seconds to detect each image. In addition, it has been proved that the detection precision will be reduced if the negative sample is very high in the dataset. The RPN network generates 300 proposal regions for each image by multiscale anchors, which are less than 2000 proposal regions of Fast-RCNN or RCNN. So the accuracy is also higher than them.

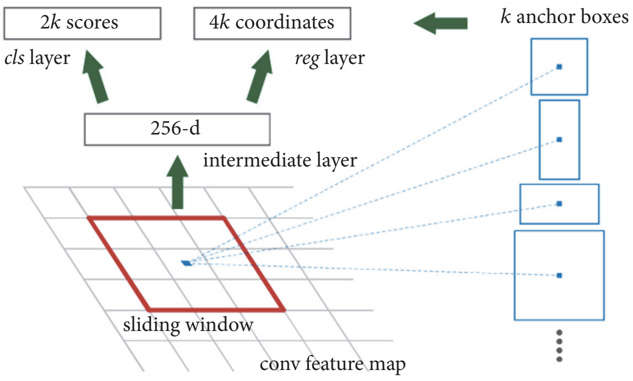


Figure 2: RPN network structure.

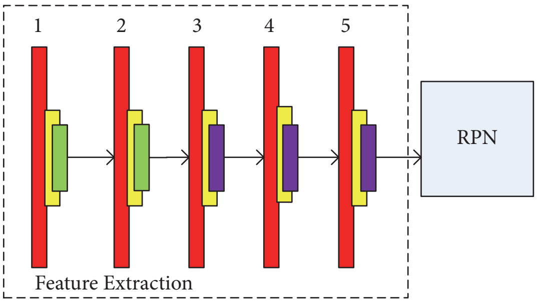


Figure 3: Faster-RCNN network structure.

Faster-RCNN only provides a RPN layer improvement compared to the Fast-RCNN network and does not improve the feature mapping layer compared to the Fast-RCNN network. Faster-RCNN network structure is shown in Figure 3. Faster-RCNN performs multiple downsampling operations in the process of feature extraction. Each sampling causes the image to be reduced by half. The output image in the fifth layer is the 1/16 of the original object for Faster-RCNN; i.e., only 1 byte feature is outputted on the last layer if the detected object is smaller than 16 pixels in the original image. The objects failed to be detected because little feature information can not sufficiently represent the characteristics of the object.

Although Faster-RCNN has achieved very good detection results on the PASCAL VOC, the PASCAL VOC is mainly composed of large objects. The detection precision will fall if the dataset is mainly composed of small objects.

3.2. Multiscale Faster-RCNN. In reality, the detected objects are low in resolution and small in size. The current model (e.g., Faster-RCNN) which has good detection accuracy for large objects can not effectively detect small objects in the image [28]. The main reason is that those models based on deep neural network make the image calculated with convolution and downsampled in order to obtain more abstract and high-level features. Each downsampling causes the image to be reduced by half. If the objects are similar to the size of the objects in the PASCAL VOC, the object's detail features can be obtained through these convolutions and

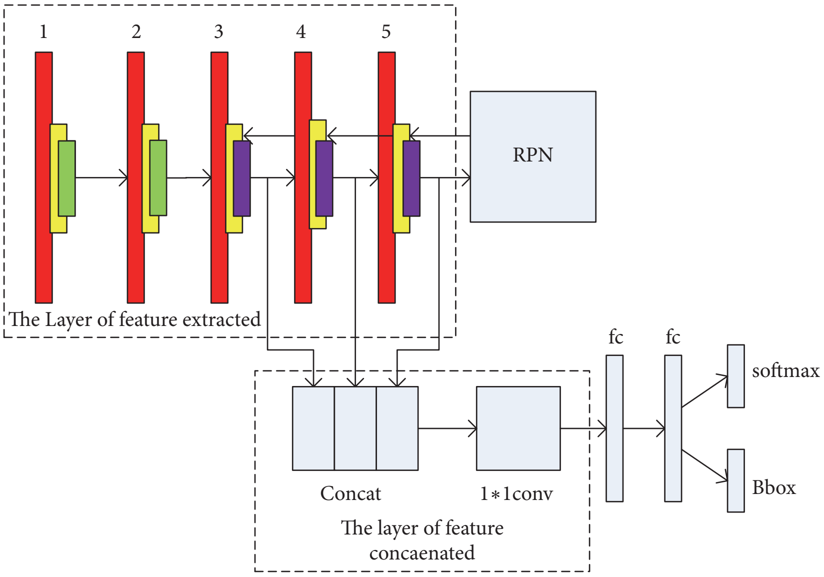

1
2
3
4
5
RPN
The Layer of feature extracted
fc
fc
softmax
Bbox
Concat
1∗1conv
The layer of feature concaenated

Figure 4: Our model structure.

downsampling. However, if the detected objects are the very small scale, the final features may only be left 1-2 pixels after multiple downsampling. So few features can not fully describe the characteristics of the objects and the existing detection method can not effectively detect the small target object.

The deeper the convolution operation, the more abstract the object features which can represent the high-level features of objects are. The shallow convolution layer can only extract the low-level features of objects. But for small objects, the low-level features can ensure rich object characteristics. In order to get high-level and abstract object features and ensure that there are enough pixels to describe small objects, we combine the features of different scales to ensure the local details of the object. At the same time, we also pay attention to the global characteristics of the object based on the Faster-RCNN. This model will have more robust characteristics. The model structure is shown in Figure 4.

The model is divided into four parts: the first part is the feature extraction layer which consists of 5 convolution layers (red part), 5 ReLU layers (yellow parts), 2 pooling layers (green parts), and 3 RoI pooling layers (purple part). We normalize the output of the 3th, 4th, and 5th convolution, respectively. Then the normalized output is sent to the RPN layer and the feature combination layer for the generation of proposal region and the extracted multiscale feature, respectively. The second part is the feature combination layer that combines the different scales features of third, fourth, and fifth layer into one-dimension feature vector by connection operation. The third part is the RPN layer which mainly realizes the generation of proposal regions. The last layer, which is used to realize classification and bounding box regression of objects that are in proposal regions, is composed of softmax and BBox.

3.3. L2 Normalization. In order to obtain the combinatorial feature vectors, we need to normalization the feature vectors of different scales. Usually the deeper convolution layer outputs the smaller scale features. On the contrary, the lower convolution layer outputs the larger scale features. The feature scales of different layers are very different. The weight of large-scale features will be much larger than that of small scale features during the network weight which is tuned if the features of these different scales are combined, which leads to the lower detection accuracy.

To prevent such large-scale features from covering small scale features, the feature tensor that is outputted from different RoI pooling should be normalized before those tensors are concatenated. In this paper, we use L2 normalization. The normalization operation, which is used to process every feature vector that is pooled, is located after RoI pooling. After normalization, the scale of the feature vectors of the $3th$,$4th$, and $5th$ layer will be normalized into a unified scale.

$$\widehat{X} = \frac{X}{\|X\|_2}, \tag{1}$$

$$\|X\|_2 = \left(\sum_{i=1}^{d} |x_i|\right)^{1/2}, \tag{2}$$

where X is the original vector from the $3th$, $4th$, and $5th$ layer, $\widehat{\mathrm{X}}$ is normalized feature vector, and D is the channel number of each RoI pooling.

The vector X will be uniformly scaled by scale facto; i.e.,

$$Y = \gamma \widehat{X}, \tag{3}$$

where $Y = [y_1, y_2, \ldots, y_d]^T$.

In the process of error back propagation, we need to further adjust the scale factor γ and input vector X. The specific definition is as follows:

$$\frac{\partial l}{\partial \widehat{X}} = \gamma \frac{\partial l}{\partial y}, \tag{4}$$

Table 1: The process of model training.

Training process
Input: VGG_CNN_M_1024 and image
Output: detection model
Step 1 Initialize the ImageNet pre training model VGG_CNN_M_1024 and train the RPN network.
(1) Initialize network parameters using pre training model parameters
(2) Initialization of caffe
(3) Prepare for roidb and imdb
(4) Set output path to save the caffe module of intermediate generated.
(5) Training RPN and save the weight of the network
Step 2 Using the trained RPN network in step 1, we generate the ROIs information and the probability distribution of the foreground objects in the proposal regions.
Step 3 First training Fast RCNN network
(1) The proposal regions got from step 2 are sent to the ROIs
(2) The probability distribution of foreground objects is sent to the network as the weight of the objects in the proposal regions
(3) By comparing the size of Caffe blob, we get the weight of objects outside the proposal regions
(4) The loss-cls and loss-box loss functions are calculated, classify and locate objects, obtain the detection models.
Step 4 Replace the detection model obtained in step 3 with the ImageNet network model in step 1, repeat steps 1 to 3, and the final model is the training model.

$$\frac{\partial l}{\partial X} = \frac{\partial l}{\partial \widehat{X}} \left(\frac{I}{\|X\|_2} - \frac{XX^T}{\|X\|_2^3} \right), \tag{5}$$

$$\frac{\partial l}{\partial \gamma} = \sum_{y} \frac{\partial l}{\partial y} \widehat{X}. \tag{6}$$

3.4. Concat Layer. After the features of the third, fourth, and fifth layer are L2 normalized and RoI pooled, output vectors need to be concatenated. The concatenation operation consists of four tuples (i.e., number, channel, height, and weight), where number and channel represent the concatenation dimension and height and weight represent the size of concatenation vectors. All output of each layer will be concatenated into a single dimension vector by concatenation operations. In the initial stage of model training, we set a uniform initial scale factor of 10 for each RoI pooling layer [11] in order to ensure that the output values of the downstream layers are reasonable.

Then in order to ensure that the input vector of the full connection has the same scale as the input vector of the Faster-RCNN, an additional 1∗1 convolution layer is added to the network to compress the channel size of the concatenated tensor to the original one, i.e., the same number as the channel size of the last convolution feature map (conv5).

3.5. Algorithmic Description. Faster RNN provides two training methods with end-to-end training and alternate training and also provides three pretraining networks of different sizes with VGG-16, VGG_CNN_M_1024, and ZF, respectively. The large network VGG-16 has 13 convolutional layers and 3 fully connected layers. ZF net that has 5 convolutional layers and 3 fully connected layers is small network and the VGG_CNN_M_1024 is medium-sized network. Experiment shows that the detection accuracy of VGG-16 is better than the other two models, but it needs more than 11G GPU. In order to improve the training speed of the model, we use the VGG_CNN_M_1024 model as a pretraining model and use the alternation training as a training method. The main process of training is shown in Table 1.

4. Experimental Analysis

4.1. Dataset Acquisition. At present, the dataset commonly used in target detection is PASCAL VOC, which is made up of larger objects or the objects whose size is very small but the area of the objects in the image is very large because of the focal length. Therefore, PASCAL VOC is not suitable for small object detection. There is no dataset for small target objects. In order to test the detection effect of the model on small objects, the paper will establish a small object dataset for object detection based on Microsoft COCO datasets and SUN datasets.

In the process of building small object dataset, we refer to the two criteria mentioned in [18]. The first criterion is that the actual size of the detected object is not more than 30 centimeters. The second criterion is that all the small objects in the image occupy 0.08% to 0.58% of the area in the image; i.e., the pixels of the object are between 16∗16 and 42∗42 pixels. The small objects in the PASCAL VOC occupy 1.38% and 46.40% of the area in the image, so it is not suitable for small object detection. The statistics table is shown in Table 2 [18].

Based on the above standards, we select 8 types of objects to make up a dataset, including mouse, telephone, outlet, faucet, clock, toilet paper, bottle, and plate. After filtering COCO and SUN dataset, we finally select 2003 images that include a total of 3339 objects. The 358 mouse are distributed in 282 images, and the other objects, e.g., toilet paper, faucet, socket panel, and clock, are shown in Table 3.

Table 2: PASCAL VOC object area account table. Unit: %.

category	cat	sofa	train	dog	table	motorbike	horse
area ratio	46. 40	33.87	32.33	30.96	23.73	23.69	23.15
category	bus	plane	bicycle	person	bird	cow	chair
area ratio	23.04	22.83	14.38	8.14	8.03	6.68	6.09
category	TV	boat	sheep	plant	car	bottle	
area ratio	5.96	3.82	3.34	2.92	2.79	1.38	

Table 3: The small object dataset.

category	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Number of images	282	265	305	423	387	245	209	353
Number of objects	358	332	477	515	422	289	371	575

Table 4: The comparison of accuracy between our model and Faster-RCNN. (40000,20000).

model	mAP	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Faster RCNN	0.479	0.360	0.409	0.519	0.392	0.643	0.350	0.485	0.676
Our model	0.589	0.402	0.482	0.600	0.506	0.687	0.641	0.585	0.806

Table 5: The comparison of accuracy between our model and Faster-RCNN (60000,30000).

model	mAP	Mouse	Telephone	Outlet	Faucet	Clock	Toilet paper	bottle	plate
Faster RCNN	0.491	0.447	0.449	0.549	0.424	0.604	0.309	0.428	0.719
Our model	0.587	0.371	0.564	0.572	0.561	0.690	0.546	0.514	0.880

The small object dataset established in this paper is based on COCO and SUN. Because the data in COCO and SUN are mainly based on the scenes of everyday life, the complexity of image background in our dataset is much larger than that in PASCAL VOC. In addition, there are more objects in single image compared with the PASCAL VOC, and most of these objects are not in the image center. These make the object detection based on the small object dataset more difficult than that based on the PASCAL VOC.

During the experiment, we randomly select 300 images as a test set and 600 as a validation set from the small dataset, and all the remaining images are trained as a training set.

4.2. Experimental Comparison. The paper compares our model with the state-of-the-art detection model Faster-RCNN for small object detection. In the process of model training, our model and Faster-RCNN model use the alternate training method. Firstly, we train the RPN network and use the RPN network as a pretraining network to train the detection network. Then we repeat the above steps to get the final detection model. In the training process, we have 40000 iterations for the RPN network and 20000 iterations for the detection network. The final accuracy of the detection is shown in Table 4.

With the increase in the number of iterations of the training network, different models will show different detection results. In the experiment, we also try to increase the number of iterations; that is, the RPN network iterates 60000 times and the detection network iterates 30000 times. The results obtained are shown in Table 5.

We can find that the detection accuracy is stable when the number of iterations of RPN network is 40000 and the number of iterations of detection network is 20000 from the above experiments. The accuracy of our model is better than that of Faster-RCNN for all types of objects. The part renderings of the objects detection are shown in Figure 5.

In order to further detect the robustness of the model, we also detect the remote sensing images in real environment. The remote sensing image dataset comes from the Google map and the insulators of the field transmission line are photographed by the UAV (unmanned aerial vehicle). Because the images in real environment have the characteristics of changeable light, complex background, and incomplete objects, we try to take all the special cases into consideration during the building the dataset. Experiments show that our proposed detection model has better detection results in small objects detection in real environment. The part renderings of the objects detection are shown in Figure 6.

5. Conclusions

Small objects are very difficult to detect because of their lower resolution and larger influence of the surrounding environment. The existing detection models based on deep neural network are not able to detect the small objects because the features of objects that are extracted by many convolution and pooling operations are lost. Our model not

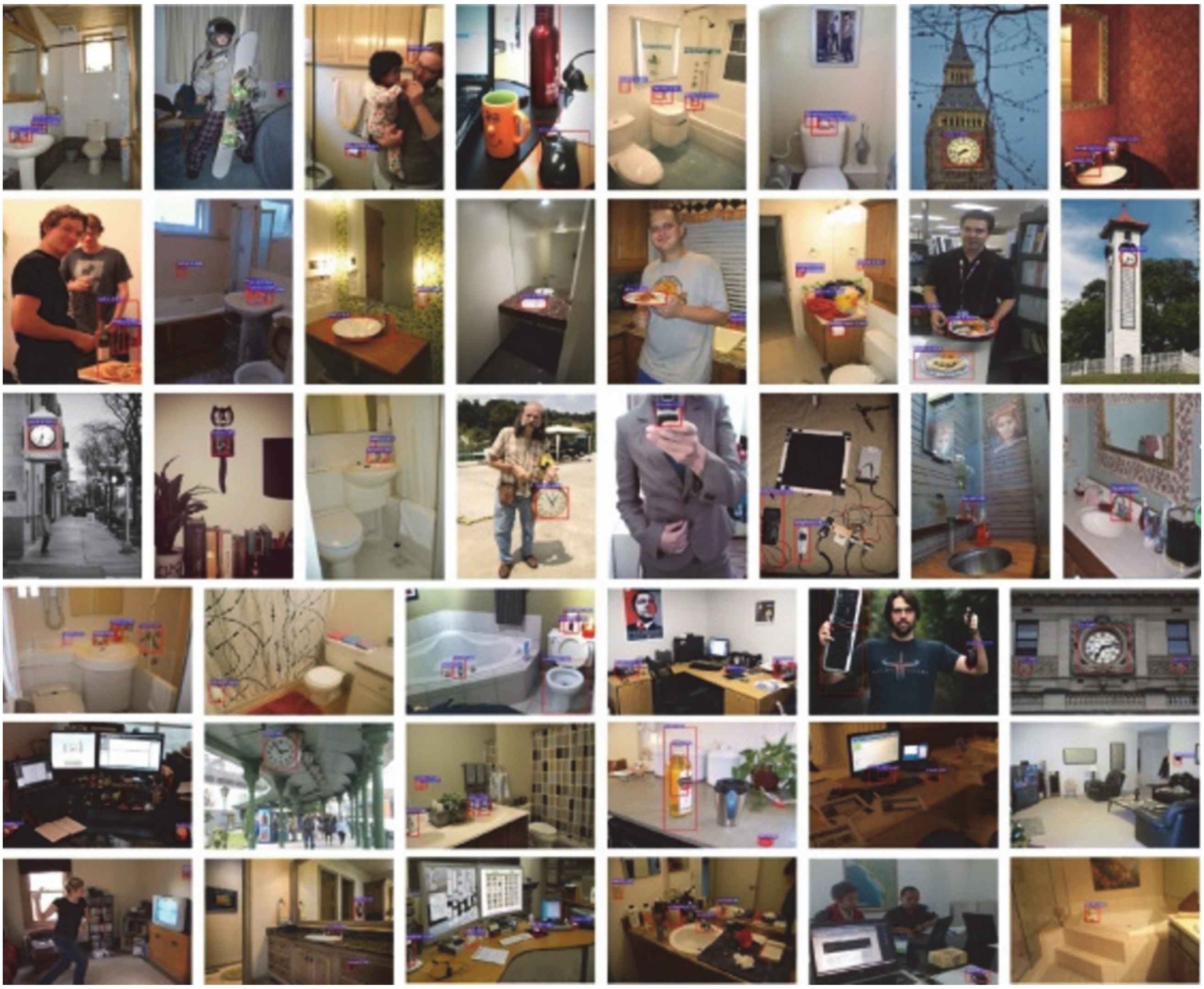


Figure 5: The detection renderings.

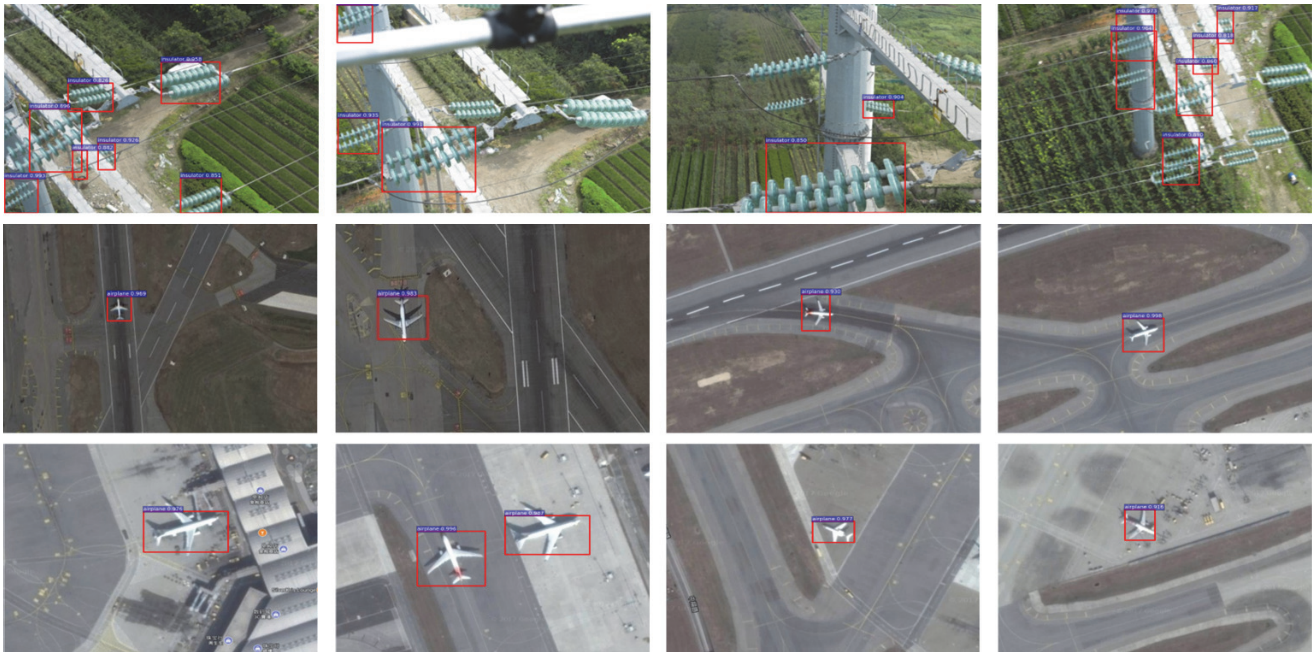


Figure 6: Effect of remote sensing image detection.

only ensures the integrity of the feature of the large object but also preserves the full detail feature of the small objects by extracting the multiscale feature of the image. So it can improve the accuracy of the detection of the small objects.

The GANs (Generative Adversarial Nets) have been widely applied to the game area and achieved good results [29]. For future work we believe that investigating more sophisticated techniques for improving the accuracy of small object detection, including the Generative Adversarial Nets, will be beneficial. Existing object detection usually detects small objects through learning representations of all the objects at multiple scales. However, the performance is usually limited to pay off the computational cost and the representation of the image. In the future, we address the small object detection problem that internally lifts representations of small objects to "super-resolved" ones, achieving similar characteristics as large objects and thus being more discriminative for detection. And finally, we use the adversarial network to train the detection model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

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 under Grants nos. 61662033 and 61473144, Aeronautical Science Foundation of China (Key Laboratory) under Grant no. 20162852031, and the Special Scientific Instrument Development of Ministry of Science and Technology of China under Grant no. 2016YFF0103702.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I511–I518, Kauai, Hawaii, USA, December 2001.

[2] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. I/900–I/903, Rochester, NY, USA, September 2002.

[3] P. Viola, J. C. Platt, and C. Zhang, "Multiple Instance boosting for object detection," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS '05)*, vol. 18, pp. 1417–1424, Vancouver, British Columbia, Canada, December 2005.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, Santiago, Chile, December 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[8] J. F. Dai, Y. Li, K. M. He et al., "R-FCN: Object Detection via Region-based Fully," in *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

[9] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[10] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster R-CNN," *Applied Sciences*, vol. 8, no. 5, article 813, 2018.

[11] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[12] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[13] C. Chen, M. Y. Liu, O. Tuzel et al., "R-CNN for small object detection," in *Asian Conference on Computer Vision*, vol. 10115 of *Lecture Notes in Computer Science*, pp. 214–230, 2016.

[14] J. S. Lim and W. H. Kim, "Detection of multiple humans using motion information and adaboost algorithm based on Harr-like features," *International Journal of Hybrid Information Technology*, vol. 5, no. 2, pp. 243–248, 2012.

[15] R. P.Yadav, V. Senthamilarasu, K. Kutty, and S. P. Ugale, "Implementation of Robust HOG-SVM based Pedestrian Classification," *International Journal of Computer Applications*, vol. 114, no. 19, pp. 10–16, 2015.

[16] L. Hou, W. Wan, K.-H. Lee, J.-N. Hwang, G. Okopal, and J. Pitton, "Robust Human Tracking Based on DPM Constrained Multiple-Kernel from a Moving Camera," *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 27–39, 2017.

[17] A. Ali and M. A. Bayoumi, "Towards real-time DPM object detector for driver assistance," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 3842–3846, Phoenix, Ariz, USA, September 2016.

[18] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2874–2883, Las Vegas, Nev, USA, July 2016.

[19] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 845–853, Las Vegas, Nev, USA, June 2016.

[20] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2129–2137, Las Vegas, Nev, USA, July 2016.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 779–788, Las Vegas, Nev, USA, July 2016.

[22] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *European Conference on Computer Vision*, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, Springer, Cham, Switzerland, 2016.

[23] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[24] C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," in *European Conference on Computer Vision*, pp. 391–405, Springer, Cham, Switzerland, 2014.

[25] M. Najibi, M. Rastegari, and L. S. Davis, "G-CNN: An iterative grid based object detector," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2369–2377, Las Vegas, Nev, USA, July 2016.

[26] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, vol. 8693 of *Lecture Notes in Computer Science*, pp. 740–755, Springer, Cham, Switzerland, 2014.

[27] http://groups.csail.mit.edu/vision/SUN/.

[28] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-RCNN approach to driver's cell-phone usage and hands on steering wheel detection," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2016*, pp. 46–53, Las Vegas, Nev, USA, July 2016.

[29] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 660–674, 2017.