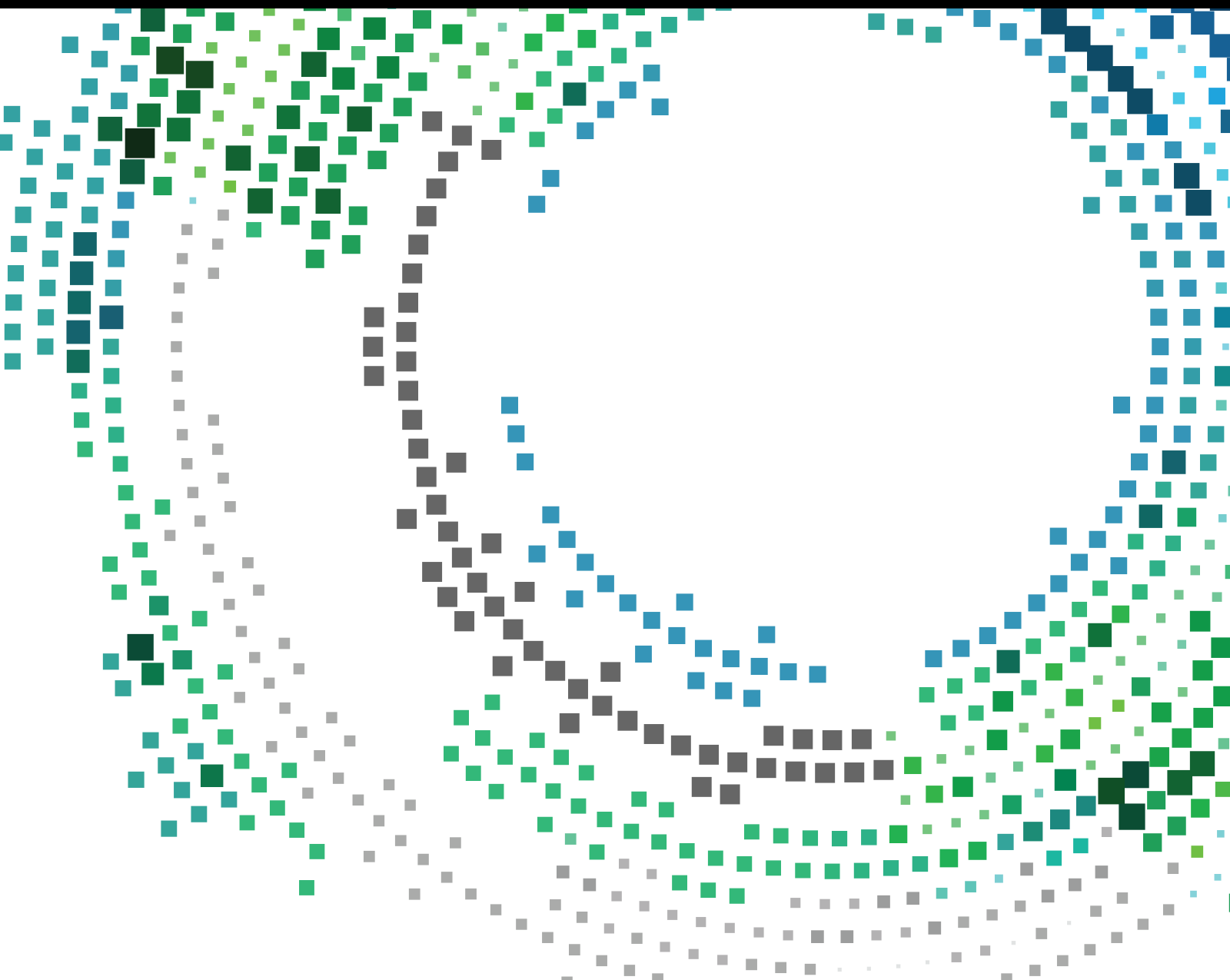


AI and Edge Computing-Driven Technologies for Knowledge Defined Networking

Lead Guest Editor: Jianhui Lv

Guest Editors: Yuhui Shi, Xingwei Wang, Hui Cheng, and Lianbo Ma





AI and Edge Computing-Driven Technologies for Knowledge Defined Networking

Mobile Information Systems

AI and Edge Computing-Driven Technologies for Knowledge Defined Networking

Lead Guest Editor: Jianhui Lv

Guest Editors: Yuhui Shi, Xingwei Wang, Hui
Cheng, and Lianbo Ma



Copyright © 2023 Hindawi Limited. All rights reserved.




This is a special issue published in “Mobile Information Systems.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Alessandro Bazzi , Italy

Academic Editors

Mahdi Abbasi , Iran
Abdullah Alamoodi , Malaysia
Markos Anastassopoulos, United Kingdom
Marco Anisetti , Italy
Claudio Agostino Ardagna , Italy
Ashish Bagwari , India
Dr. Robin Singh Bhadoria , India
Nicola Bicocchi , Italy
Peter Brida , Slovakia
Puttamadappa C. , India
Carlos Calafate , Spain
Pengyun Chen, China
Yuh-Shyan Chen , Taiwan
Wenchi Cheng, China
Gabriele Civitarese , Italy
Massimo Condoluci , Sweden
Rajesh Kumar Dhanaraj, India
Rajesh Kumar Dhanaraj , India
Almudena Díaz Zayas , Spain
Filippo Gandino , Italy
Jorge Garcia Duque , Spain
Francesco Gringoli , Italy
Wei Jia, China
Adrian Kliks , Poland
Adarsh Kumar , India
Dongming Li, China
Juraj Machaj , Slovakia
Mirco Marchetti , Italy
Elio Masciari , Italy
Zahid Mehmood , Pakistan
Eduardo Mena , Spain
Massimo Merro , Italy
Aniello Minutolo , Italy
Jose F. Monserrat , Spain
Raul Montoliu , Spain
Mario Muñoz-Organero , Spain
Francesco Palmieri , Italy
Marco Picone , Italy
Alessandro Sebastian Podda , Italy
Maheswar Rajagopal, India
Amon Rapp , Italy
Filippo Sciarrone, Italy
Floriano Scioscia , Italy

Mohammed Shuaib , Malaysia
Michael Vassilakopoulos , Greece
Ding Xu , China
Laurence T. Yang , Canada
Kuo-Hui Yeh , Taiwan

Contents

Retracted: Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities

Mobile Information Systems


Retraction (1 page), Article ID 9873534, Volume 2023 (2023)

Retracted: Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing

Mobile Information Systems

Retraction (1 page), Article ID 9850629, Volume 2023 (2023)

Research on Influence of Attribute Frame Effect on Loan Decision of Undergraduate and Risk Assessment Model of Undergraduate Loan Behavior

Jinsong Luan 


Research Article (8 pages), Article ID 9910442, Volume 2021 (2021)

Artificial Intelligence and Neural Network-Based Shooting Accuracy Prediction Analysis in Basketball

Hongfei Li  and Maolin Zhang 

Research Article (11 pages), Article ID 4485589, Volume 2021 (2021)

Analysis of Physical Expansion Training Based on Edge Computing and Artificial Intelligence

Zhongle Liu 

Research Article (9 pages), Article ID 9145952, Volume 2021 (2021)

[Retracted] Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing

Fang Xu  and Wentao Chu


Research Article (8 pages), Article ID 3534577, Volume 2021 (2021)

Positioning of Apple's Growth Cycle Based on Pattern Recognition

Wenfeng Li, Yulin Yuan, Shikang Hu, Mei Li, Wenxiu Feng, and Jiaxin Zheng 

Research Article (11 pages), Article ID 9687950, Volume 2021 (2021)

Efficient English Translation Method and Analysis Based on the Hybrid Neural Network

Chuncheng Wang 

Research Article (10 pages), Article ID 9985251, Volume 2021 (2021)

Artificial Intelligence-Based Joint Movement Estimation Method for Football Players in Sports Training

Bin Zhang, Ming Lyu , Lei Zhang, and Yang Wu


Research Article (9 pages), Article ID 9956482, Volume 2021 (2021)

Data Collection and Analysis of Track and Field Athletes' Behavior Based on Edge Computing and Reinforcement Learning

Di Han 

Research Article (11 pages), Article ID 9981767, Volume 2021 (2021)

[Retracted] Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities

Ni Hong , Xuefeng Wang, and Zhonghua Wang

Research Article (9 pages), Article ID 9977736, Volume 2021 (2021)

High-Performance Server-Based Live Streaming Transmission Optimization for Sports Events in Smart Cities

Si Chen and Gang Zhao 

Research Article (7 pages), Article ID 9958703, Volume 2021 (2021)

Intelligent Forwarding Strategy for Congestion Control Using Q-Learning and LSTM in Named Data Networking

Sanguk Ryu , Inwhee Joe , and WonTae Kim 


Research Article (10 pages), Article ID 5595260, Volume 2021 (2021)

Evaluation and Analysis of Traditional Physical Training by Using Mobile Edge Computing and Software-Defined Networking

Wenwen Pan , Jianzhi Wang , and Jingsheng Ji


Research Article (9 pages), Article ID 5596291, Volume 2021 (2021)

Education Data-Driven Online Course Optimization Mechanism for College Student

Ziqiao Wang and Ningning Yu 


Research Article (8 pages), Article ID 5545621, Volume 2021 (2021)

Pattern Recognition and Neural Network-Driven Roller Track Analysis via 5G Network

Yuliang Guo 


Research Article (8 pages), Article ID 6675140, Volume 2020 (2020)

In-Network Caching and Edge Computing-Based Live Broadcasting Optimization for Football Competitions

Zhigang Li 

Research Article (8 pages), Article ID 6698448, Volume 2020 (2020)

DRL-Based Edge Computing Model to Offload the FIFA World Cup Traffic

Hongyi Li and Xinrui Che 

Research Article (11 pages), Article ID 8825643, Volume 2020 (2020)

Performance Optimization Mechanism of Adolescent Physical Training Based on Reinforcement Learning and Markov Model

Mingze Wei  and Lei Yuan 

Research Article (10 pages), Article ID 8868225, Volume 2020 (2020)

Human Falling Detection Algorithm Based on Multisensor Data Fusion with SVM

Daohua Pan , Hongwei Liu , Dongming Qu, and Zhan Zhang

Research Article (9 pages), Article ID 8826088, Volume 2020 (2020)

Retraction

Retracted: Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities

Mobile Information Systems

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

Copyright © 2023 Mobile Information Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] N. Hong, X. Wang, and Z. Wang, "Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities," *Mobile Information Systems*, vol. 2021, Article ID 9977736, 9 pages, 2021.

Retraction

Retracted: Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing

Mobile Information Systems

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

Copyright © 2023 Mobile Information Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] F. Xu and W. Chu, "Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing," *Mobile Information Systems*, vol. 2021, Article ID 3534577, 8 pages, 2021.

Research Article

Research on Influence of Attribute Frame Effect on Loan Decision of Undergraduate and Risk Assessment Model of Undergraduate Loan Behavior

Jinsong Luan 

Shenyang Sport University, Shenyang 110102, China

Correspondence should be addressed to Jinsong Luan; luanjs@syty.edu.cn

Received 24 March 2021; Revised 23 April 2021; Accepted 24 May 2021; Published 4 June 2021

Academic Editor: Jianhui Lv

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

The article uses virtual lending scenarios to study the influence of attribute frame effect on undergraduates' loan decisions. The results show that undergraduates have attribute frame effects in the three major areas of electronic products, life entertainment, and learning and training. There is a significant difference between the positive frame and the negative frame; that is, they are more inclined to make loan decisions under the positive frame. According to the research results, the article designs a loan risk assessment model based on Kohonen neural network and conducts simulation experiments. The experimental results show that the model' classification accuracy is 72.65%.

1. Introduction

Nowadays, undergraduates' consumption concept such as comparison consumption and overspending has become prevalent. In 2009, the CBRC issued the *Notice on Further Regulation of Credit Card Business*, which demanded banks and financial institutions not to issue credit cards to students under 18-year-old. Besides, while issuing credit cards to students who have reached 18-year-old but do not have a stable income source, they need to achieve a second source of repayment with the ability to repay, after which major banks have tightened the credit card business to undergraduates. In contrast to the high threshold and severe audit of banks and financial institutions, peer-to-peer (P2P) lending platforms emerged in July 2013 and were favored by undergraduates for their convenience of application, simple procedures, rapid lending, and no mortgage. In order to improve the market share, the platforms have developed their business to universities and quickly developed P2P lending for undergraduates. Besides, to attract undergraduates to apply P2P lending, the platform usually attracts customers through inducing and opaque slogans. This makes it easy for uninitiated undergraduates to lose themselves in borrowing

and consumption, making borrowing decisions that are not in line with their spending power. A range of negative events have occurred frequently since then, and even vicious events have happened. The chaos of campus P2P lending has alleviated since proposing the corresponding regulations and policies in 2016; thus, the use of campus P2P lending has been controlled. Simultaneously, the P2P lending platforms are not allowed to promote on campus, and universities have started to educate students to stay away from warning cases. By contrast, undergraduates who take P2P lending and installment consumption still occupy a particular proportion, and news of crisis made by the inability to repay the loan on time is common. Excessive repayment pressure can easily point to psychological crisis events. Additionally, regular overconsumption through P2P lending will undoubtedly change students' consumption habits, affect credit, and carry a heavy burden to families, which is also a negative factor influencing social harmony and stability from the social prospect. Therefore, research on the psychology of undergraduates' P2P lending consumption and introducing effective educational measures can overcome a range of negative effects of campus P2P lending. In the meantime, it can also lead undergraduates to construct

rational consumption concepts and financial concepts and prevent crisis events.

The article uses virtual lending scenarios to study the influence of attribute frame effect on undergraduates' loan decisions and designs a loan risk assessment model to solve this situation.

2. Related Work

P2P (peer-to-peer) lending means the verification of relevant information, money issuance, contracts signing, and other related procedures and process, all realized through the Internet. P2P lending implies individual-to-individual lending, a financial model in which individuals present small loans to other individuals via a third party on the premise of charging interest. On the other hand, campus P2P lending determines the online financial service that integrates P2P lending and network lending for college students.

Most borrowers of campus P2P lending platforms only require to implement a relevant proof of identity to apply for loans or installment consumption without collateral or guarantee. The platform does not rigidly examine the application information, and some platforms do not have the responsibility of due diligence; hence, undergraduates can use other students' identity information to reach loans quickly. The lax audit and inadequate supervision make it challenging for some students to get rid of it and make the platform full of risks.

In 2016, the *Special Research Report on China's Campus Consumer Financial Market* issued by Yiguan Think Tank surveyed 80 mainstream P2P lending platforms in the market [1]. The outcomes revealed that the average monthly interest rate was 0.92%, which translated into monthly interest repayment and capital repayment, and the actual annualized interest rate was regularly over 20%, some even up to 35%. Once the repayment is overdue, 48% of the platforms have unknown overdue rates, and 58% of the platforms that clearly announce overdue rates have a daily rate of 1% after the overdue date. Subsequently, P2P lending platforms also have so-called service fees, default fees, and other charges, but they are never explicitly stated in the advertisements, attracting undergraduates to borrow with opaque and inducing advertising slogans.

The use of net loans for advanced consumption has become a common phenomenon among college students. In 2015, Tencent Technology published a survey entitled *Report on the Real Demand for undergraduates' Installment of P2P Lending*, which showed that approximately 60% of undergraduates used installment loans to buy electronic products and 39.8% of them were expected to borrow more than 2,000 yuan, but 65.1% of them lived on only 500 yuan to 1,500 yuan per month at that time [2]. In 2016, in the *Special Research Report on China's Campus Consumer Financial Market* published by EnfoDesk, undergraduates utilized installment consumption relatively frequently, with 36% of active users and 66% stickiness to the installment platform, with 70% of males in installment consumption and 86% consumption took for purchasing electronic products [3]. In contrast to the extensive application of

installment consumption for overspending, the *2017 Survey Report on the Credit Status of Chinese undergraduates* released by CSCU revealed that only 65% of undergraduates who took campus loans would calculate the total cost of the loan [4]. However, 35% of students did not even comprehend the actual cost of interest but only obey what the campus loan platform tells them. If they could not repay the loan on time, 42% of students did not ask their families for assistance, and 65% of them did not know their credit reports. Based on the *Consumption Insight Report of College Students in 2018* released by iResearch, in 2017, the annual consumption scale of 30.18 million undergraduates was as high as 381.568 billion yuan [5]. The online e-commerce platform was the main channel for college students' consumption, and Ant Credit Pay and JD Baitiao were the preferred installment consumption techniques for undergraduates.

Jing surveyed 792 undergraduates in Yunnan, and the consequences explained that 33.2% of students had experienced P2P lending [6]. Fan managed a survey on students in six universities in Yunnan, and the outcomes revealed that 24.63% of students had adopted P2P lending, of which 55.71% were male students, 77.58% were senior students (juniors and seniors), and 85.39% of students had P2P lending of less than 5,000 yuan [7]. Fu surveyed undergraduates, and the results explicated that 27% of students had applied P2P lending, 24.7% of them regarded P2P lending to be safe, and 35.9% of them cited parents and online reviews as the main influencing factors for P2P lending determination [8]. Cong et al. surveyed 672 students in one of the universities in Shanghai, and the outcomes proved that 56.1% of students had taken P2P lending online loan platforms, with a higher proportion of male students utilizing P2P lending, accounting for 61.02%, and 53.44% of female students [9].

In 2015, CMRC and Beijing Yixin Zhicheng Credit Management Co., Ltd. jointly published the *Kindling Project—National Research Report on undergraduates' Credit Awareness*—which revealed that most undergraduates who lacked a reasonable budget for money have a weak awareness of P2P lending and were significantly lacking in credit knowledge. Meanwhile, more than 80% of undergraduates did not comprehend personal credit reports, and 30% did not understand that overdue records would be recorded in personal credit reports and influenced their future financial lives [3]. Furthermore, only 12% of them knew that the overdue records in credit reports were kept for several years, and 40% of them were late in repayment, and 10% of them repaid through borrowing.

The trend of consumerism has enhanced society's mainstream and has exceedingly affected contemporary undergraduates' consumption concept. Contemporaneously, the spurt development of major e-commerce platforms and the overwhelming advertisements of large and small consumer festivals continually influence undergraduates' rational nerves. Consumerism is constantly alienating people's economic behavior, and bad trends simply affect undergraduates whose world and life perspective values are

not yet completely built up to make irrational behavioral decisions.

According to Erikson's theory of psychological development step, undergraduates are in self-identity integration. At this step, the group of undergraduates has the compelling requirement for identity, and the internal standards will turn into the guidelines that undergraduates have to obey; otherwise, they will feel tremendous internal and external pressure. The same is true for the consumption process. In consumption, undergraduates will consume and pick goods through imitation, conformity, and assimilation to obtain their peers' praise or envy. This variety of consumer behavior is also one of the triggers for blind, irrational consumption.

Currently, the laws, regulations, and industry self-regulation norms for P2P lending are not yet perfect. In order to reach the enormous profit of undergraduates, primary campus P2P lending platforms utilize advertising, QQ groups, WeChat official accounts, Weibo, and other communication media to incite the excessive consumption of undergraduates to expand market share. They induce undergraduates to take out P2P lending with advertisements such as "without a down payment, zero interest rate, no collateral, receive the payment in ten minutes" and deliberately evade the description of borrowing risks. If undergraduates are late in repaying the loan, the P2P lending platform employs malicious disclosure, harassment, threats to violently draw the loan, or illegally applies the student's personal information for personal gain.

3. Research on the Influence of Attribute Frame Effect on the Loan Decision of Undergraduate

3.1. Attribute Framing Effects. In 1981, Kahneman and Tversky discovered a framing effect in the decision-making process, i.e., changing the expression form of essentially the same decision plan could inspire people to make diverse decisions [10]. Levin et al. defined the framing effect into risky choice framing effect, attribute framing effect, and goal framing effect based on three dimensions: the object being framed, the object influenced by the framing, and the standard measurement of the effect, and argued that the three types of framing effects were independent of each other and symbolized separate processing processes [11].

The attribute framing effect refers to the fact that a consequence or event can be rated separately in attractiveness based on some of its attributes $d_1, d_2, d_3, \dots, d_n$, with positive framing showing the attribute as A+ (e.g., success, getting slender) and negative framing revealing it as A- (e.g., failure, getting fatter). The attribute framing effect indicates that subjects are more inclined to make higher attractiveness ratings for consequences or events in the A+ context relative to consequences or events in the A- context.

The attribute framing effect has two characteristics as follows. (1) The risk-free attribute framing effect involved only the rating of things and did not include the choice of risky conditions, whereas the choice of risky conditions was present in the risk framing effect and the goal framing effect (see Levin et al. [11]). (2) The specificity attribute framing effect only impacted the rating process of commodity rating,

performance appraisal, and gambling (see Wen et al. [12]), whereas the risk and goal framing were pervasive (see Levin et al. [11]). For example, if a student requires to borrow 6,000 yuan to get a cell phone, one lending platform defines "borrowing for 12 months with a monthly repayment of 580 yuan." Another platform describes "borrowing for 12 months interest-free, but you have to pay a monthly fee of \$80." In the same repayment amount, only the variation in the described approach can surely influence the undergraduates' P2P lending decision. Various P2P lending platforms utilize this kind of inducement or ambiguous slogan to attract undergraduates to choose them.

3.2. Research Subjects and Research Methods. The research subjects were students from the first to the fourth year in one of the universities in Shenyang, 600 questionnaires were distributed according to random sampling, and 581 valid questionnaires were obtained. Among them, 342 (58.9%) were male students, 239 (41.1%) were female students, 198 (34.1%) were first-year students, 166 (28.6%) were second-year students, 123 (21.2%) were third-year students, and 94 (16.2%) were fourth-year students. The questionnaire was self-administered, and the questions were designed to know undergraduates' P2P lending consumption and the impact of the attribute framing effect on P2P lending decisions.

3.3. Undergraduates' Consumption of P2P Lending. This paper intuitively knew the usage and distribution features of undergraduates' monthly living expenses through the questionnaire survey. The total monthly living expenses of RMB 1000 and below valued for 9.1%, RMB 1000 to 1500 occupied for 34.9%, RMB 1500 to 2000 accounted for 32.5%, RMB 2000 to 2500 valued for 16.5%, and RMB 2500 and above occupied for 6.9%, with average monthly living expenses of RMB 1656.9. In general, there was a certain gap in the level of expenses among individual undergraduates.

The distribution of living expenses was principally focused on six areas: food and drink, clothing, shoes, and hats, make-up and daily necessities, electronic products (including game equipment), living and entertainment, and learning and training. The proportion of food and drink valued for the most significant proportion of monthly expenses accounted for 35% to 55% of monthly expenses, with only 6.8% of students accounted for less than 35% (Table 1). It was noteworthy that gatherings, a common form of daily interaction among undergraduates, also occupied a specific proportion of monthly expenses. There was a meaningful variation between male and female students in three categories of living expenses: clothing, shoes, and hats, cosmetics, and electronic products (including gaming devices). Among the females, 60.1% spent more than 20% on clothing and shoes and 51.2% spent more than 20% on daily necessities. The proportion of males who spent more than 20% in these two categories was lower, 41.2% and 19.7%, individually. In electronics (including gaming devices), 32% of the males spent more than 15% of their monthly electronic products expenses. In terms of living and entertainment expenses, 64% of females spent more than 20% of their monthly expenses

TABLE 1: Undergraduates' perception of P2P lending consumption.

| Serial no. | Issue | Option percentage | | | |
|------------|--|----------------------------|----------------------------|---------------------------------|--------------------|
| 1 | Knowledge of P2P lending process and business | Do not understand 9.0% | Do not know much 60.1% | Better understand 25.5% | Know well 5.5% |
| 2 | Understand the channels of P2P lending | Network platform 35.9% | Media advertising 28.2% | Classmates and friends 31.2% | Relatives 4.8% |
| 3 | Will the comprehensive cost of interest be measured before applying the P2P lending? | Yes 53% | | No 47% | |
| 3 | Whether P2P lending is safe or not | Yes 35.5% | | No 64.5% | |
| 4 | Attitude towards P2P lending | Convenient 47.8% | Low threshold 54% | High interest 72.1% | High risk 67.1% |
| 5 | Will you ask your family for help when you cannot make repayments on time? | Yes 21.9% | | No 78.1% | |
| 6 | How much do you know about personal credit reports? | Do not understand 32.5% | Do not know much 42.2% | Better understand 17.6% | Know well 7.7% |

and 61% of males spent more than 20% of their monthly expenses, so it could be observed that living and entertainment expenses were more extensive in the undergraduates. The proportion of learning and training consumption was low, with 89% of them spent less than 10% of their monthly expenses on learning and training, and the difference between male and female students was not noticeable.

The survey revealed that undergraduates' awareness of online loans was inadequate, and the phenomenon of insufficient financial knowledge was still common. The outcomes showed that 70% of undergraduates did not know much or even comprehend the process of P2P lending, and only half of them would determine the comprehensive cost of interest before applying P2P lending. Approximately 80% of them would not ask for help from their families if they could not repay the loan on time, and less than 30% of them knew about their credit reports (Table 2). Online platforms and friends were the main channels for them to learn about P2P lending.

The investigation revealed that only 30% of undergraduates would decide to give up consumption if their living expenses cannot satisfy their expenses, and the awareness of rational consumption was inadequate. Among them, females were comparatively more rational, and 68% of the people who gave up consumption were females. 12.2% of undergraduates have utilized P2P lending, and the main reasons for taking P2P lending were low threshold and fast lending. Classmates and friends and interest rates were the main reasons for taking P2P lending; 70% of them took P2P lending with RMB 2,000 to 4,000 yuan for consumption and shopping (Table 2).

3.4. The Influence of Attribute Framing Effect on the Loan Decision of Undergraduate. The attribute framing effect only influences the evaluation process of commodity evaluation, performance assessment, and gambling. As a result, relying on the attribute framing effect, this article also composed a virtual borrowing scenario to analyze further the effect on undergraduates' lending decisions.

The virtual P2P lending scenario materials are extracted from a specific characteristic of the commodity, and the positive frame designates this characteristic positively

(Table 3). In contrast, the negative frame describes it negatively to investigate whether the lending description affects the reception decision. According to undergraduates' life scenarios, this research picked electronic products, living and entertainment, and learning and training among the six major life expenses for the material design. The three scenarios of purchasing cell phones, travel expenses, and insufficient training expenses were adopted to represent positively and negatively, and the responses were scored on a five-point Likert scale.

Undergraduates had attribute framing effects in the three major areas of electronic products, living and entertainment, and learning and training, and the distinction between positive and negative frames was meaningful (Table 4). That is, they were more inclined to make borrowing decisions under the positive frame. Accordingly, the description of borrowing matters by lending platforms could widely impact undergraduates' borrowing behavior. The demographic variable analysis also revealed no critical difference in undergraduates' borrowing decisions with various living expenses and diverse genders in the three major domains under the attribute framework (Tables 5 and 6).

3.5. Discussion. The survey explained that subsistence consumption and enjoyment consumption were the principal consumption patterns of undergraduates, particularly enjoyment consumption accounts for a higher proportion. On the other hand, the proportion of developmental consumption was comparatively low, the proportion of over-consumption was high, and the phenomenon of inadequate knowledge of P2P lending and insufficient financial knowledge was also widespread. Meanwhile, the attribute framing effect also had an essential influence on the borrowing decision process, and the induced slogan of P2P lending platforms might influence undergraduates to determine to borrow blindly. Therefore, it was crucial to promoting financial literacy and mastered financial knowledge for undergraduates. University should develop education channels, supervise undergraduates to launch rational consumption concepts, master financial credit knowledge, and enhance financial management ability and

TABLE 2: The consumption status of undergraduates' P2P lending.

| Serial no. | Issue | Option percentage | | | |
|------------|---|-------------------------|---|-----------------------------------|-----------------------|
| 1 | While living expenses cannot satisfy expenses | Give up spending 34% | Borrow from friends and relatives 10.1% | Use consumption installment 41.5% | Use P2P lending 14.3% |
| 2 | Have you used P2P lending? | Yes 12.2% | | No 87.8% | |
| 3 | Reasons for selecting P2P lending | Convenient audit 17.6% | Low threshold 35.5% | Quick release of funds 38.9% | Flexible repayment 8% |
| 4 | Reasons that affect the choice | Online review 16.9% | Classmates and friends 53% | | Interest 30.1% |
| 5 | Choose the amount of P2P lending | Less than 2000 18.2% | 2000 to 3000 39.4% | 3000 to 4000 32% | More than 4000 10.3% |
| 6 | Use of P2P lending funds | Temporary emergency 17% | Consumption and shopping 70.2% | Learning and training 5.2% | Entrepreneurship 7.6% |

TABLE 3: Materials for virtual scenarios.

| Living expenses | Description frame | Scenario material |
|--------------------------|-------------------|---|
| Electronic product | Positive frame | Suppose you intend to buy a new mobile phone, but the price of 6000 yuan is beyond your budget, and now a P2P lending platform proposes installment purchase business with fees. If you choose 12 installments, you need to repay 500 yuan per month. How much are you willing to pick the P2P lending at this time? |
| | Negative frame | Suppose you intend to buy a new mobile phone, but the price of 6000 yuan is beyond your budget, and now a P2P lending platform proposes installment purchase business with fees. If you choose 12 installments, you need to pay an extra fee of 80 yuan per month. How much are you willing to pick the P2P lending at this time? |
| Living and entertainment | Positive frame | Suppose you plan to travel with your classmates during the short holiday, but the cost of 3,000 yuan surpasses your living budget. Currently, a P2P lending platform drives a one-month interest-free loan service. If the lending is fast within one month, no interest is required. To repay, you demand to repay 0.2% of the loan amount every day, that is, to repay an extra 6 yuan in interest every day. How much are you willing to pick the P2P lending at this time? |
| | Negative frame | Suppose you plan to travel with your classmates during the short holiday, but the cost of 3,000 yuan surpasses your living budget. Currently, a P2P lending platform drives a one-month interest-free loan service. If the lending is fast within one month, no interest is required. To repay, you need to repay 6% of the loan amount every month, that is, to repay an additional 180 yuan in interest per month. How much are you willing to pick the P2P lending at this time? |
| Learning and training | Positive frame | Suppose you plan to engage in an overseas study tour during the holiday, but the cost of 6000 yuan surpasses your living budget. Currently, a P2P lending platform establishes a six-month low-interest loan business, charging a service fee of 1.7% of the loan amount per month, which is 100 yuan per month, if you pay back on time for the month. How much are you willing to pick the P2P lending at this time? |
| | Negative frame | Suppose you plan to engage in an overseas study tour during the holiday, but the cost of 6000 yuan surpasses your living budget. Currently, a P2P lending platform establishes a six-month low-interest loan business, charging a service fee of 10% of the loan amount, which is 600 yuan per month, if you pay back on time for the month. How much are you willing to pick the P2P lending at this time? |

TABLE 4: Under the attribute framework, the differences in the loan decisions of undergraduates in the three scenarios.

| Living expenses | Description frame | <i>t</i> |
|--------------------------|-------------------|----------|
| Electronic product | Positive frame | 1.874** |
| | Negative frame | |
| Living and entertainment | Positive frame | 3.016** |
| | Negative frame | |
| Learning and training | Positive frame | 4.285* |
| | Negative frame | |

Note. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

TABLE 5: Comparison of decision-making differences between various levels of living expenses under the attribute framework.

| Serial no. | Areas of living expenses | <i>F</i> |
|------------|--------------------------|----------|
| 1 | Electronic product | 0.26 |
| 2 | Living and entertainment | 0.315 |
| 3 | Learning and training | 0.248 |

Note. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

TABLE 6: Comparison of decision-making differences between different genders under the attribute framework.

| Serial no. | Areas of living expenses | T |
|------------|--------------------------|------|
| 1 | Areas of living expenses | 1.7 |
| 2 | Electronic product | 0.98 |
| 3 | Living and entertainment | 1.45 |

Note. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

self-planning. Moreover, to advance financial management ability and self-planning level, undergraduates were assisted to keep rational consumption in the contemporary society where consumerism was prevalent, the spirit of struggle was stimulated, and self-development was made the main theme of university life.

4. The Risk Assessment Model of Undergraduate Loan Behavior

4.1. Kohonen Neural Network-Based Risk Assessment Model. Artificial neural network (ANN) is based on the basic principles of neural networks in biology. After comprehending and abstracting the structure of the human brain and the response mechanism of external stimuli, it is a mathematical model that affects the processing mechanism of the human brain's nervous system for complex information under the theoretical knowledge of network topology. This model is characterized by parallel distributed processing capability, high fault tolerance, intelligence, and self study, and this model combines information processing and storage. Meantime, its unique knowledge representation and intelligent adaptive learning capability have attracted attention in multiple disciplines [13].

Kohonen network is a self-organizing feature mapping network introduced by Professor Kohonen in Finland. In the biological nervous system, there is a phenomenon of lateral inhibition; that is, after a nerve cell is stimulated, it will have an inhibitory impact on the surrounding cells. This inhibition will provoke competition between nerve cells. The competition's outcome is that the winner is excited, and the loser is inhibited, and the Kohonen network copies this biological phenomenon. If a neural network accepts external input patterns, it will be split into numerous corresponding areas. Every area has diverse response features to the input mode, and this process is done automatically. The self-organizing feature map is based on this prospect, and its features are similar to the self-organizing feature of the human brain. Kohonen neural network is an unsupervised learning neural network with self-organization function. Through its own training, the network can automatically classify the input mode.

In terms of network structure, it is regularly a two-layer network consisting of an input layer and a competing layer. The neurons between the two layers fulfill bidirectional connections, and the network has no hidden layer. Sometimes there are also lateral connections between the neurons in the competitive layer. In the learning algorithm, it mimics the kinetic principles of excitation, coordination, and inhibition, and competitive action of information processing

between biological neurons to manage the learning and working of the network, unlike most neural networks that utilize the mistake or energy function of the network as the criterion of the algorithm.

The basic idea of the competitive neural network is that each neuron in the network competes to reply to the input pattern. Eventually, only one neuron converts the winner of the competition. This winning neuron designates the classification of the input pattern. In the competition layer, only one winning neuron will be generated for each sample input [14]. Considering that the input layer has m neurons and the core layer has n neurons, the weight should be an $m \times n$ matrix. If the input vector is $p = [p_1, p_2, \dots, p_n]$, the output of the network is $Y = P \omega$. In reply to the n neurons in the competition layer, one neuron must become the winning neuron Y_k . The weight correction method of the winning neuron is $\Delta \omega_{ik} = \eta (P_i - \omega_{ik}) Y_k$. It can be observed that the weight is close to the sample P_i at a rate of η because the weight of the winning neuron Y_k is near to the input sample P_i . Consequently, Y_k has a more prominent probability of winning in the next iteration. If the step size of the rate η is suitable, the input sample's network weight corresponding to the winning neuron will approach the input sample, and the gap will become smaller and tinier [15].

In order to accommodate the characteristics of online lending and borrowing decisions, the Kohonen network is enhanced to make it clustered. An additional output layer is added after the competitive layer to turn it into a supervised learning network [16]. The number of neurons in the added output layer is identical to the number of suitable categories in the database. The neurons' connection weights in the output layer and the neurons in the competitive layer are ω_{jk} . If the weights are modified, the weights of the input layer and those of the competing layer and the output layer are adjusted concurrently in two categories, and the weights are adjusted as $\omega_{jk}(n+1) = \omega_{jk}(n) + \eta_2 (Y_k - \omega_{jk}(n))$. Among them, ω_{jk} is the weight between the core and output layers, η_2 is the learning rate, and Y_k is the actual class to which the sample belongs.

The borrowing risk of undergraduates principally includes both personal factors and platform factors. In accordance with this study's outcomes, age and gender influence borrowing decision, and consumption and personal credit status can also effectively evaluate undergraduates' daily consumption level and credit history. The platform factors comprise the information attribute framework of P2P lending in this research and the interest rate, late interest rate, and default loss of lending. If the P2P platform presents information through an active framework, the borrowing interest rate and overdue interest rate are not high, and the default loss is small, and undergraduates will make borrowing decisions. Hence, undergraduates' borrowing risk model indicators involve age, gender, consumption, personal credit status, attribute framework of P2P lending information, interest rate, overdue interest rate, and default loss. Since the neural network cannot identify the textual content, the chosen model indicators are numerically transformed before simulation training. Every indicator's textual information

is quantified into a distinct value between $[0, 1]$ according to the probability size of default risk.

The simulation experiment of the model selects the data set of the number of loan borrowers as the analysis sample, of which 34 are nondefaulting borrowers, 16 are default borrowers, and the credit rating is A (the highest grade), B, C, D, E, and HR (the lowest grade). With the chosen 8 indicators as input, the number of neurons in the input layer is set to 8, the distance function is set to the dist function, the network topology is $8 * 8$, that is, the competition layer has 64 neurons, and the number of network learning is 10000 times. The data set comprises 6 varieties of credit ratings, so there are 6 varieties of the data set, and hence 6 extra input layer neurons are added.

4.2. Simulation Results. Simulation experiments are managed based on the MATLABR2014b platform, and the results are revealed in Table 7. The overall accuracy of the model is 72.65%, the data volume of defaulted borrowers with E credit rating is small, and the sample learning is not adequate (Table 7). As a result, the accuracy rate is lower, and each credit rating model's prediction ability is more vital for the other two kinds of borrowers.

5. Recommendations

5.1. Improve Laws and Regulations and Complete Regulatory Mechanism. As a new variety of lending, P2P lending has implemented numerous impetus to developing the financial industry. However, a thorough regulatory mechanism has not been built in time with the industry's successful development, making a range of social problems in the early stage of development. Some of the regulations in the Contract Law of the People's Republic of China and the Opinions of the Supreme People's Court on the Trial of Lending Cases by the People's Courts are currently the relevant legal basis for P2P lending. With the spurt of developing P2P lending, related departments have published management measures to manage it one after another. In 2015, the Central Bank led the formulation of the Guiding Opinions on Promoting the Healthy Development of Internet Finance, which supervised network lending, a component of the seven major Internet finance industries. In 2016, the China Banking Regulatory Commission took the lead in composing the Interim Measures for the Management of the Business Activities of Online Lending Information Intermediaries to strengthen the supervision and management of the P2P lending information intermediaries' business activities. In 2019, the leading group for the special rectification of Internet financial risks and the leading group for the special rectification of online lending risks jointly issued the Notice on Strengthening the Construction of the Credit Investigation System in the P2P Lending Field to assist the access of P2P lending institutions in operating credit investigation system. Although relevant management measures have been introduced for regulation, the network platform carries significant regulation challenges while being convenient and productive. Clear regulatory responsibilities,

increasing supervision, establishing early warning mechanisms, reducing nonperforming loans, and improving the credit system have become essential safeguards to guarantee the healthy development of the P2P lending industry. Consequently, the relevant departments should continue to advance laws and regulations and enhance the regulatory mechanism to escort Internet finance development.

5.2. Rectify the Chaos of the Industry and Manage Healthy Development. In accordance with the monitor from National Internet Financial Security Technical Expert Committee, as of the end of July 2017, only 153 of the more than 2,000 platforms involved in cash loans and other small P2P lending businesses in the market had reached small P2P lending licenses. In 2017, the Office of the Leading Group for Special Work on Internet Financial Risks issued a Notice on the Immediate Suspension of the Establishment of Small P2P Lending Companies. It determined that the regulatory authorities of small P2P lending companies at each level shall not allow new small P2P lending companies and forbid new batches of microfinance companies to achieve small P2P lending companies across provinces (districts and cities). Apart from the shortage of qualifications, numerous online lending platforms do not have professional financial services and cannot present specialized services. Under the premise that China has increased the industry threshold and suspended approval, the industry should strengthen self-discipline within the industry, define industry self-regulation norms immediately, and reduce induced advertising. Consistently, interest rate repayment transparency advances practitioners' professional level and supervises the industry's healthy and orderly development.

5.3. Advance Financial Education and Supervise Rational Consumption. Since P2P lending has turned into a social issue, universities attach prominent importance to educational warnings and operational supervision. Nevertheless, most of the education process is based on case warning education and consumption concept guidance, and there is less education on financial literacy and financial planning. Accordingly, university students usually lack financial knowledge, credit knowledge, and inadequate financial management ability and are simply influenced by induced consumption. As a result, universities should encourage financial education, concentrating on improving financial literacy, financial knowledge, personal credit investigation, and other financial literacy. Conformably, relying on innovation and entrepreneurship education, schools should lead university students to build up a rational consumption concept, enhance the spirit of struggle, and transition to the university campus with higher personal qualities and a positive condition of hard work.

5.4. Strengthen One's Literacy and Rational Career Planning. University students themselves should launch a rational consumption outlook and a developmental consumption outlook, take the initiative to master the knowledge of

TABLE 7: Simulation outcomes.

| | Credit rating | Quantity | Number of correct classifications | — | Accuracy (%) | — |
|----------------------|---------------|----------|-----------------------------------|-------|--------------|--------|
| Nondefault borrowers | A | 12 | 10 | 83.3% | 76.5 | 72.65% |
| — | B | 8 | 6 | 75% | — | — |
| — | C | 8 | 6 | 75% | — | — |
| — | D | 6 | 4 | 66.7% | — | — |
| Default borrower | C | 5 | 3 | 60% | 68.8 | — |
| — | D | 4 | 3 | 75% | — | — |
| — | E | 2 | 1 | 50% | — | — |
| — | HR | 5 | 4 | 80% | — | — |

financial credit investigation, and improve their financial management ability. Under the premise of guaranteeing survival consumption, more expenditures will be utilized for developmental consumption to lay a stable foundation for further career development. At the moment, it will strengthen its innovation and entrepreneurship capabilities, making full use of professional advantages and personal expertise to accomplish part-time and entrepreneurial activities actively. Besides, a spirit of hard work and enterprising is organized, and financial independence is carried out while transitioning to society.

6. Conclusion

The attribute framework effect has an important influence in the lending decision-making process. The inductive slogans of online lending platforms can easily lead college students to blindly choose loans. Therefore, improving financial literacy and mastering financial knowledge are very important for college students. The loan risk assessment model based on Kohonen neural network has strong predictive ability. When the amount of data is sufficient, it shows better risk assessment ability. The application of this model to guide individuals with larger loan risk predictions can effectively avoid the behavior of borrowing and lending in breach of contract protects individual interests and forms a rational consumption concept.

The limitation of this study is that the influence of other frame effects on lending behavior is not discussed, and a comprehensive model is not established. Future research can carry out in-depth research on this content.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

- [1] Yiguan Think Tank, *The Special Research Report on China's Campus Consumer Financial Market [EB/OL]*, 2016, https://www.sohu.com/a/63680354_334205.
- [2] Tencent, . *Report on the Real Demand for Undergraduates' Installments of P2P Lending [EB/OL]*, Tencent, Shenzhen, China, 2015, <https://tech.qq.com/a/20150121/011525.htm>.
- [3] Beijing Yixin Zhicheng Credit Management Co., Ltd. and CMRC, *Kindling Project: National Research Report on undergraduates' Credit Awareness [EB/OL]*, 2015, <http://sf.ruc.edu.cn/?p=388>.
- [4] *Research Consumption Insight Report of College Students in 2018 [EB/OL]*, 2018, <http://report.iresearch.cn/report/201808/3262.shtml>.
- [5] CSCU, *2017 Survey Report on the Credit Status of Chinese Undergraduates [EB/OL]*, 2017, <https://www.iyiou.com/p/52180.html>.
- [6] Y. Jing, *Research on the Consumption Behavior and Psychology of Undergraduates' Campus P2P Lending*, Yunnan University, Kunming, China, 2017.
- [7] L. Fan, R. Gan, X. Fan, S. Xu, Yu Gao, and M. Li, "Analysis of undergraduates' problems and countermeasures of P2P lending-taking the universities in chengong city as an example," *Times Finance*, vol. 11, pp. 106-107, 2017.
- [8] X. Fu, "Research on the deconstruction and inhibition of undergraduates' P2P lending consumption behavior," *Journal of Southwest Normal University (Natural Science Edition)*, vol. 8, pp. 128-133, 2017.
- [9] L. Cong, Y. Yuan, and X. Yan, "A study on the correlation between undergraduates' consumption and P2P lending behavior based on questionnaire survey-taking one of the universities in Shanghai as an example," *Consumer Economics*, vol. 94, no. 1, pp. 94-96, 2019.
- [10] D. Kahneman and A. Tversky, "The framing of decisions and the psychology of choice," *American Association for the Advancement of Science*, vol. 211, no. 4481, pp. 453-458, 1981.
- [11] I. P. Levin, S. L. Schneider, and G. J. Gaeth, "All frames are not created equal: a typology and critical analysis of framing effects," *Organizational Behavior and Human Decision Processes*, vol. 76, no. 2, pp. 149-188, 1998.
- [12] G. Wen, F. Xu, H. Yu, B. Huang, and L. Wang, "The psychological mechanism and influencing factors of the characteristic frame effect," *Advances in Psychological Science*, vol. 19, no. 12, pp. 1822-1833, 2011.
- [13] T. Kohonen, *Self-Organization and Associative Memory*, Springer, vol. 8pp. 3406-3409, Berlin, Germany, 3rd edition, 2006.
- [14] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 1109, no. 10, pp. 1-20, 2020.
- [15] C. Sun, Y. Sun, and K. Xie, "Mind-evolution-based machine learning:an efficient approach of evolution computation," in *Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No.00EX393)*, Hefei, China, June 2000.
- [16] Z. Zhang, N. Zhou, and J. Fan, "P2P online loan borrower credit risk assessment model-optimize SKohonen neural network based on thought evolution algorithm," *Taxation and Finance*, vol. 20, no. 10, pp. 84-85, 2020.

Research Article

Artificial Intelligence and Neural Network-Based Shooting Accuracy Prediction Analysis in Basketball

Hongfei Li ¹ and Maolin Zhang ^{2,3}

¹Shanxi Agricultural University, Jinzhong 030801, China

²Hoseo University, Asan 31499, Republic of Korea

³Business College of Shanxi University, Taiyuan 030031, China

Correspondence should be addressed to Maolin Zhang; zhangmaolin@bcsxu.edu.cn

Received 13 April 2021; Revised 30 April 2021; Accepted 12 May 2021; Published 3 June 2021

Academic Editor: Jianhui Lv

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

In order to improve the accuracy of shooting in basketball. A shooting accuracy prediction method based on the convergent improved resource allocating network (CIRAN) online radial basis function neural network (RBFNN) is proposed, and the RBFNN learning algorithm is improved. Through the collection of shooting motion images, feature point extraction, and edge contour feature extraction, the shooting motion trajectory is obtained. Using the online neural network based on the CIRAN learning algorithm to predict the accuracy of shooting, this method analyzes the radial basis function (RBF) network. Based on the RBF analysis, the number of network layers and the number of hidden layer neurons are adjusted and optimized. In order to improve the prediction accuracy of shooting in basketball, a method based on. Through the analysis, it can be known that the accuracy of both the traditional RBFNN and the CIRAN-based online neural network for the prediction of shooting accuracy is above 70%. The prediction accuracy of the online neural network for shooting is higher than that of the traditional one. This is mainly because the online update function of the learning algorithm can better adjust the corresponding structure with the development of the game and has a better generalization ability. In addition, because the CIRAN learning algorithm introduces the hidden layer neuron deletion strategy, its network structure is simpler than that of the traditional one, the number of hidden layer neurons is less, and the running time required is less, which can better meet the real-time requirements and provide a more scientific method for basketball training.

1. Introduction

With the development of computer image processing technology, embedded digital image and video information analysis methods are used to carry out image analysis and feature extraction of sports, establish a feature analysis model of sports images, and improve the ability of feature identification and movement correction of sports. In basketball, the accuracy of shooting determines the key to scoring. It is necessary to study the extraction of basketball players' shooting motion trajectory, combined with the image feature analysis method of basketball shooting, to reconstruct and quantitatively track the basketball players' shooting motion trajectory [1], establishing the image analysis model of basketball players' shooting motion

trajectory and improving the calibration ability of basketball players' shooting motion. The research on the extraction method of basketball players' shooting motion trajectory has attracted great attention. Based on Harris corner detection, a method for extracting characteristics of lower-upper extremity action in basketball [2], this method first maps the spatial distribution of the pixel gray level in the upper extremity action area of the image, using Gaussian mixture model standards and normalized athletes and the contours of the lower-upper limbs of the strong smash, the Harris corner detection method is used to carry out the affine invariant closed area enhancement processing on the continuous motion images of the athletes, and the corner detection of the contours of the upper limbs of the athletes is performed to complete the lower-upper limbs of the

basketball smash. The action feature extraction method, however, has low accuracy in extracting the action features of the lower-upper limbs of basketball players. The volleyball player's motion trajectory optimization recognition method based on chaos theory [3] is based on the background difference principle to detect the player's motion trajectory, and the particles of the color histogram are used. Filtering for dynamic tracking, fusion with chaos theory to reconstruct the phase space of the athlete's motion trajectory, the chaotic invariant representing the athlete's motion trajectory is extracted from the reconstructed phase space, the motion trajectory with three-dimensional space characteristics is converted into a one-dimensional motion trajectory, and the optimized recognition of the volleyball player's trajectory is completed. However, this method has a low accuracy in predicting the trajectory of volleyball players. To solve the above problems, this paper extracts the basketball player's shooting motion trajectory based on the block growth optimization algorithm, extracts the edge contour feature of the collected basketball player's shooting motion trajectory image, establishes the image fusion model of the basketball player's shooting motion trajectory, and extracts the image feature of the shooting motion trajectory, and the corner points are marked to realize the extraction of the basketball player's shooting motion trajectory. Finally, the simulation experiment analysis is carried out to show the superior performance of this method in improving the ability of basketball player's shooting motion trajectory extraction.

At present, in the field of video-based analysis of moving human bodies, most of the research results are mainly on the discussion of motion behavior recognition, but there is no in-depth study on motion behavior prediction. However, in the process of real life, the prediction of human motion behavior based on videos has more practical value than behavior recognition [4]. For example, in many crowded public places, video-based human motion behavior prediction technology can be used to predict possible criminal behaviors in surveillance videos, and prompt criminal behavior prediction alerts the public security department to facilitate timely actions by the public security department corresponding solutions. In this way, it can not only reduce the manpower, material, and financial resources that the public security department spends on security investigations but also effectively prevent sudden crimes. In the field of sports, a comprehensive set of human motion behavior prediction technology in videos can accurately obtain the game data of some excellent teams during related sports training and tailor a sports behavior prediction discriminator for each athlete [5]. Through the discriminator, the coach can distinguish the difference between the movement made by the exercise and the standard movement, so as to adjust the training intensity in a targeted manner. In the course of the basketball game, there are three sports behaviors taken by players throughout the game, namely, shooting, passing, and dribbling. Basketball game is a team sport with multiple players; players must cooperate with each other in order to win the game. During the basketball game, players will be affected by various factors such as players or opponents when they choose related sports

behaviors. Therefore, the process of basketball player behavior prediction has a high degree of complexity, which brings a certain degree of difficulty to the establishment of the behavior prediction mathematical model [6]. Combining this feature, we can also regard the prediction of basketball behavior as a nonlinear problem. In the prediction of solving nonlinear problems, artificial neural networks are widely used in the field of nonlinear system modeling due to their own self-adaptive and nonlinear characteristics. In this regard, this article combines the advantages of artificial neural networks, proposes a prediction method based on the CRIAN algorithm, and elaborates its implementation in detail.

The main contribution of this paper is to propose a CIRAN-based online RBFNN shooting prediction method; the accuracy of shooting prediction is above 90%. The rest of the paper is organized as follows. Section 2 summarizes domestic and foreign research work in the analysis of movements. Section 3 introduces shooting image collection and motion trajectory extraction optimization. Experimental results are reported in Section 4, and finally, Section 5 concludes this paper.

2. Related Work

At present, many universities and research institutions at home and abroad have carried out research on the introduction of digital video technology into auxiliary sports. Well-known research units abroad include the Media Analysis Laboratory of the Massachusetts Institute of Technology, the Digital Video Multimedia Laboratory of Columbia University, the School of Engineering and Applied Sciences, the University of Rochester, the Department of Computer Science at the University of Texas at San Antonio, the University of Delft in the Netherlands, Multimedia Analysis Laboratory and Microsoft Asia Research Institute, American Research Institute, Mitsubishi Electric American Research Institute, and Singapore Institute of Information and Communication Research. Domestic research institutions mainly include Advanced Human-Machine Communication Laboratory, Institute of Computing Technology, Chinese Academy of Sciences, State Key Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Institute of Digital Media, Peking University, and Visual Intelligence Interface Laboratory, Harbin Institute of Technology.

According to the complexity of the current research objects, the existing literature can be divided into two levels: the analysis of single-person individual sports and the analysis of multiperson team sports.

The current research on individual sports is mainly reflected in the recognition and analysis of athletes' movements. In 1996, Pennsylvania State University in the United States developed the "computer graphics for the improvement of springboard diving" system to help diving coaches and athletes to strengthen their understanding of the entire body posture during diving; some researchers in National Chiao Tung University analyzed the motion trajectory of a tennis player which is used to judge the volley or the baseline ball in a

tennis match [7]; Tsinghua University's "video-based diving posture analysis system" uses target detection and tracking technology to extract sports targets from diving videos, and to compare sports targets, video synthesis is performed; Roh et al. proposed an action recognition method based on the curvature scale space template [8–10] and applied it to the player's action recognition in a tennis match. The actions of ice hockey and football players are analyzed and studied using the histogram of oriented gradient (HOG) and hidden Markov model (HMM) to identify the direction of the players' movement [11]; Su et al. proposed a method of recognizing periodic motion [12]; Zhong et al. proposed an appearance-based method [13] to identify and label players' shoulder swings in tennis matches (overshoulder swing) and a series of actions such as forehand swing and backhand swing. This method has been further expanded in the follow-up work. Combining the position information of the players and the ball, an action based on reasoning ideas is proposed. Ramasso et al. used the TBM (transferable belief model) [14] to identify back jump, pole vault, triple jump, and other actions in track and field competitions; Roh et al. recognized various postures [15]; Min et al. obtained the dance trajectory by tracking some key points of the body parts of the color bud dancer [16] and realized the automatic analysis of dance movements; Tong et al. realized the recognition of the four swimming styles in the competition video [17].

At present, there are many research studies on the contour tracking and extraction of basketball shooting motion video images, and the relative research has also produced certain results. Based on the Surendra background difference, a basketball shooting action video image contour tracking extraction method [18–20], this method first uses the Surendra background subtraction method to establish a basketball shooting action background model, giving players a complete shooting action and obtaining dynamic motion area; based on this, the contour tracking and extraction of the video image of the basketball shooting action is completed. This method is relatively simple, but there is a problem of large limitations of the method. The video image contour tracking and extraction method of basketball shooting action is based on visual analysis [21, 22]. This method first detects the edge contour of the shooting image and gives the dynamic feature segmentation threshold of the basketball shooting action, which is used as the basis to complete the basketball shooting action. The video image contour tracking extraction method has high marking efficiency, but when the current method is used for marking, the dynamic pixel information characteristics of basketball shooting actions cannot be given, and there is a problem of low contour tracking extraction accuracy. A basketball shooting action video image contour tracking extraction method based on figure contour feature extraction [23, 24] first extracts the dynamic figure edge contour feature points of the shooting action and uses the bright spot model diffraction method to achieve visual penetration. According to the technical characteristics of basketball shooting action, the contour tracking and extraction of basketball shooting action video image can be completed. This method has high marking accuracy, but there is a problem that the marking process is more cumbersome.

3. Shooting Image Collection and Motion Trajectory Extraction Optimization

3.1. Shooting Image Collection in Basketball. In order to achieve the extraction of the basketball player's shooting motion trajectory, the video sensor image tracking method is used to collect the image of the basketball player's shooting motion trajectory; the edge contour feature extraction of the collected basketball player's shooting motion trajectory image and the fuzzy decision method are used for state recognition and action trajectory planning. The grid segmentation method is used to divide the basketball sports video images collected by the video into feature blocks, α is the angle between the projection speed direction and the horizontal direction, that is, the shooting angle, v is the shooting point of the shot speed, g is the acceleration of gravity, the basketball is thrown at $t = 0$, and $f(x, y)$ is the trajectory function of the basketball which is defined as follows:

$$f(x, y) = x \tan \alpha - \frac{x^2 g}{2v^2 \cos^2 \alpha}. \quad (1)$$

In a single scale, the pixel space of the basketball motion image feature collection is defined as follows:

$$a = \frac{1}{\nabla f(x, y)} \left(\frac{\partial f}{\partial y} i - \frac{\partial f}{\partial x} j \right), b = \frac{1}{\nabla f(x, y)} \left(\frac{\partial f}{\partial y} i + \frac{\partial f}{\partial x} j \right). \quad (2)$$

Using the adaptive weighting method for threshold modulation, the threshold of image grid segmentation is M , the initial value of the characteristic points of the basketball flight trajectory is calculated, and the edge pixel feature decomposition method is used for image template matching, and the number of template pixel blocks of the image is obtained as

$$B(x_n, y_n) = \prod_{x_i \in N} \prod_{m=1}^M \beta_m g(x_{ij}, y_{ij} | \mu_m, \sigma_m^2). \quad (3)$$

The position conversion set in basketball shooting is h_c , and the target configuration θ_g is unknown. The multiscale wavelet decomposition method [25] is used to segment the gray value in the image. The segmentation threshold meets $\nabla_x = [1, -1]$ and $\nabla_y = [1, -1]^T$ and produces the high-frequency part of the basketball shooting image. $y = [\nabla_x h', \nabla_y h']$ represents the low-frequency component of the image pixel value. The image feature is collected through normalization processing, and the image collection result obtained is

$$\min_{x, m} \lambda \|x \otimes m - y\|_2^2 + \frac{\|x\|_1}{\|x\|_2} + \delta \|m\|_1. \quad (4)$$

When the output pixel feature set of basketball shooting meets the constraint condition $k > 0$, $\sum_i m_i = 1$, the collected motion image can better reflect the shooting angle information and flight trajectory characteristics.

3.2. Feature Points' Extraction. On the basis of optimizing the collection of the image, the characteristic corner points of the basketball shooting trajectory are extracted from the image. In the state of motion, the state equation of the scattering model of the basketball shooting trajectory is defined as follows:

$$S(x) = J(x)t(x) \frac{v' \times \varepsilon_0}{\eta} + A(1 - t(x)). \quad (5)$$

Among them, $J(x)$ is the strength of the shot force in a fixed-point shot, x is the edge pixel sequence of the collected original image, v' is the motion function of the multicontour viewpoint switching of the shooting action, ε_0 is the standard deviation of the Gaussian function, and η is the direction of the filter.

In shooting, the Monte Carlo mathematical expectation of the edge corners of the image is extracted, and the pixel value extracted from the characteristic corners of the image in the basketball shooting motion is defined as follows:

$$\tau(Z; D_X) = \sum_{i>j} |d_{ij}(Z) - d_X(x_i, x_j)|^2. \quad (6)$$

Among them, $d_{ij}(Z)$ is the fitness judgment, which is a Euclidean distance; $d_X(x_i, x_j)$ represents the three-dimensional coordinate component value in sports shooting and fixed-point shooting.

When the basketball is flying in the air, the trajectory deviation will occur. Due to the influence of wind resistance and other factors, the attenuation coefficient of the flight trajectory is $e^{-\delta d(x)}$. Assuming that $A(1 - e^{-\delta d(x)})$ represents the edge pixel error of the shooting motion trajectory, the pixel value of the edge contour corner point is defined as follows:

$$p_t = C_t \sum_{x_i \in n} m(\|x_i\|^2). \quad (7)$$

Among them, C_t is a normalized constant. The statistical characteristic analysis of basketball shooting angles and the calculation of characteristic corner points are performed through the spatial adaptive correction method. The number of image pixels is defined as follows:

$$G(x) = \frac{S(x) - A}{\max(t(x), t_0)}. \quad (8)$$

Among them, $G(x)$ is the degree of similarity of gray values, and $\max(t(x), t_0)$ is the maximum visual deviation of the basketball caused by changes in shooting mechanics and other factors.

The number of information features with respect to the basketball player's shooting motion image is x_k . The discrete pixel sequence reconstruction method is used to reconstruct the three-dimensional basketball player shooting motion image. The video information collection method is used to obtain the image sampling output result. The gray information fusion is performed on the basketball shooting action feature points to obtain the basketball shooting action feature, and the video tracking fusion formula for points is defined as follows:

$$\begin{aligned} x(k+1) &= G_i(k)x(k) + w_i(k), \quad i = 1, 2, \dots, m, \\ u(k) &= R_i(k)x(k) + v_i(k), \quad i = 1, 2, \dots, m. \end{aligned} \quad (9)$$

Among them, $w_i(k)$ and $v_i(k)$ are the state feature quantity and observation feature quantity extracted from the basketball shooting action feature. $G_i(k)$ and $R_i(k)$ obey the mean value of 0, and the variance is the normal distribution of x_k . According to the above analysis, the method of video sensor image tracking is used to collect basketball players' shooting motion trajectory images, and the shooting motion trajectory extraction and image information monitoring are performed according to the image collection results.

3.3. Motion Trajectory Extraction Optimization. Based on the aforementioned video sensor image tracking method for basketball player shooting motion trajectory image acquisition and edge contour feature extraction, the basketball player's shooting motion trajectory prediction and feature extraction are carried out, and the shooting motion trajectory extraction is optimized. This paper proposes an extraction method of basketball players' shooting motion trajectory based on block growth algorithm. The image fusion model of basketball player's shooting motion trajectory is established, the area linear growth method is combined to extract the image feature of basketball player's shooting motion trajectory, and the corner points are marked. The texture structure information of the basketball shooting action feature points is changed, the smooth area is distinguished, and the basketball shooting action is calculated. The amplitude modulation information of the characteristic defect image component, the component $x(t)$ of the pixel feature point of the n shooting action image of the image r_n , is defined as follows:

$$x(t) = p_t \sum_{i=1}^n e_i + r_n. \quad (10)$$

Among them, p_t is the pixel value of the edge contour corner point, and e_i represents the pixel value of the empirical mode decomposition of the basketball shooting action feature point. Using computer vision analysis, the basketball shooting action feature image is divided into N segmented regions, and N segmented regions perform multidimensional search iterations. Assuming that the two-dimensional feature segmentation function of the basketball player's shooting motion image satisfies $n \in N(0, \sigma_n^2)$, where σ_n^2 is the variance of the noise, combined with the regional linear growth method for the feature extraction of the basketball player's shooting motion trajectory image and the corner point labeling, the corner point distribution satisfies

$$w(i, j) = \frac{1}{x(t)} \exp\left(-\frac{d(i, j)}{h^2}\right) \tau(Z; D_X). \quad (11)$$

Among them, $\exp(-d(i, j)/h^2)$ is the least square feature quantity of the basketball player's shooting motion image, and h is the block fusion information entropy of the basketball player's shooting motion image. The fuzzy correlation fusion method is used to obtain the regional growth

function model $g(x, y) = \xi(x, y) + \psi(x, y)$ of the basketball player's shooting motion trajectory. Among them, $\xi(x, y)$ is the variance of the pixel distribution, and $\psi(x, y)$ represents the pixel intensity of the basketball player's shooting motion image. Through the regional linear growth analysis, when $\varphi_m(x, y) \in \{1, -1, 0\}$ is satisfied, the expression of the characteristic component of the regional linear growth of basketball shooting action is defined as follows:

$$p(\varphi_m(x, y)) = \begin{cases} \frac{r}{4}, & \varphi_m(x, y) = -1 \\ 1 - \frac{r}{2}, & \varphi_m(x, y) = 0 \\ \frac{4}{r}, & \varphi_m(x, y) = 1 \end{cases}. \quad (12)$$

Among them, $\varphi_m(x, y)$ is the regional linear growth characteristic of basketball shooting action, and r is the detection threshold of the basketball shooting action trajectory, $0 \leq r \leq 1$. The three-dimensional feature quantity of the basketball player's shooting motion image is extracted, and the three-dimensional visualization surface reconstruction method is used to reconstruct the motion track. According to the corner point distribution and edge contour distribution of the basketball player's shooting motion trajectory image, the characteristics of the basketball player's shooting motion trajectory are extracted, and the expression of the basketball shooting motion trajectory $p(x, t)$ is defined as follows:

$$p(x, t) = -\sigma \frac{\Delta u(x, t)}{p(\varphi_m(x, y))x(k)}. \quad (13)$$

Among them, $\Delta u(x, t)$ is the associated pixel point of the global threshold segmentation of the basketball player's shooting motion image, and σ is the feature quantity of the basketball player's shooting motion trajectory tracking. In summary, the optimization of the trajectory extraction of the basketball player's shooting is realized.

3.4. Radial Neural Network Prediction Model. The process of predicting the accuracy of basketball players' shooting has a relatively high complexity, which brings a certain degree of difficulty to the establishment of mathematical models of behavior prediction. Combining this feature, we can also regard the prediction of basketball behavior as a nonlinear problem. In the prediction of solving nonlinear problems, artificial neural networks are widely used in the field of nonlinear system modeling due to their own self-adaptive and nonlinear characteristics. In this regard, this paper combines the advantages of artificial neural networks and proposes a behavior prediction method based on an improved RBF algorithm.

RBFNN is an extremely efficient feedforward neural network [26]. Compared with other neural networks, RBFNN has two unique advantages of best approximation performance and global optimal characteristics and has the

basic characteristics of other neural networks such as simple structure and fast training speed. It is a typical feedforward neural network composed of three layers. The function of the input layer node is to pass the signal to the hidden layer; the hidden layer node is composed of radial basis functions; the output layer node is usually a simple linear function. In gamma, the transformation from the input layer to the hidden layer is nonlinear. The role of the hidden layer is to perform a nonlinear transformation on the input vector, and the transformation from the hidden layer to the output layer is linear, that is, the output of the network is a hidden node, the linear weighted sum of the output. The topology of the specific single-output RBF network is shown in Figure 1.

The commonly used hidden layer radial basis functions have the following forms:

(i) Multiquadric function:

$$\phi(r) = (1 + (\varepsilon r)^2)^{1/2}, \quad \text{among them } \varepsilon > 0, r \in R. \quad (14)$$

(ii) Inverse multiquadric function:

$$\phi(r) = \frac{1}{(1 + (\varepsilon r)^2)^{1/2}}, \quad \text{among them } \varepsilon > 0, r \in R. \quad (15)$$

(iii) Gauss function:

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad \text{among them } \sigma > 0, r \in R. \quad (16)$$

In formulas (14)–(16), ε is a constant, and r is the expansion constant of the RBF.

The hidden layer in the RBFNN in this paper uses the Gaussian function as the radial basis function, and the specific form is defined as follows:

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|^2}{\sigma_k^2}\right), \quad k = 1, 2, \dots, K. \quad (17)$$

Among them, x represents the l -dimensional input vector; μ_k is the center of the k -th radial basis function, a vector with the same dimension as x , and σ_k is the width of the radial basis function of the k -th hidden layer neuron. $\|x - \mu_k\|$ is the Euclidean norm of the vector $x - \mu_k$, which usually represents the radial distance between x and μ_k . $\phi_k(x)$ has a unique maximum value at μ_k , and as $\|x - \mu_k\|$ increases, it quickly decays to 0. For a given input $x \in R^l$, only a small part of the input close to the center μ_k is activated, that is, the radial basis function is a function with local induction characteristics. Suppose the number of hidden layer neurons of the RBFNN is K , then the output form of the network can be defined as follows:

$$\hat{y} = f(x) = w_0 + \sum_{k=1}^K w_k \exp\left(-\|x - \mu_k\|^2 / \sigma_k^2\right). \quad (18)$$

Among them, w_0 represents the bias, which is the connection weight between the hidden layer and the output layer.

It can be seen from the structure of the neural network that the construction and training of one is to determine the

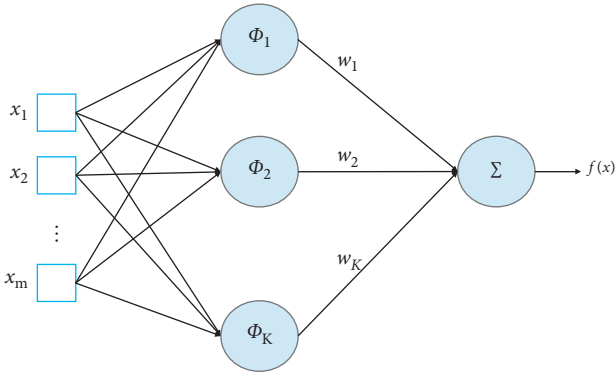


FIGURE 1: RBF network topology diagram.

number of hidden layer neurons K , the center μ of the radial basis function of each hidden layer neuron, the width σ , and the connection weight of the hidden layer to the output layer. Value w is the learning process of these parameters to complete the required mapping from the input to the output. For the RBFNN, the performance of its network mainly depends on the selection strategy of hidden layer neurons and the update process of the connection weights of the hidden layer to the output layer in the network.

The learning algorithm of the RBFNN can be divided into two categories according to the time: offline learning, also called batch learning, and online learning, also called sequential learning.

(1) *Offline Learning.* The offline training method trains the RBF network with a fixed structure. The center is extracted from the distribution pattern of the training sample space. After the center point is selected, the structure of the network is determined. When the data mode changes, the network cannot make corresponding changes. Although the network characteristics can be adjusted as much as possible by adjusting the weights, this method is not fundamental, and the adjustment range is limited. Especially when there is a sample that has not been learned, the network will lose its recognition ability. Therefore, the offline method is not strong in fitting time-varying systems.

(2) *Online Learning.* The offline learning mode is adopted, and the network parameters are adjusted only after complete learning is performed with all training samples, that is, an epoch. The learning process is one epoch after one epoch, until the network parameters stabilize and the average output error on the entire training set converges to a certain minimum, the learning can be ended. When using online learning algorithms, after each training sample enters the network and is calculated, the network parameters are adjusted. Specifically, suppose an epoch contains a training sample $((X(1), Y(1)), \dots, (X(N), Y(N)))$. When the first sample $(X(1), Y(1))$ enters the network, the network output is calculated. Then, the learning algorithm is run to update the network parameters. This process is repeated until the learning of the first training sample is completed. The online RBFNN learning algorithm is actually a variable-structure

training method. The so-called “variable structure” means that, in the training process, the current RBFNN is allocated or added hidden layer neurons according to the input samples in real time, and the network topology is dynamically constructed. At the same time, in the process of correcting the weight, the position of the “center” is further corrected. Adopting the concise “central selection” principle and effective weight training algorithm can make it have the characteristics of real time and rapidity.

At present, many scholars have conducted research on online learning algorithms. Typical algorithms are as follows:

3.4.1. *RAN (Resource Allocating Network) Learning Algorithm.* Resource allocating network (RAN) reflects the complexity of the original function to be simulated by adjusting the number of hidden layer units [27]. In Platt’s RAN learning algorithm, the “novelty” of the training sample is used as the standard for introducing hidden layer neurons, and then the network parameter LMS (least mean square) is updated through the algorithm. When a data point is far away from the existing basis function center and there is a large error between the network output and the actual output, the data point is considered “novel.” If the input sample does not meet the requirements of novelty, no hidden layer neurons are added, but the LMS algorithm is started to adjust the existing network parameters including center, width, and weight.

3.4.2. *RANEKF (RAN Extended Kalman Filter) Learning Algorithm.* The RANEKF algorithm is an improvement on the basis of the RAN learning algorithm. Its hidden layer neuron introduction strategy is the same as that of the RAN learning algorithm. The difference is that the adjustment of network parameters uses the extended Kalman filter (EKF) instead of the LMS method [28]. The extended Kalman filter method has a faster convergence rate than the LMS method but requires more computer resources. However, with the development of computer hardware technology, the EKF method has more advantages in the case of a small problem.

3.4.3. *MRAN (Minimal RAN) Learning Algorithm.* The MRAN algorithm not only combines the hidden layer node growth criterion of the RAN learning algorithm but also introduces a hidden layer neuron deletion strategy in order to obtain an ideal minimum neural network structure. Cheng proposed a method to delete hidden nodes in the batch learning algorithm in 1994 [29]. In this method, each epoch must check the weight of each hidden layer node, and those hidden layer nodes with a weight value less than a certain threshold will be deleted.

Inspired by Cheng’s method, L. Ying Wei et al. proposed another hidden node deletion strategy for the MARN algorithm and called this RBFNN with the addition of the deletion strategy and the RMS sliding window MRAN (minimal resource allocating network). The first difference between MRAN’s deletion strategy and Cheng’s method is

that the MRAN is for sequential learning algorithms instead of batch learning algorithms. In addition, the MRAN deletion strategy not only considers the weight of the hidden node [30, 31] but also considers the output of the hidden node.

3.5. Shooting Prediction Based on the CIRAN Online RBFNN. The MRAN algorithm is developed on the basis of the RAN learning algorithm and the RANKEF learning algorithm. The learning process of the MRAN algorithm involves the introduction of new hidden layer neurons, the adjustment of network parameters, and the deletion of hidden layer neurons. The learning algorithm is defined as follows:

(i) For each input, calculate

$$\phi_k(x) = \exp\left(\frac{-\|x_i - \mu_k\|^2}{\sigma_k^2}\right), \quad k = 1, 2, \dots, K,$$

$$f(x_i) = w_0 + \sum_{k=1}^K w_k \phi_k(x_i),$$

$$d_i = \min_{1 \leq k \leq K} \|x_i - \mu_k\|, \quad (19)$$

$$\varepsilon_i = \max\{\gamma^i \varepsilon_{\max}, \varepsilon_{\min}\},$$

$$e_{rms}^i = \sqrt{\sum_{j=i-B+1}^i \frac{\|e_j\|^2}{B}}.$$

Among them, d_i represents the Euclidean distance from the center of the hidden layer closest to x_i , ε_{\max} represents the maximum distance between input data, ε_{\min} represents the minimum distance between input data, and $0 < \gamma < 1$ is an attenuation coefficient. As the input data increase, ε_i decreases at an exponential rate until ε_{\min} . B is the width of the RMS sliding window (generally 40–50, empirical setting is required), and e_{rms}^i is the root mean square (RMS) of the input error after the window is added when the i sample enters the RBFNN.

(ii) If three conditions $|e_i| > \lambda$, $d_i > \varepsilon$, and $e_{rms}^i > \lambda_i$ (where λ is the desired approximation accuracy and λ_i is the threshold set in advance) are met at the same time, then a new hidden layer neuron is added to the network, and $K = K + 1$; then, the hidden layer neuron parameters are the following three formulas:

$$\begin{aligned} w_{K+1} &= e_i, \\ \mu_{K+1} &= x_i, \\ \sigma_{K+1} &= \kappa d_i. \end{aligned} \quad (20)$$

Among them, κ is the overlap factor, which determines the response width of hidden layer neurons. If the conditions are not satisfied, then the EKF is used to update the network parameters.

(iii) Update the network parameters according to the following formula:

$$\begin{aligned} u_i &= u_{i-1} + K_i e_i \\ K_{i(\text{exnode}_y)} &= P_{i-1} A_i [R_i + A_i^T P_{i-1} A_i]^{-1}, \\ P_i &= [I_{z \times z} - K_i A_i^T] P_{i-1} + I_{z \times z}. \end{aligned} \quad (21)$$

Among them, $u_i = [w_0^i, w_1^i, \mu_1^i, \sigma_1^i, \dots, w_K^i, \mu_K^i, \sigma_K^i]^T$ represents the parameter state after the i sample enters the network, and $K_{i(\text{exnode}_y)}$ is the Kalman gain matrix:

$$K_{i(\text{exnode}_y)} = P_{i-1} A_i [R_i + A_i^T P_{i-1} A_i]^{-1}. \quad (22)$$

Among them, $z = \text{node}_y + K_i \times (j + \text{node}_y + 1)$ is the number of network parameters, node_y is the number of output nodes, and node_y is the variance matrix of the measurement noise. $A_i = \nabla_v f(x_i)$ is the gradient matrix of $f(x)$ with respect to the parameter vector v_i , and P_i is the error covariance matrix everywhere. Use the following formula to update:

$$P_i = [I_{z \times z} - K_i A_i^T] P_{i-1} + ss_0 I_{z \times z}. \quad (23)$$

Among them, $I_{z \times z}$ is the identity matrix, and ss_0 is a scalar, representing a random step length, used to determine the size of a random walk in the gradient direction. When a new hidden layer node is introduced into the network, the dimension of P_i is increased, and new rows and columns need to be added to P_{i-1} :

$$P_i = \begin{pmatrix} P_{i-1} & 0 \\ 0 & p_0 I_{z_1 \times z_1} \end{pmatrix}. \quad (24)$$

Among them, p_0 is a parameter value initially estimated, here is the covariance of sample data x_i and y_i , and z_1 is the number of new parameters added due to the introduction of new hidden layer nodes, $z_1 = l + \text{node}_y + 1$.

(iv) Calculate the output vectors $(\rho_{k1}^i, \dots, \rho_{kj}^i, \dots, \rho_{k\text{node}_y}^i)$ and $\|\rho_{j,\max}^i\|$ of all hidden nodes, which represent the maximum absolute value of all hidden nodes to the j output unit when sample i is input. Calculate the normalized output vector of each hidden node:

$$r_{kj}^i = \frac{\|\rho_{k,j}^i\|}{\|\rho_{j,\max}^i\|}, \quad k = 1, \dots, K. \quad (25)$$

During the continuous input of N_w samples, if $r_{kj}^i < \lambda_2$ is established, then the k hidden node can be deleted, and the dimension of P_i can be reduced accordingly to facilitate the adjustment of the EKF parameters in the next step.

The main problems of the aforementioned MRAN learning algorithm are as follows: firstly, due to the use of the extended Kalman filter to adjust the network parameters, the parameters must be updated in each iteration, which leads to the process of updating the parameters with the hidden layer neurons. The scale of the matrix is very large, which increases the computational complexity of the RBFNN structure, causes the algorithm to calculate too much burden, consumes a lot of computer resources, and limits the real-time application of the MRAN algorithm; then, initializing the algorithm, there are too many parameters, and improper selection of initialization parameters will greatly reduce the performance of the algorithm. Sometimes, an exhaustive method has to be used for multiple trials. This will consume a lot of time and cause the algorithm promotion performance to drop significantly.

In this regard, this paper proposes a CIRAN (convergent improved RAN) improved algorithm, which is mainly reflected in the following:

- (i) In order to reduce the initialization parameters of the algorithm, the idea of the GAP-RBP algorithm is absorbed, only the parameters of the hidden layer neuron closest to the current input data are updated, and the definition and estimation formula for measuring the importance of the hidden layer neuron are introduced, which reduces the algorithm number of initialization parameters and improves the generalization performance of the algorithm to a certain extent. The importance of hidden layer neurons is defined as follows:

$$E_{\text{imp}}(k) = \|w_k\|_q \left(\int_X \exp\left(-\frac{q\|x - \mu_k\|^2}{\sigma_k^2}\right) p(x) dx \right)^{1/q}. \quad (26)$$

Among them, $\mu_k = (\mu_{k,1}, \dots, \mu_{k,j})^T \in R^l$ and σ_k ($k = 1, \dots, K$) are the center and width of the radial basis function, and l is the dimension of the input vector. If the importance of a hidden layer neuron k is less than the realization of the given learning accuracy λ , then the hidden layer neuron k is considered no longer important and is deleted from the network; otherwise, the hidden layer neuron k should be retained in the network.

- (ii) The dynamic adjustment method of the coincidence degree threshold is introduced into the algorithm so that the CIRAN need not set the values of the parameters ε_{max} , ε_{min} , and γ in the MRAN. Parameter γ can be dynamically obtained during the execution of the algorithm. Its updated formula is

$$\varepsilon_i = \max\left(0, (1 - \tau) \left| \frac{f(x_i)}{e_i} \right| \right). \quad (27)$$

Among them, τ is the expected accuracy of the single-point output, $f(x_i)$ is the current actual output, and e_i is the current output error.

- (iii) A new adaptive adjustment method for the width of the radial basis function of hidden layer neurons is

$$\left\{ \begin{array}{l} K = \frac{e_i}{e_i \sqrt{-\ln \lambda / e_i}} \\ \sigma_{K+1} = \frac{e_i d_i}{e_i \sqrt{-\ln \varepsilon / e_i}} \end{array} \right\}. \quad (28)$$

So, the CIRAN learning algorithm can be defined as follows.

Input: given estimated error λ and single-point output expected accuracy τ . For the input sample (x_i, y_i) , $x_i \in R^l$,

- (i) Calculate the network output:

$$f(x_i) = w_0 + \sum_{k=1}^K w_k \exp\left(-\|x_i - \mu_k\|^2 / \sigma_k^2\right). \quad (29)$$

Among them, K is the number of hidden layer nodes.

- (ii) Calculate the following quantities in the novelty criterion:

$$\begin{aligned} \varepsilon_i &= \max\left(0, 1 - (1 - \tau) \left| \frac{f(x_i)}{e_i} \right| \right), \\ e_i &= y_i - f(x_i), \\ d_i &= \|x_i - \mu_{ir}\|. \end{aligned} \quad (30)$$

Among them, μ_{ir} is the center of the hidden layer neuron closest to x_i in the sense of the Euclidean norm.

- (iii) Apply the novelty criterion to judge whether to add hidden layer neurons. If $d_i > \varepsilon_i$ and $\|e_i\| \left(\int_X \exp\left(-\frac{q\|x - x_i\|^2}{\kappa^2 \|x_i - \mu_{ir}\|^2}\right) p(x) dx \right)^{1/q} > \lambda$, then a new hidden layer neuron $K + 1$ is added to the network, and the corresponding parameters are set as

$$\begin{aligned} w_{K+1} &= e_i, \\ \mu_{K+1} &= x_i, \\ \sigma_{K+1} &= \kappa d_i. \end{aligned} \quad (31)$$

Otherwise, use the EKF method to update the node parameters of the hidden layer neuron closest to the current input in the network: w_{ir} , μ_{ir} , and σ_{ir} , and check whether the hidden layer neuron meets the output conditions; if $E_{\text{imp}}(ir) = \|w_{ir}\| \left(\int_X \exp\left(-\|x - \mu_{ir}\|^2 / \sigma_{ir}^2\right) p(x) dx \right)^{1/q} < \lambda$, delete the ir hidden layer neuron, and correspondingly, reduce the dimensionality of the EKF. When using too many hidden layer neuron nodes, the model will tend to overtrain the data, the generalization ability will be poor, and the classification effect will become less and less obvious. Therefore, the CIRAN learning algorithm introduces a hidden layer

TABLE 1: Shooting accuracy analysis.

| Group no. | 0-100 | 101-200 | 201-300 | 301-400 | 401-500 | 501-600 | 601-700 | 701-800 | 801-900 | 901-1000 |
|-----------------------------|-------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| Correct analysis | 99 | 98 | 95 | 96 | 100 | 99 | 97 | 92 | 100 | 99 |
| Unrecognized | 1 | 2 | 5 | 4 | 0 | 1 | 3 | 8 | 0 | 1 |
| Prediction success rate (%) | 99 | 98 | 95 | 96 | 100 | 99 | 97 | 92 | 100 | 99 |

neuron deletion strategy. Its network structure is simpler than that of the traditional one. The number of hidden layer neurons is less, and the running time required is less, which can better meet the real-time requirements.

3.6. Shooting Accuracy Prediction. Using the CIRAN online RBFNN-based shooting prediction proposed in this paper, from equation (1), the basketball should be hollow when entering the net or the basketball should be within a certain speed when it hits the inner basket, except for the conditions already obtained, when the ball is in the air, the ball cannot touch the basket. The basketball and the basket have custom sizes. Set the diameter of the basketball to d and the diameter of the basket to D , ignoring the air resistance. L is the shooting distance, h is the shooting height, and θ is the shooting angle; then, the shooting speed is defined as follows:

$$v = \sqrt{\frac{L^2 g^2}{2 \cos^2 \theta (L \tan \theta - h)}}. \quad (32)$$

Combining the distributed trajectory of ball shooting (13), if the following formula holds, then the shooting accuracy predicts success.

$$\lim_{x \rightarrow v} f(x_i) \mp p(x, t) \omega_j \sim E_{\text{imp}}(ir). \quad (33)$$

4. Experiments

In order to verify the effectiveness of the shooting prediction based on the CIRAN online RBFNN proposed in this paper, the shooting prediction and running time are performed on the sample number of basketball players' shooting motion images.

4.1. Model Evaluation. In the basketball player shooting motion assistance training system, choose 1000 groups of images as the test set and random variables $w_0 = 0.2$, $w_k = 5$, $\mu = 3$, and $TT = \sigma$, divided into 10 groups; the shooting prediction success rate is shown in Table 1.

As seen from Table 1, the number of unrecognized experiments in each group is relatively small, and the accuracy of the correctly analyzed shots is relatively large, reaching more than 95%. The effectiveness of the shot prediction based on the CIRAN online RBFNN can be seen.

4.2. Prediction Ratio. In order to determine the weight of equation (33), let ratio = ω_1/ω_2 , and determine the weights ω_1 and ω_2 by comparing the relationship between ratio and

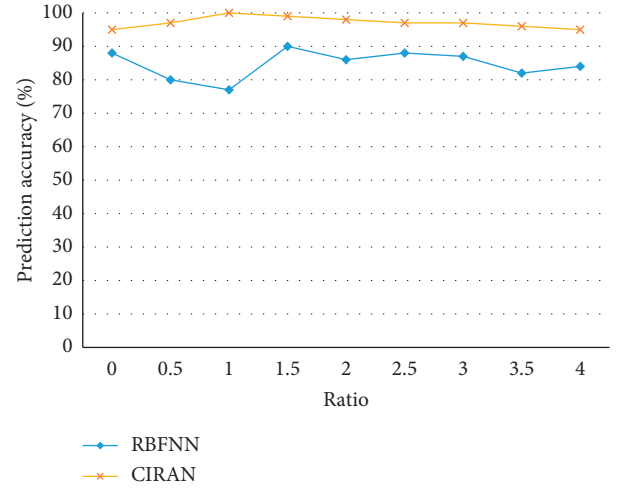


FIGURE 2: Prediction accuracy of the RBFNN and CIRAN.

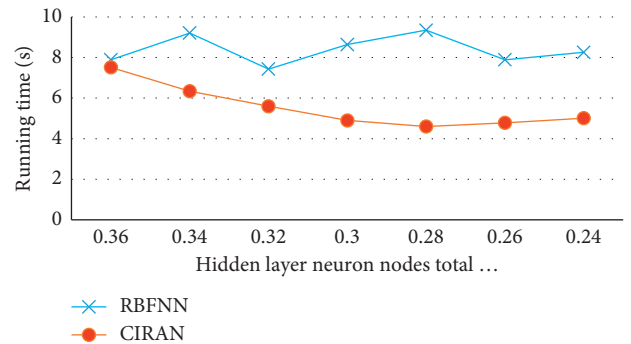


FIGURE 3: Running time of the RBFNN and CIRAN.

prediction accuracy under different ratios. As shown in Figure 2, when the ratio of the two is 0.98, CIRAN-based online RBFNN has a much better prediction accuracy than the traditional offline RBFNN. The shooting prediction accuracy of the method proposed in this paper is above 90%, while for the RBFNN method, it is only above 70%. The shooting accuracy prediction of the basketball player in this method is more accurate than that of the traditional method.

Through analysis, the accuracy of both the traditional RBFNN and the CIRAN-based online neural network for the prediction of shooting accuracy is above 70%. The prediction accuracy of the online neural network for shooting is higher than that of the traditional one. This is mainly because the online update function of the learning algorithm can better adjust the corresponding structure with the development of the game and has a better generalization ability.

4.3. Running Time Analysis. The CIRAN learning algorithm introduces a hidden layer neuron deletion strategy, which requires less running time and can better satisfy shooting accuracy prediction analysis. It can be seen from Figure 3 that when the ratio of hidden layer neuron nodes to the total number of nodes is 0.28, the running time is the shortest. In general, the CIRAN takes less time to predict the shooting accuracy than the RBFNN.

5. Conclusions

In today's highly information-based society, sports training is also quietly undergoing changes. In the arena of various sports, it is no longer purely a competition of athletes' sports skills and physical and psychological qualities but a comprehensive competition of the development of science and technology and the cohesion of people between countries. Introducing artificial intelligence and neural network into basketball shooting prediction can greatly improve the success rate of prediction. This paper proposes a shooting prediction based on the CIRAN online RBFNN and improves the RBFNN learning algorithm. The results show that the method proposed in this paper has a high accuracy rate in predicting the accuracy of basketball players' shooting and less running time. In the future, we will further analyze the behavior prediction of team members through the prediction of shooting accuracy to improve team cooperation; through the prediction of shooting, the trajectory of basketball players' shooting is analyzed, so as to improve the accuracy of basketball players' shooting.

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.

References

- [1] Q. Huang, W. Gao, H. Yao et al., "Event tactic analysis based on broadcast sports video," *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 49–67, 2009.
- [2] B. A. Jasani, M. Wu, S.-K. Lam, and P. K. Meher, "Threshold-guided design and optimization for Harris corner detector architecture," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 12, pp. 3516–3526, 2018.
- [3] S.-W. Wang, H.-W. Chen, "Volleyball players arm trajectory simulation research on optimization of recognition," *Computer Simulation*, vol. 34, no. 2, pp. 270–273, 2017.
- [4] B. Yeong-Hyeon, D. Kim, J. Lee, and K. C. Kwak, "Body and hand-object ROI-based behavior recognition using deep learning," *Sensors*, vol. 21, no. 5, pp. 1838–1860, 2021.
- [5] S. Alexandros and P. Ronald, "Learn to cycle: time-consistent feature discovery for action recognition," *Pattern Recognition Letters*, vol. 141, pp. 1–7, 2021.
- [6] Xiaohong Dong, "Shooting motion mathematical model of the optimal solution," *Technology Wind*, no. 28, pp. 43–44, 2018.
- [7] J. Landlinger, T. Stöggel, S. Lindinger, H. Wagner, and E. Müller, "Differences in ball speed and accuracy of tennis groundstrokes between elite and high-performance players," *European Journal of Sport Science*, vol. 12, no. 4, pp. 301–308, 2012.
- [8] H. Zhang, Z. Liu, H. Zhao, and G. Cheng, "Recognizing human activities by key frame in video sequences," *Journal of Software*, vol. 5, no. 8, pp. 818–825, 2010.
- [9] W. Wang and D. Zhou, "A multi-level approach to highly efficient recognition of Chinese spam short messages," *Frontiers of Computer Science*, vol. 12, no. 1, pp. 135–145, 2018.
- [10] H. Wu, J. Weng, X. Chen, and W. Lu, "Feedback weight convolutional neural network for gait recognition," *Journal of Visual Communication and Image Representation*, vol. 55, no. 8, pp. 424–432, 2018.
- [11] M. R. Malgireddy, J. J. Corso, S. Setlur, V. Govindaraju, and D. Mandalapu, "Framework for hand gesture recognition and spotting using sub-gesture modeling," in *Proceedings of the ICPR 2010, 20th International Conference*, pp. 3780–3783, Istanbul, Turkey, August 2010.
- [12] B.-Y. Su, J. Jiang, Q.-F. Tang, and M. Sheng, "Human dynamic action recognition based on functional data analysis," *Acta Automatica Sinica*, vol. 43, no. 5, pp. 866–876, 2017.
- [13] J. Zhong, H. W. Liu, and C. L. Lin, "Human action recognition based on hybrid features," *Applied Mechanics and Materials*, vol. 2594, no. 748, pp. 1188–1191, 2013.
- [14] E. Ramasso, C. Panagiotakis, D. Pellerin, and M. Rombaut, "Human action recognition in videos based on the Transferable Belief Model," *Pattern Analysis and Applications*, vol. 11, no. 1, pp. 1–19, 2008.
- [15] J. Kittler, B. Christmas, S. W. Lee, and M. C. Roh, "Gesture spotting for low-resolution sports video annotation, Pattern Recognition," *The Journal of the Pattern Recognition Society*, vol. 41, no. 3, pp. 1124–1137, 2008.
- [16] O. Camps, J. Min, and R. Kasturi, "Extraction and temporal segmentation of multiple motion trajectories in human motion," *Image and Vision Computing*, vol. 26, no. 12, pp. 1621–1635, 2008.
- [17] X. F. Tong, L. Y. Duan, and C. S. Xu, "Local motion analysis and its application in video based swimming style recognition," in *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 32–50, Hong Kong, China, August 2006.
- [18] X. Song, "Research on basketball shooting trajectory automatic capture method based on background difference method," *Automation and Instrumentation*, no. 7, pp. 42–45, 2020.
- [19] G. Liu, "3D vision based correction method for basketball shooting angle," *Modern Electronics Technique*, vol. 40, no. 5, pp. 45–48, 2017.
- [20] Y. Zhang, "Research on visual analysis based standardization judgment method for basketball shooting action," *Modern Electronics Technique*, vol. 40, no. 3, pp. 47–50, 2017.
- [21] D. Ma, Y. Chen, H. Chen, and L. Yun, "Augmented reality based basketball shoot trajectory analysis and simulation," *Computer Applications and Software*, vol. 31, no. 5, pp. 53–56, 2014.
- [22] H.-W. Xu, W.-G. Wan, B. Cui, J.-C. Lin, and K.-Y. Zhang, "Simulation of basketball shooting in virtual reality," *Journal of Applied Sciences*, vol. 27, no. 4, pp. 414–418, 2009.
- [23] D. Zhang and Y. Chen, "Realization of basketball training simulation system," *Journal of Graphics*, vol. 36, no. 5, pp. 789–794, 2015.

- [24] Y.-H. Chai, "Application of digital image processing technology to analysis of factors influencing the basketball shot rate," *Journal of Hefei University of Technology(Natural Science)*, vol. 30, no. 12, pp. 1607–1609, 2007.
- [25] B. Liu and W. Liu, "The lifting factorization of 2D 4-channel nonseparable wavelet transforms," *Information Sciences*, vol. 456, pp. 113–130, 2018.
- [26] X.-J. Wei, N.-Z. Li, X.-Z. Zhou, J. Ding, and W.-C. Ding, "Optimization of RBFNN based on dynamic multiple sub-population collaboration quantum-behaved particle swarm optimization algorithm," *Journal of Lanzhou Jiaotong University*, vol. 33, no. 3, pp. 98–103, 2014.
- [27] Z.-J. Liu, Z.-C. Zhu, H. Deng, and X. Liu, "Power allocation algorithm with respect to system capacity maximization based on RAN architecture of wireless access network," *Application Research of Computers*, vol. 30, no. 5, pp. 1486–1488, 2013.
- [28] Y. Hou and Y. Wang, "Research on attitude solution of four-rotor based on improved EKF," *Application of Electronic Technique*, vol. 43, no. 10, pp. 83–85, 2017.
- [29] Y. H. Cheng and C. S. Lin, "A learning algorithm for radial basis function networks: with capability of adding and pruning neurons," in *Proceedings of the 1994 IEEE International Conference on Neural Networks*, pp. 797–801, Orlando, FL, USA, June 1994.
- [30] B. Li and X.-P. Lai, "An improved GGAP-RBF algorithm and its application to function approximation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 2, pp. 230–235, 2007.
- [31] Q. Bao and W. Song, "GAP-RBF neural network learning algorithm based ON population partitioning optimisation," *Computer Applications and Software*, vol. 33, no. 11, pp. 215–220, 2016.

Research Article

Analysis of Physical Expansion Training Based on Edge Computing and Artificial Intelligence

Zhongle Liu 

Jilin Agricultural Science and Technology University, Jilin 132101, China

Correspondence should be addressed to Zhongle Liu; liuzhongle@jlnku.edu.cn

Received 27 April 2021; Revised 20 May 2021; Accepted 25 May 2021; Published 2 June 2021

Academic Editor: Jianhui Lv

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

The effective development of physical expansion training benefits from the rapid development of computer technology, especially the integration of Edge Computing (EC) and Artificial Intelligence (AI) technology. Physical expansion training is mainly based on the collective form, and how to improve the quality of training to achieve results has become the content of everyone's attention. As a representative technology in the field of AI, deep learning and EC evolving from traditional cloud computing technology are all well applied to physical expansion training. Traditional EC methods have problems such as high computing cost and long computing time. In this paper, deep learning technology is introduced to optimize EC methods. The EC cycle is set through the Internet of Things (IoT) topology to obtain the data upload speed. The CNN (Convolutional Neural Network) model introduces deep reinforcement learning technology, implements convolution calculations, and completes the resource allocation of EC for each trainer's wearable sensor device, which realizes the optimization of EC based on deep reinforcement learning. The experiment results show that the proposed method can effectively control the server's occupancy time, the energy cost of the edge server, and the computing cost. The proposed method in this paper can also improve the resource allocation ability of EC, ensure the uniform speed of the computing process, and improve the efficiency of EC.

1. Introduction

Physical training generally refers to all physical activities that maintain and develop proper physical expansion and improve physical health through exercise. Regular physical training can activate the body's immune system and prevent or improve some civilization diseases, such as cardiovascular disease, type 2 diabetes, and obesity. It can also improve mental health, reduce depression, increase resistance ability to stress, improve sleep quality, improve insomnia problems, and help form positive self-esteem. Regular exercise is one of the keys to maintaining health, and it has a significant contribution to maintaining a healthy weight, digestive system, bone density, muscle capacity, free movement of joints, physiological function, reducing the chance of facing surgery in the future, and strengthening the immune system.

Physical fitness is the human body's ability to overcome resistance, rapid movement ability, continuous work (exercise) ability, coordinated movement ability, and sensitive

and accurate movement ability that the human body shows in exercise, labor, and life [1]. It can be considered that physical fitness not only reflects the basic functional capabilities of human activities but also reflects the basic functional capabilities of human labor and life. Physical training is an indispensable basic athletic ability for work and life. It is conducive to mastering complex technical movements and improving exercise effectiveness, withstand heavy load training and high-intensity exercise, and maintain a stable and good mentality in daily training and competition status. The physical fitness training is a team style physical training with physical fitness as the guide, games as a tool, and mental fitness as the main purpose. Physical expansion training is rarely completed by individuals, usually in a collective form. The relationship within the group is directly related to the actual training benefits. In addition to the conventional training methods, the content of physical development training should also include some other contents, such as handstands and walking backward.

Physical expansion training is mainly based on the collective form, and how to improve the quality of training has become the content of everyone's attention. In recent years, with the rapid development of social economy and science and technology on a global scale, many emerging technologies have continuously emerged in information and communication technology industry [2, 3]. Among them, two representative technologies are widely regarded as having a huge impetus and far-reaching influence on the human economy and society. First, as a representative technology in the field of AI, deep learning has benefited from advances in algorithms and data sets [4]. It has been developed by leaps and bounds in recent years and is used in unmanned driving, e-commerce, smart home, and smart finance. The field has played a big role, profoundly changed people's lifestyle, and improved production efficiency. The other technology is EC evolved from traditional cloud computing technology. Compared with cloud computing, EC sinks strong computing resources and efficient services to the edge of the network, thereby having lower latency, lower bandwidth usage, higher energy efficiency, and better privacy protection. The introduction of EC and AI into physical expansion training can better help people train [5].

The rapid development of the IoT has brought us into the postcloud era, which will generate a lot of data in our daily lives [6, 7]. IoT applications may require extremely fast response time, data privacy, and so on. If the data generated by the IoT is transmitted to the cloud computing center, the network load will be increased, the network may cause congestion, and there will be a certain data processing delay. With the increase of mobile devices and the increase of camera deployment in cities, the use of video to achieve a certain purpose has become a suitable means, but the cloud computing model is no longer suitable for this kind of video processing, because a large amount of data in the network transmission in the video may cause network congestion, and the privacy of video data is difficult to guarantee. Therefore, EC is proposed to allow the cloud center to delegate related requests. Each edge node processes the request combined with local video data and then only returns the relevant results to the cloud center. This not only reduces network traffic but also guarantees privacy to users to some extent [8]. EC refers to processing and analyzing data at the edge of the network, which can reduce request response time, improve battery life, reduce network bandwidth, and ensure data security and privacy. An edge node is any node with computing resources and network resources between the source of data generation and the cloud center. For example, smart wearable devices are the edge nodes between people and cloud centers. In an ideal environment, EC refers to analyzing and processing data near the source of data generation, without data circulation, thereby reducing network traffic and response time. In order to quickly update the training model and improve efficiency.

Physical expansion training is completed through groups; everyone in the group is equipped with a wearable sensor. The wearable sensor can analyze the exercise quality of each trainer. Two epoch-making new technologies, AI and EC, are currently facing bottlenecks in their further

development. On the one hand, for deep learning technology, because it requires high-density calculations, current intelligent algorithms based on deep learning usually run in cloud computing data centers with powerful computing capabilities. With the high popularity of mobile terminal devices consideration, how to effectively deploy deep learning models in resource-constrained terminal devices has attracted great attention from academia [9, 10]. It has aroused great attention from academia and industry [9, 10]. On the other hand, with the sinking and decentralization of computing resources and services, EC nodes will be widely deployed at network edge access points (such as cellular base stations, gateways, and wireless access points). The high-density deployment of EC nodes also brings new challenges to the deployment of computing services; users usually have mobility; therefore, when the user moves between different nodes coverage frequently, whether computing services should be with the trajectory of the mobile user migration, this is a dilemma problem, because although service migration can reduce delay and improve with experience, it will bring additional costs such as bandwidth and energy consumption.

The development bottlenecks faced by AI and EC can be alleviated through synergy. On the one hand, for deep learning, mobile devices running deep learning applications offload part of model inference tasks to adjacent EC nodes for calculations, thereby cooperating with terminal devices and edge servers to integrate the local computing capabilities and strong computing capabilities of the two complementary advantages. In this way, because a large number of calculations are executed on EC nodes with strong computing power adjacent to the mobile device, the resource and energy consumption of the mobile device itself and the delay of task inference can be significantly reduced, thereby ensuring good user experience. On the other hand, in view of the dynamic migration and placement of EC services, AI technology is also promising. Specifically, based on high-dimensional historical data, AI technology can automatically extract the mapping relationship between the optimal migration decision and high-dimensional input, so that when a new user location is given, the corresponding machine learning model can quickly map it to the optimal migration decision. In addition, based on the user's historical trajectory data, AI technology can also efficiently predict the user's movement trajectory in the short term in the future, thereby realizing predictive edge service migration decisions and further improving the service performance of the system. In general, EC and AI will generate a new paradigm of "edge intelligence," which will generate a large number of innovative research opportunities.

Starting from the dimension of EC combined with AI, the main contribution of the paper is to introduce deep reinforcement learning technology to EC and propose a method for EC to drive real-time deep reinforcement learning. The rest of the paper is organized as follows. Section 2 analyzes and summarizes domestic and foreign research work in physical expansion training using EC and AI. Section 3 proposes that EC drives real-time deep

reinforcement learning methods. The experimental results are reported in Section 4, and finally, Section 5 concludes this paper.

2. Related Work

AI has improved people's quality of life and living standards. At the same time, with the rapid development of the mobile Internet and the IoT industry, as a novel combination of AI application and EC, intelligent edge system is promising by researchers in the field of AI and network computing. The research on intelligent edge system based on EC architecture is becoming more and more important.

Traditional EC research mostly considered the problem of task offloading, that is, whether the tasks generated by the end device should be handed over to the network edge device for calculation and processing. There has been a lot of research in this area. In the research direction of task offloading, a lot of work focuses on the energy consumption optimization problem of mobile devices [11]. In [12, 13], considering the task offloading problem in energy harvesting systems, the study in [12] proposed an effective based on the resource management algorithm of reinforcement learning; the algorithm obtains the optimal strategy of dynamic offloading through online learning. The study in [13] proposed a low-complexity online algorithm based on the Lyapunov optimized dynamic computing offloading algorithm, which can make decisions for task offloading only by relying on the current system state. In [14, 15], the authors studied the service caching mechanism in EC systems, and some fault-tolerant mechanisms for EC systems were also being studied [16–18]. The study in [19] proposed a series of methodologies, using association rule mining technology to analyze the physical fitness index of basketball players, using data processing and database management functions, which can also solve the management of athletes' physical ability indicators and assist coaches in managing players and calculating training results to improve the efficiency of data processing. According to the data mining technology analysis of the player's physical training data, from the aspect of competitive sports, the goal of training was to create excellent sports performance, as well as the most basic competitiveness and the most controllable factors for the improvement of the player's physical ability. Physical training was the basic way for coaches to know the physical fitness of players. The coaches regularly test the physical performance of the players. According to different test calculation standards, they calculate the results of the physical fitness test for each item of each player. Later, based on their own experience, they would evaluate the physical fitness of the players to formulate a reasonable effective training plan to train. But with the accumulation of test data, it would become more and more difficult to analyze this pile of data with manual management work, and the commonly used computer data processing and database management functions can solve the management of the player's physical test data. It cannot find potential knowledge outside the database, and it cannot provide effective evaluation and prespeculation of the player's physical condition.

In terms of the design of an intelligent physical training platform, there was a ZigBee-based physical training platform designed by Huo et al. of Shenyang Ligong University. The platform achieved the functions of data collection and transmission at the same time [20]. With the rapid popularity of mobile devices, many wearable sports data collection products for ordinary sports enthusiasts have appeared on the market. For example, Huawei smart bracelets, Nike+, and Adidas miCoach are powerful, but they are only for professionals. The physical team or a certain kind of training program, and these products required expensive auxiliary equipment, which was not suitable for physical expansion training. In terms of training method recognition, the study in [21] proposed a hidden condition random field object recognition model based on the maximum boundary value and combined a large number of global and local features to distinguish different actions. In the training process, people mainly pay attention to the impact of training on physical functions. Different amounts of training have different effects on physical functions [22]. When the energy consumption is small, the body's metabolism is higher; when the energy consumption is large or even too large, the human body's metabolism is large. Although it has reached the training volume, it will also cause excessive energy consumption and produce some harmful effects. Metabolic wastes, in severe cases, can even cause shock or crushing death, which has harmful effects on body functions. It can be seen that only when the energy consumption during physical training is controlled within a reasonable range, can it have a positive impact on the changes in human body function.

3. EC Drives Real-Time Deep Reinforcement Learning Method

In the physical development training, each trainer is equipped with wearable sensors. These devices generate a large amount of data, which makes the traditional computing framework and the amount of data incompatible. At the same time, the trainer cannot get real-time feedback on the network transmission speed delay. Only real-time feedback can help the development of physical expansion training, so as to establish an effective training model for each trainer and adjust the training method in time. The delay makes it impossible to realize the cloud transmission of IoT data. A large amount of data is directly consumed at the edge of the network. Therefore, it is necessary to carry out calculations on the edge of the IoT. In physical development training, each trainer is an edge node.

As one of the mainstream technologies in the field of AI, deep learning has been strongly sought after by academia and industry in recent years [23, 24]. Since deep learning models require a lot of calculations, intelligent algorithms based on deep learning usually exist in cloud computing data centers with powerful computing capabilities. With the rapid development and popularization of mobile terminals and IoT devices, how to break through the resource limitations of terminal devices so as to efficiently run deep learning models on resource-constrained terminal devices has attracted a lot of attention. To solve this problem,

consider the idea of EC to empower AI and use the characteristics of near real-time computing of EC to reduce the delay and energy consumption of deep learning model inference. For this reason, in this research, deep reinforcement learning is integrated into EC to reduce computing costs. Deep reinforcement learning combines the perception ability in deep learning with the decision-making ability in reinforcement learning and optimizes the original EC through AI to ensure the validity and feasibility of the optimized results.

EC is a technology which deploys computing tasks between the cloud and the terminal. The characteristics of EC and its proximity to the device are destined to have the advantages of real-time processing, so it can better support the real-time processing and execution of local services. EC directly filters and analyzes the data of the terminal equipment, which saves energy and saves time and efficiency. In other words, some terminals will offload the computing task to the EC device and perform the computing task through the resources allocated by the edge device. In the cloud computing model, terminal equipment is the data-consuming role. Data producers (such as YouTube) publish data to the cloud, and data consumers (such as mobile phones) request data to obtain cloud data. This is a traditional cloud computing model, but with the popularity of IoT devices, the physical expansion training in this paper, terminal trainers use wearable sensors to generate a large amount of data, and the processing and transmission of these data will encounter some problems. EC is committed to solving these problems, so EC is integrated with the IoT technology, the amount of data is very large, it needs to take up a large upload bandwidth, so the data needs to be processed on the device side and processed into a suitable format for transmission, and the computing task is processed on the terminal device. Therefore, the terminal is no longer pure data consumption but plays a role in data generation. In summary, EC uses the processing power of the LAN gateway to process more real-time information.

3.1. EC Cycle Setting. The EC cycle needs to analyze the topology of the IoT and set the EC cycle of the IoT based on the analysis results. Under normal circumstances, the topology of the IoT can be divided into two parts: a flat network structure and a hierarchical structure. Aiming at the relevant characteristics of the IoT, in this paper, the hierarchical results are used as the research object.

According to the trainer's wearable device business data and the randomness of renewable energy, the continuous time scale is set to divide the time at equal intervals, and the time interval is set to Δt , which is the calculation decision period at equal intervals. Using the above period, the decision time and the decision period Δt can be dynamically adjusted to meet the complexity and variability of EC in the IoT. At each calculation decision point, the generation rate of business data is set to $bg_0(t)$ and $\eta(t)$; then, the accumulated data size and energy value of the IoT in the calculation period of equal time interval can be expressed as

$$bg(t) = bg_0(t)\Delta t, \quad (1)$$

$$E(t) = x(t)\Delta t. \quad (2)$$

Among them, $bg(t)$ is defined as the amount of data accumulated by the trainer's wearable device during the calculation period, and $E(t)$ represents the accumulated energy value of the trainer's wearable device during the calculation period. Through the above formula, the data generated in the edge calculation cycle is controlled. In the case of different calculation cycles, in order to facilitate the development of calculations, the generation rate of the set business data and the energy attainment rate are both reflected in the form of independent and identical distribution. Set the bandwidth of the wireless broadband in the IoT of the trainer's wearable device as B , and there is only one base station in an IoT; ignoring the interference of the base station, the data upload rate U_i of the IoT can be expressed as

$$U_i = \frac{Bp_i^l}{E(t)} + \frac{bg(t)f_i^l}{\chi^2} \log \text{ber}_{\text{up}}. \quad (3)$$

Among them, p_i^l represents the computing power of the wearable device of each trainer, f_i^l represents the local computing power of the wearable device, χ^2 is the variance of Gaussian white noise, and ber_{up} is the target bit error rate. Using the above parts, the design part of the EC cycle of the IoT is completed, and this result is used as the data basis for constructing the EC execution process.

3.2. EC Execution Process. Using the above analysis results, as the basis for the construction of the EC execution process of each trainer's wearable device, deep reinforcement learning technology is used to complete the EC execution process.

For the local computing part $(1 - x_i)q_i$ of each trainer's wearable device, define t_i as the local execution delay; this part contains the processing time of the server; set cf_i^t as the CPU frequency for the calculation; then, the execution delay de_i^t can be expressed as

$$de_i^t = \frac{(1 - x_i)q_i l_i}{cf_i^t} + \log \left(1 + \frac{|cg_i|}{\Gamma(\text{ber}_{\text{up}})} \right). \quad (4)$$

Among them, l_i is the length of the communication channel. $\Gamma(\text{ber}_{\text{up}})$ represents the margin of signal-to-noise ratio introduced to meet the uplink target error rate. Set EC_i^t as the energy consumption during local execution. According to formula (4), it can be expressed as

$$EC_i^t = \rho_i cf_i^t (1 - x_i) q_i l_i N_0. \quad (5)$$

Among them, ρ_i TT is set to the energy density in the IoT, which represents the energy consumed in the decision-making cycle during the calculation process, and N_0 represents the noise power of the channel. Considering that the change of cf_i^t will affect the change of calculation energy consumption, the dynamic voltage scaling technology is used to set the overall calculation time ω_i^t , and the local

calculation frequency cf_i^t is reasonably allocated. The sub-part can be expressed as

$$cf_i^t = \min \left\{ \frac{(1-x_i)q_i^t l_i}{\omega_i^t}, cf_i^{t \max} \right\}. \quad (6)$$

Use the above formula to control the execution process of EC. Ensure that the energy consumption of EC matches the characteristics of each trainer's wearable device. According to the data transmission speed calculated in formula (3), the delay time of the calculation process can be calculated as follows:

$$t_i^0 = \frac{q_i}{\omega_i}. \quad (7)$$

Using the above formula to control the time during the execution of EC, according to the EC execution process designed in this part, in-depth reinforcement learning technology is integrated into the EC resource allocation of each trainer's wearable device.

3.3. EC Resource Allocation. In this paper, the CNN model in deep reinforcement learning technology is used as the design basis to realize the EC resource allocation of the IoT, and the convolution processing is mainly used to complete the rational allocation of resources [25, 26].

According to the relevant knowledge of the signal and the network, the convolution operation of the two signals in the decision time period can be embodied in the form of a formula as follows:

$$\text{conv}(t) = s(t) * w(t). \quad (8)$$

Among them, $s(t)$ and $w(t)$, respectively, represent the signals in edge calculation. The discrete sequences $f(n)$ and $g(n)$ in resource allocation are obtained through the translation, multiplication, and integration results of these two signals in the time period, and the convolution results are embodied in discrete forms.

$$\text{conv}(n) = f(n) * g(n) = \sum_{m=0}^{N-1} f(n)g(n-m). \quad (9)$$

Using the above formula, set the number of connections in the EC of each trainer's wearable device. According to the parameter characteristics of the CNN model, set the weight and bias in the allocation process as the number of connections; then, the number of parameters in the edge calculation can be expressed as $ct * wi_k * he_k + 1$, where ct is the number of channels in the calculation and wi_k and he_k are the width and height of the convolution kernel. Through this formula, the calculation amount of the calculation process can be obtained as $ct * wi_k * he_k$, and the rationalized distribution of the calculation amount can be expressed as

$$\rho_i(t) = \frac{ct * wi_k * he_k}{\text{conv}(n)}. \quad (10)$$

Using the above formula, complete EC resource allocation. Connect this part with the design part above in an orderly

manner to realize the application of deep reinforcement learning in EC of the IoT. So far, the design of real-time deep reinforcement learning method driven by EC is completed.

4. Experiment Design and Result Analysis

The experiment evaluates the application effect of the previously designed deep reinforcement learning in EC through real data sets. In addition, this paper also compares the proposed method with three methods in server occupancy time, server power consumption, and calculate waiting time.

4.1. Data Set Design. Python 3.8 is used to implement the calculation process of this experiment. In order to make the experimental results more convincing, Google Cluster is used as the data set, and the task samples are constructed using attributes such as CPU request and memory request. In the simulation scenario of this paper, suppose there are 10 edge nodes, that is, 10 trainers, and each trainer is equipped with wearable sensors. The bandwidth, computing power, and computing power per unit time of the edge node (each trainer) can be seen from Table 1 for consumption. At the same time, suppose that the generation rate of business data is 0.36 and 0.73 for $bg_0(t)$ and $\eta(t)$, respectively, the CPU frequency cf_i^t is 4 GHz, $\Gamma(\text{ber}_{\text{up}}) \leq 6$ dB, the local computing power f_i^t of the wearable sensor device equipped by the trainer is 30 Mb/s, and the bit error rate ber_{up} is 0.48.

The above-mentioned setting part is used as the preparation stage of the experiment, using the above-mentioned setting results to complete this experiment and comparing the difference between the calculation method after using the deepening intensity learning and the method before using this technology.

4.2. Contrast Indicators. The content of the experiment is set as the performance comparison of the calculation method, and the calculation cost is the focus of the experiment. The so-called computing cost is composed of edge server occupancy time, edge server energy cost, and average computing waiting time. In the experiment, in order to increase the effectiveness, the computing environment is set to both the full work of the edge server and the part of the edge server to verify the applicability of each calculation method. In the course of the experiment, the calculation performance of each algorithm is studied in the form of a uniform increase in the number of tasks, and the specific results are embodied in the form of images.

4.3. Analysis of Experimental Results

4.3.1. Server Occupancy Time. In order to verify the server occupancy time of different methods in EC, compare the server occupancy time after the optimization method of deep reinforcement learning, the improved cat group algorithm, and the edge-cloud collaborative IoT optimization method, and the results are shown in Figure 1.

According to Figure 1, it can be seen that the edge server takes different time under different methods. In Figure 1(a),

TABLE 1: Bandwidth, computing power, and computing energy consumption of edge nodes.

| No. | Bandwidth (MHz) | Calculated ability (Mb/s) | Calculated energy consumption per unit time (J) |
|-----|-----------------|---------------------------|---|
| 1 | 100 | 150 | 0.002 |
| 2 | 100 | 200 | 0.003 |
| 3 | 100 | 100 | 0.001 |
| 4 | 150 | 150 | 0.002 |
| 5 | 150 | 200 | 0.003 |
| 6 | 150 | 100 | 0.001 |
| 7 | 150 | 150 | 0.002 |
| 8 | 200 | 150 | 0.002 |
| 9 | 200 | 200 | 0.003 |
| 10 | 200 | 100 | 0.001 |

when the number of tasks is 20, the occupancy time of the edge server without optimization is 44.8 s, and the occupancy time of the edge server of the edge-cloud cooperative optimization method is 26.7 s. Improved cat group algorithm edge server takes 23.5 s, and the server occupancy time of the deep reinforcement learning optimization method is 7.5 s. Although the three methods can effectively reduce the server occupancy time, the occupancy time of the method in this paper is significantly lower than that of other methods, which shows that the trainer is equipped with wearable sensor equipment to occupy less server time.

In Figure 1(b), in some working environments, when the number of tasks is 30, the edge server that is not optimized takes 38.6 s, and the edge server of the edge-cloud cooperative optimization method takes 24.7 s. Improved cat group algorithm edge server takes 18.2 s, and the server occupancy time of the deep reinforcement learning optimization method is 2.7 s. When the number of tasks is 70, the unoptimized edge server takes 49.8 s, the edge-cloud cooperative optimization method takes 42.5 s, and the improved cat group algorithm IoT optimization method takes 33.5 s. The server occupancy time of the deep reinforcement learning optimization method is 3 s. In some working environments, the server occupancy time of the method in this paper is also significantly lower than other methods, which shows that the method in this paper has higher computational efficiency and strong applicability.

With the continuous increase of the number of tasks, in two different edge server working states, the algorithm using deep reinforcement learning technology can ensure the normal operation of the server. The use of deep reinforcement learning technology can effectively control the server's occupancy time so that each trainer's wearable device has more time to process local business and establish an optimized training model to better achieve the purpose of physical expansion.

4.3.2. Server Power Consumption. Based on the above, the server power consumption of the above three methods is

obtained through statistics, and the results are shown in Table 2.

Analysis of Table 2 shows that the server energy costs vary from different methods. The energy cost of the method in this paper is significantly lower than the other two methods. The use of deep reinforcement learning technology in the calculation method can effectively control the energy cost of the edge server in the calculation process, so as to ensure the calculation cost and minimize the energy consumption of the wearable sensor device.

4.3.3. Calculation Waiting Time. In order to further verify the calculation efficiency of different methods, the average calculation waiting time experiment under different tasks is added, and the results are shown in Figure 2.

Analyzing Figure 2 shows that under different number of tasks, the calculation waiting time is different. When the number of tasks is 5, the average calculation waiting time of the deep reinforcement learning method is 1 ms, the average calculation waiting time of the improved cat group algorithm is 3 ms, and the average calculation waiting time of the edge-cloud cooperation method is 3.2 ms. When the number of tasks is 40, the average calculation waiting time of the deep reinforcement learning method is 1.25 ms, the average calculation waiting time of the improved cat group algorithm is 4 ms, and the average calculation waiting time of the edge-cloud cooperation method is 3.9 ms. The method in this paper can improve the resource allocation ability of EC, ensure the uniform speed of the computing process, and improve the efficiency of EC.

Integrating the results of the average calculation waiting time, the energy cost results of the edge server and the edge server occupancy time results show that the EC method based on deep reinforcement learning designed in the paper can effectively control the computing cost and complete efficient EC during performance. In the process, the model for the trainer can be applied to the training faster, so that the physical expansion training can be carried out more effectively.

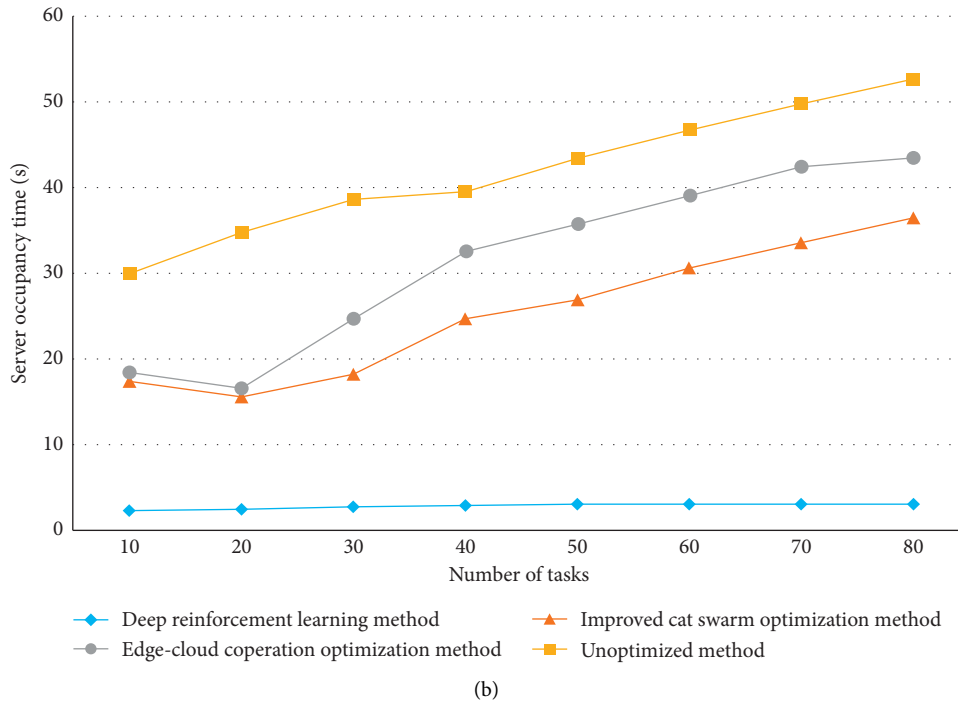
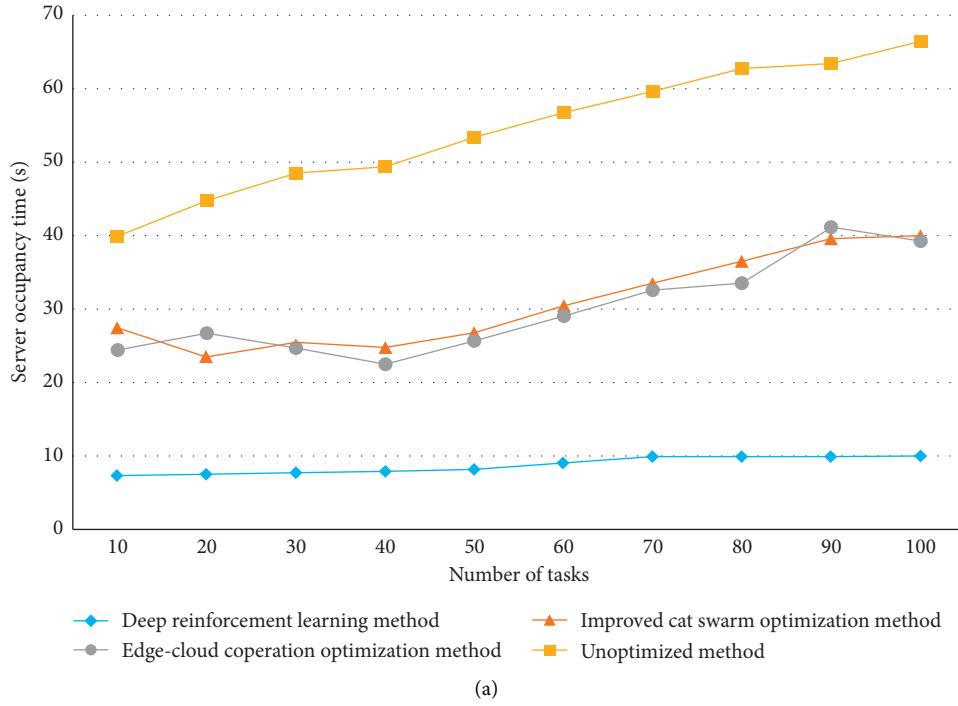


FIGURE 1: Edge server occupancy time. (a) Edge server occupancy time with all edge nodes working. (b) Edge server occupancy time with partial edge nodes working.

TABLE 2: Server power consumption by different methods (W).

| Operation hours (H) | Deep reinforcement learning | Improved cat swarm | Edge-cloud cooperation |
|---------------------|-----------------------------|--------------------|------------------------|
| 0.5 | 4 | 24 | 23 |
| 1 | 6 | 41 | 38 |
| 1.5 | 8 | 60 | 57 |
| 2 | 9 | 80 | 78 |

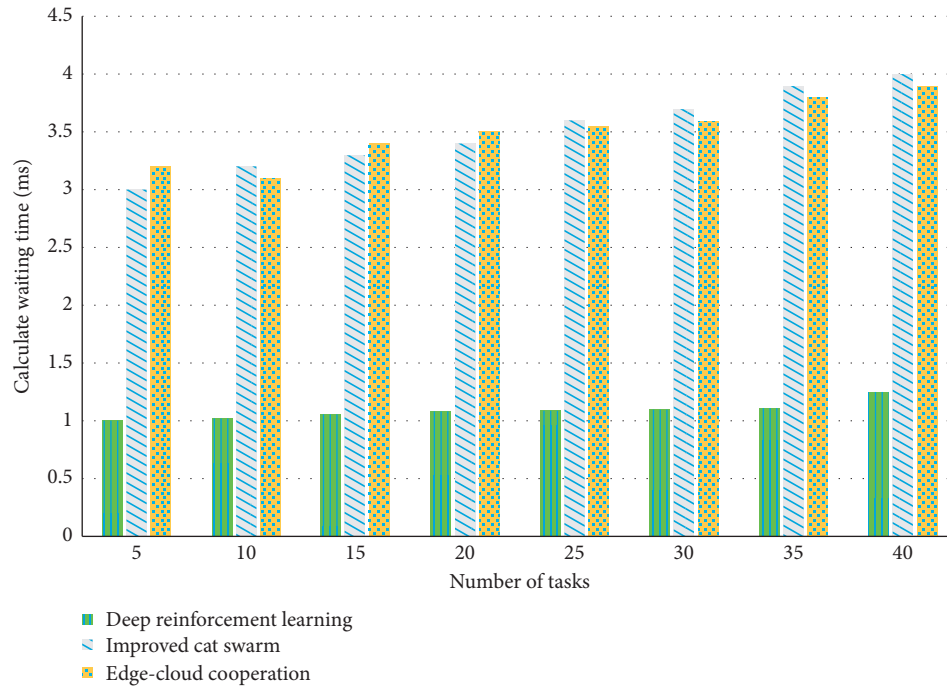


FIGURE 2: Average calculation waiting time under the same method.

5. Conclusions

Physical expansion training has always been a topic of discussion. As an important part of the field of AI, deep learning technology has been strongly sought after by academia and industry in recent years, and the emergence of EC technology corresponding to cloud computing has once again attracted great attention from the academic community. In this paper, the CNN model in deep reinforcement learning technology is used to realize the EC resource allocation of the IoT, and the convolution processing is mainly used to complete the rational allocation of resources. This paper combines AI and EC to propose a method for EC to drive real-time deep reinforcement learning. This method proposed in this paper is used on the wearable sensor device of each trainer in physical expansion training. Through experimental analysis, the method proposed in this paper has low occupancy time, high computing efficiency, and strong applicability; server power consumption is small; and it can effectively control the calculation cost, complete the efficient EC process so that each trainer's training model can be applied to training faster, and improve the quality and accuracy of the training, so as to more effectively carry out physical expansion training. In future research, further optimizations will be made on how to more effectively complete EC, while ensuring the reliability and timeliness of task processing and balancing the load of each edge server during peak periods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Q. Yan, T. Liao, and Y.-j. Zhang, "Digital-based performance training: concepts, advances and applications," *China Sport Science*, vol. 38, no. 11, pp. 3–16, 2018.
- [2] Y. Cao and L. Zheng, "The values, difficulties and countermeasures of artificial intelligence applied in sports field," *Sports Culture Guide*, no. 11, pp. 31–35, 2018.
- [3] J. Ge, "Research on the application of artificial intelligence and big data in sports," *Sport Science And Technology*, vol. 41, no. 2, pp. 30–31, 2020.
- [4] W. Ma and M. A. Jing, "Application and prospect of artificial intelligence in physical education," *Journal of Sports Adult Education*, vol. 36, no. 6, pp. 42–45, 2020.
- [5] L. Lu and X. Li, "Clustering and evolution of artificial intelligence technology in the field of international sports," *Journal of Shandong Institute of Physical Education and Sports*, vol. 36, no. 3, pp. 21–32, 2020.
- [6] Q. Liu, "Network big data technology exploration based on cloud computing and IoT," *Modern Industrial Economy and Informationization*, vol. 11, no. 2, pp. 101–102, 2021.
- [7] Y. Chen, "Research on IoT technology based on big data," *Wuxian Hulian Keji*, vol. 16, no. 13, pp. 27–28, 2019.
- [8] H. Gao and X. h. Li, "Research on cloud edge task distribution strategy in EC," *Information Technology*, no. 2, pp. 103–108, 2021.
- [9] F. N. Iandola, S. Han, M. W. Moskewicz et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, <https://arxiv.org/abs/1602.07360>.
- [10] J. X. Wu, L. Cong, Y. H. Wang et al., "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern*

- Recognition (CVPR) 2016*, pp. 4820–4828, IEEE Press, Las Vegas, NV, USA, June 2016.
- [11] K. Kumar and Y. H. Lu, “Cloud computing for mobile users: can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, 2010.
 - [12] J. Xu, L. Chen, and S. Ren, “Online learning for offloading and autoscaling in energy harvesting mobile edge computing,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.
 - [13] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
 - [14] J. Xu, L. Chen, and P. Zhou, “Joint service caching and task offloading for mobile EC in dense networks,” in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 207–215, IEEE, Honolulu, HI, USA, April 2018.
 - [15] S. Wang, K. Chan, I.-H. Hou et al., “Red/LeD: an asymptotically optimal and scalable online algorithm for service caching at the edge,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1857–1870, 2018.
 - [16] A. Javed, K. Heljanko, A. Buda et al., “Cefiot: a fault-tolerant IoT architecture for edge and cloud,” in *Proceedings of the 2018 IEEE 4th World Forum on IoT (WF-IoT)*, pp. 813–818, IEEE, Singapore, February 2018.
 - [17] J. Dean, G. Corrado, R. Monga et al., “Large scale distributed deep networks,” *Advances in Neural Information Processing Systems*, pp. 1223–1231, 2012.
 - [18] W. Zhang, S. Gupta, X. Lian et al., “Staleness-aware async-sgd for distributed deep learning,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2350–2356, AAAI Press, New York, NY, USA, July 2016.
 - [19] M. Dai and Y. Huang, “Analysis of physical ability data based on data mining,” *Computer Engineering and Applications*, vol. 39, no. 9, pp. 38–40, 2003.
 - [20] S. Huo, B. Li, Z. Zhao, J. Li, and W. Qu, “Motion data acquisition and transmission based on ZigBee module,” *Microcontrollers & Embedded Systems*, no. 12, pp. 41–42, 2009.
 - [21] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, “Discriminative latent models for recognizing contextual group Activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1549–1562, 2012.
 - [22] T. Wang, Y. Zhang, and J. Wang, “Characteristics of time variation in energy metabolism during exercise with moderate intensity,” *Journal of Zhejiang Normal University (Natural Sciences)*, no. 4, pp. 470–474, 2014.
 - [23] Y. Xiao, “Sports video classification method based on deep learning,” *Electronic Design Engineering*, vol. 29, no. 3, pp. 162–166, 2021.
 - [24] Z. Li, “Research on sports video classification based on deep learning and transfer learning,” *Electronic Measurement Technology*, vol. 43, no. 18, pp. 21–25, 2020.
 - [25] L. Kong, R. Wang, N. Zhang, and H. Li, “Survey on AI detection and recognition algorithms based on EC,” *Radio Communications Technology*, vol. 45, no. 5, pp. 453–462, 2019.
 - [26] R. Cai, C. Zhong, Y. Yu, B. Chen, Y. Lu, and Y. Chen, “CNN quantization and compression strategy for EC applications,” *Journal of Computer Applications*, vol. 38, no. 9, pp. 2449–2454, 2018.

Retraction

Retracted: Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing

Mobile Information Systems

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

Copyright © 2023 Mobile Information Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] F. Xu and W. Chu, "Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing," *Mobile Information Systems*, vol. 2021, Article ID 3534577, 8 pages, 2021.

Research Article

Sports Dance Movement Assessment Method Using Augment Reality and Mobile Edge Computing

Fang Xu  and Wentao Chu

Anhui University of Arts, Hefei 230011, China

Correspondence should be addressed to Fang Xu; 114007@ahua.edu.cn

Received 19 April 2021; Revised 10 May 2021; Accepted 17 May 2021; Published 25 May 2021

Academic Editor: Jianhui Lv

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

AR (Augment Reality) is an emerging technology that combines computer technology and simulation technology. It uses a computer to generate a simulation environment to immerse users in the environment. AR can simulate the real environment or things and present it to users by virtue of its multiperceptual, interactive, immersive, and other characteristics, to achieve an immersive effect. For sports dance, the same can be used to enhance the effect of teaching and learning through the use of AR technology. Aiming at the problems of delay and terminal equipment energy consumption caused by high-speed data transmission and calculation of virtual technology, this paper proposes a sports dance movement transmission scheme that uses equal power distribution on the uplink. Firstly, based on the collaborative attributes of the AR sports dance business, a system model for AR characteristics is established; secondly, the system frame structure is analyzed in detail, and the constraint conditions are established to minimize the total energy consumption of the system; finally, the mathematical model of Mobile Edge Computing (MEC) based on convex optimization is established under the condition that the delay and power consumption meet the constraints, so as to obtain the optimal communication and computing resource allocation scheme of sports dance in AR. The experimental results reveal that the proposed sports dance movement assessment method based on AR and MEC is efficient.

1. Introduction

Dance is a highly practical subject, and most of the knowledge and skill information in the training process requires learners to obtain by imitating and feeling the teacher's movements. Practical dance training can hardly be the same as other training methods. Modern training methods such as online teaching and remote teaching are adopted. Because in these training methods, learning is only by watching movement teaching videos, it is difficult to get an immersive and expressive feeling of dance movement performance, and it is impossible to evaluate and feedback the effect of movement learning in time. At the same time, traditional classroom teaching methods are limited by time and space after all, and students' review of learning content after class will also be affected. The application of AR provides the possibility of solving these problems to a large extent. As a new human-computer interaction method in the

computer field, AR can provide users with a three-dimensional virtual world similar to the real world, with the characteristics of immersion, interaction, and imagination. The virtual reality system can receive movements or other instruction information from the user through input devices such as helmets and gloves, process them in the system, and affect human senses such as vision, hearing, and touch through some output devices, so that people can produce the same feeling like a certain behavior in the real world.

AR applications are being developed day by day and are receiving more and more attention because they can combine computer-generated data with the real world through hardware devices. AR sports dance training is extremely sensitive to delay, and it has high requirements for computing and communication. Moreover, when performing AR sports dance training on a mobile device, it consumes the battery of the mobile device [1] and is unable to meet user expectations. In order to solve this problem, literature [2]

has proposed the use of MEC to solve the current problems. Users will migrate the calculation of large amounts of data involved in running AR sports dance training to the nearest cloud connected to the base station. Executed on the server, compared with local computing, it can save local energy consumption, and compared with central cloud computing, it can reduce transmission delay.

The work of literature [3] shows that by jointly optimizing the allocation of communication resources and computing resources, it is possible to significantly reduce mobile energy consumption under delay constraints, and their work can be applied to multiple users independently running general-purpose applications. However, AR applications have their unique properties. All users may upload and download part of the same data, and their computing tasks are also shared on one or more servers. Therefore, communication and computing resources can be jointly optimized and reduce communication and calculation overhead [4].

The AR sports dance application superimposes some computer images onto real-world images through the screen and camera of the mobile device. Five components are needed to complete this process [5]: (1) video source, which can first obtain the original video frame from the mobile camera; (2) tracker, which recognizes and tracks the relative position of the user in the current environment; (3) mapper, which builds a model of the current environment; (4) object recognizer, which recognizes known objects in the current environment; and (5) renderer, which displays the processed frames. The video source and renderer components must be executed on the mobile device, and the calculations performed by the tracker, mapper, and object recognizer with the largest amount of calculation can be offloaded to the cloud. In addition, if the task is offloaded, the mapper and object recognizer can collect input from all user devices in the same geographic location, limiting the redundant information transmitted in the user's uplink. In addition, the results calculated by the mapper and object recognizer can be multicast from the cloud to all users at the same location in the downlink.

Different from literature studies [6, 7], this paper clearly illustrates the collaborative nature of AR sports dance applications to solve the problem of minimizing mobile energy consumption through communication and calculation of dance motion resource allocation under delay constraints [8, 9]. In [10], the continuous convex approximation method is used to solve the mobile energy consumption minimization problem, and the solution method is complicated.

2. Background and Related Work

2.1. Research Background. With the development of mobile networks and smart devices, a large number of resource-sensitive applications have been spawned, such as large-scale interactive games, virtual reality AR, and augmented reality AR [2]. These applications require devices with ultrahigh computing power and battery energy to support ultralow latency requirements, and the most significant of them is AR applications. In the application of AR in dance teaching, in

order to realize the interactive strategy of AR in dance teaching and the functional strategy requirements of dance teaching on AR, it is necessary to clarify the four major components of the virtual reality system. The major components are the motion matching system, the motion capture system, the three-dimensional drawing system, and the dance movement database. The movement matching system will match and compare the student's dance movements with the standard movements in the dance movement database; the movement capture system can quickly capture the students' dance movements in real time; the three-dimensional drawing system can perform three-dimensional dance movement information that needs to be learned. The model is constructed to form a dance animation that can be demonstrated, replayed, adjustable speed, and visualized. In the dance movement database, a large number of standard dance moves are stored. AR technology highly integrates the real-world environment and virtual information and uses computer graphics to combine the real world with computer-generated virtual images to generate information that is perceived by human senses to obtain a sensory experience beyond reality. AR supports new applications and services such as 3D movies, virtual games, etc. [3], which can be used on mobile devices such as smartphones, smart glasses, and tablets. However, when AR processes complex data such as videos and images, it also needs to interact with users in real time. Therefore, AR requires ultralow latency and extremely high data transmission rates. In order to meet this demand, with the development of 5G, the traditional cloud computing network architecture is quietly sinking to the edge of the network, and an emerging computing model-MEC has emerged. The MEC server is closer to the user and can solve the AR problem well.

AR application is a delay-sensitive application, which has high requirements for computing and communication. Therefore, AR tasks can be offloaded to nearby MEC servers for execution so that AR has less execution delay and mobile devices have lower energy consumption. At present, the technology of applying MEC to AR has attracted extensive attention and research from academia and industry and has achieved preliminary results. Literature [11] proposes that users migrate a large amount of data involved in running AR applications to a cloud server close to the base station for calculation. Compared with local computing, it can save the energy consumption of mobile devices, and compared with traditional cloud computing, the transmission delay can also be reduced. The work of literature [12] shows that joint optimization of the allocation of communication resources and computing resources can significantly reduce the energy consumption of mobile devices under delay constraints. However, all users using AR applications may upload and download the same data, and the data that need to be migrated may also be shared on one or more servers. Therefore, literature also proves through experiments that communication can be jointly optimized and computing resources reduce overhead.

Although the abovementioned researches have achieved good results in the application of AR and MEC, the application of sports dance training still needs further

improvement. In addition, the difference is that this article considers using an edge computing framework to transfer complex AR processing data tasks to the edge server for calculation, so as to speed up dance movement matching, capture, imaging, and other processing.

2.2. Research on AR. We realize the construction of a virtual environment based on computer technology, create corresponding scenes suitable for sports dance training, and thus achieve the effect of sports training. This is the basic principle of the integration of AR into the education field. This technology has strong immersion, interactivity, and conception and belongs to the category of computer advanced human-computer interaction applications. In the process of this technology's effectiveness, multidimensional graphics technology, multimedia technology, simulation technology, sensor technology, etc. will be penetrated, and the combination of technologies will ensure that the sensory functions can be simulated, that is to say, as if you are in it, you can hear and touch. At this point, we can see that it belongs to a kind of artificial virtual environment creation process. For the user, with the help of such technology, the interaction with the system can be realized, which can be physical or verbal, and then create a multidimensional anthropomorphic spatial pattern, which will make people feel immersed sense. From this point of view, virtual display technology is not only used for presentation media but also an effective design tool.

2.3. Evaluation Method of AR in Sports Dance Movements. There are many difficult movements in sports dance training, and these movements need to be carried out in accordance with the corresponding music rhythm, which makes it highly performative, which means that the participants need to ensure the music, background, and body. The language is integrated to complete the corresponding deductive task, thereby presenting the values of health and beauty. A set of sports dance training must be of high quality, be able to skip frames for difficult movements, meet the demands of programming and performance, and reach certain artistic targets. In the actual education and teaching process, if the real environment of the competition scene can be simulated, it can ensure that each combined movement enters a state of refined analysis and research. Relying on such refined data can make the students' dance level get. A better evaluation can find the deviations in the corresponding training node and then ensure that the subsequent training develops and progresses in a more ideal direction.

The use of reality and virtual technology can make the various movement parameters of the students in the aerobics process be presented. Whether it is the movement details at the time point or the connection in the continuous movement, it can be presented in this way, so that you can better understand the training knowledge of each student. With the help of AR, the corresponding movements can be incorporated into the virtual environment, and multiple movements can be rehearsed before the corresponding movement points are even reached. With the help of video and

professional research software, accurate movements and student movements can be actively realized. The comparative analysis between the two can provide a more comprehensive understanding of the defects and deficiencies of the students in the movement, thereby reducing the difficulty for the teacher to find the differences in the training of the students, so that the teacher can grasp the corresponding movement defects more quickly. The development of follow-up training plans can make the actual education and teaching design develop in a more coherent and interactive direction. Based on this consideration, the actual student's movement quality, corresponding to the overall training effect, will also develop in a more ideal direction [13].

Incorporating the standard parameters of domestic and international sports dance venues and the specifications of actual sports venues into the actual system, and the reasonable setting of subsidy roles from the perspective of software can enable the actual training and competition context to be constructed. It can guide students to enter a more ideal training pattern, which makes the actual AR and sports dance training and competition related. For students, being able to exercise in such a virtual environment can quickly enter the state. During this process, the system will record the student's movement performance and rely on the previously established movement assessment to realize the actual training in time. After the results are generated and the corresponding training report is obtained, students can see their own deficiencies and deficiencies in the system and then use this as a node for subsequent training, so that students can be guided to a good state of self-learning and self-adjustment. In this way, the actual sports dance training can develop and progress in a more autonomous and personalized direction [14].

3. Optimization of Sports Dance Movement Evaluation Based on MEC and AR

3.1. Description of the Dual Objective Problem. At present, immersive AR is still in the immature stage. Some key technologies are being researched, improved, and perfected, such as high-definition panoramic three-dimensional display technology and relatively natural interaction methods. Related hardware devices are inconvenient to use and have unsatisfactory effects. In this case, it is difficult to meet the requirements of the virtual reality system, which affects the interest of the user experience; and the equal power MEC optimization solution that considers the collaborative transmission between users mainly solves the lag of screen imaging and user interaction, and viewing 3D images through eyepieces. The problem of the gap between the real images enables the AR application in sports dance to increase the operating speed of the equipment and reduce the energy consumption in transmission, so that the quality of the displayed pictures is greatly improved.

This section presents the calculation model of sports dance movement on the mobile edge in the AR scenario. Considering a base station, there are a total of K users running AR for sports dance training. The user set is $K = \{1, 2, \dots, k\}$, and the base station is equipped with a

cloud server with high computing power. It is used to process the data of sports dance movements uploaded by users. The cloud server is connected to a single-antenna base station and uses Time Division Duplexing (TDD) to provide services for all users in the dance training venue through a flat frequency fading channel. Based on the content introduced above, this article assumes that the migrated dance motion application shares input, output, and computing tasks, which are related to the tracker, mapper, and object recognizer components. This section specifically introduces the collaboration in the process of sports dance movement data transmission.

3.2. Uplink Sports Dance Data Transmission. When user $k \in K$ runs AR sports dance application in an area, the dance movement data to be processed, such as the input bit D_k^u of object recognition, should be sent to the cloud server for processing [15]. It is assumed that part of the input bit D_s^u is the same in each user's input bit. This means that this part of the dance movement data can be transmitted by all users in the area, rather than the need for multiple users to upload repeatedly. In this paper, the same input bit is called the shared input bit D_s^u and $\sum_{k=1}^K D_{s,k}^u = D_s^u$. The input bit $D_{s,k}^u$ is shared by the cooperative transmission part of each user k and $\sum_{k=1}^K D_{s,k}^u = D_s^u$. Then, the input bit independently uploaded by each user k is $\Delta D_k^u = D_k^u - D_s^u$.

3.3. Cloud Server Processing Sports Dance Data Process. The cloud server processes the dance movement data uploaded by the user to generate the output bits required by the user [16]. The number of CPU cycles required by the cloud server to process the input bit D_k^u of user k is C_k . Assume that a part of the CPU cycles is used to calculate and generate all the output bits required by the user, for example, update the dance training environment model that the user overlaps. This article refers to this part of the CPU cycle number as the shared CPU cycle number C_s , and $C_s \leq \min\{C_k\}$; then, $\Delta C_k = C_k - C_s$ CPU cycles are used to calculate the output bits required by user k individually.

3.4. Downlink Sports Dance Data Transmission. Part of the output bits needs to be passed to all users [17]. For example, users in the same geographic location need the output bits of the mapper component to update the map. To describe this scenario with a model, this article assumes that $D_s^d \leq \min_k\{D_k^d\}$ output bits can be sent to all users in the dance training venue in a multicast manner, and $\Delta D_k^d = D_k^d - D_s^d$ bits need to be sent to each user k in a unicast manner.

4. Transmission Process of Sports Dance Movement Evaluation System

In the key data frame of the dance movement of this system, the shared communication and calculation tasks are

performed first, and then, the traditional independent migration tasks are performed, as described below.

4.1. Sports Dance Data Transmission Rate. Assuming that the channel state remains unchanged during the transmission process, let α'_k be the channel gain between user k and the base station, and α_k is the normalized value, then the uplink dance movement data transmission rate is shown as follows:

$$R_k^u(P_k^u) = B^u \alpha_k \text{lb} \left(1 + \frac{\alpha'_k P_k^u}{N_0 B^u \alpha_k} \right), \quad (1)$$

where P_k^u is the transmission power of the mobile device of user k , B_k^u is the transmission bandwidth allocated to user k , and B^u is the total uplink transmission bandwidth; then, $\sum_{k=1}^K B_k^u = B^u$ and N_0 are the noise power spectral density.

For the shared output bit $D_{s,k}^u$, it can be sent to all users in a multicast manner, and the dance movement data transmission rate of the downlink multicast is shown as follows:

$$R_{m,k}^d(P_m^d) = B^d \text{lb} \left(1 + \frac{\alpha'_k P_m^d}{N_0 B^d \alpha_k} \right), \quad (2)$$

where P_m^d is the downlink multicast transmission power and B^d is the total downlink transmission bandwidth.

For the output bit D_k^d that is sent to user k separately, it is sent by unicast, and the data transmission rate of the downstream unicast dance movement is shown as follows:

$$R_k^d(P_k^d) = B^d \alpha_k \text{lb} \left(1 + \frac{\alpha'_k P_k^d}{N_0 B^d \alpha_k} \right), \quad (3)$$

where P^d is the downlink transmission power corresponding to user k , B^d is the downlink transmission bandwidth allocated to user k , and $\sum_{k=1}^K B_k^d = B^d$.

4.2. Sports Dance Data Transmission and Processing Time. Define the time T_s^u required for user k to upload part of the shared input bit $D_{s,k}^u$ as $T_s^u = D_{s,k}^u / (R_k^u(P_k^u))$.

The time T_s^c required by the cloud server to execute the number of shared CPU cycles is $T_s^c = C_s / (f_s F_c)$, where F_c is the total processing capacity of the cloud server and f_s is the scale factor allocated by the cloud server to execute the number of shared CPU cycles.

The time T_s^d required for user k to receive the multicast shared output bit D_s^d is $T_s^d = D_s^d / (R_{m,k}^d(P_m^d))$.

In the same way, the time T_1 required for user k to independently upload the remaining number of bits ΔD_k^d is $T_1 = \Delta D_k^d / (R_k^u(P_k^u))$.

The time T_2 required by the cloud server to execute the number of individual CPU cycles for user k is $T_2 = \Delta C_k / (f_k F_c)$, where f_k is the scale factor that the cloud server allocates to execute the number of independent CPU cycles for different users k , and $\sum_{k=1}^K f_k \leq 1$.

The time T_3 required for user k to receive the unicast output bit number ΔD_k^d is $T_3 = \Delta D_k^d / (R_k^d(P_k^d))$.

It can be seen from the above that in the sports dance movement system, the delay T required for user k to perform mobile edge calculation is

$$T = \max_k (T_s^u) + T_s^c + \max_k (T_s^d) + (T_1 + T_2 + T_3). \quad (4)$$

4.3. Sports Dance Data Transmission Energy Consumption. The energy consumed by users performing MEC lies in the uplink dance movement data transmission and the downlink dance movement data reception.

The energy generated by user k due to the downlink dance movement data reception is

$$E_k^u(D_{s,k}^u) = \left(\frac{D_{s,k}^u + \Delta D_k^u}{R_k^u(P_k^u)} \right) (P_k^u). \quad (5)$$

The energy generated by user k due to downlink data reception is

$$E_k^d(P_k^d, P_m^d) = \left(\frac{\Delta D_k^d}{R_k^d(P_k^d)} + \frac{D_s^d}{R_{m,k}^d(P_m^d)} \right) l_k^d, \quad (6)$$

where l_k^d is the energy consumed by user k to capture the downward dance movement data per second.

4.4. Uplink Equal Power Allocation. Considering that the uplink transmission power of each user is a fixed value, and the uplink and downlink bandwidth of user k changes in proportion to its channel gain [18], suppose that the optimization variable includes the number of shared input bits uploaded by user k D_k^s , and the cloud server allocates it for sharing the scale factor f_s for the number of CPU cycles, the cloud server allocates a scale factor f_k for executing the number of independent CPU cycles for different users k , and the base station corresponds to the downlink transmission power P_k^d of the user k and the downlink power P_m^d for multicast.

Consider the following optimization problem [14]:

$$\begin{aligned} & \min_z \sum_{K \in M} E_k^u(D_{s,k}^u) + E_k^d(P_k^d, P_m^d), \\ & \text{S.t } T \leq T_{\max}, \\ & \sum_{k \in K} f_k \leq 1, 0 \leq f_s \leq 1, f_k \geq 0, \forall k \in K, \\ & \sum_{k \in K} D_{s,k}^u = D_s^u, \\ & \sum_{k \in K} P_k^d \leq P_{\max}^d, P_m^d \leq P_{\max}^d, P_k^u \leq P_{\max}^u, \forall k \in K. \end{aligned} \quad (7)$$

5. Performance Evaluation

This section presents the results of minimizing the total energy consumption of the user terminal by using the collaborative characteristics of AR sports dance applications based on MEC and using power allocation methods such as uplink transmission. Consider a training venue where 10

users are running AR sports dance applications. The users are randomly distributed in the training venue, and the wireless channel meets the Rayleigh fading [18]. In the sports dance movement system, each user needs to upload dance movement data D_k^u of 10^6 bits, each user needs to receive dance data D_k^d of 10^6 bits, and the total bandwidth of the uplink and downlink channels B^u and B^d is 1×10^8 Hz. Downlink users receive dance movement data energy l_k^d of 0.628 J/s. In addition, the two parameters, N^0 and F_c , are set to 1×10^{-10} and 10^{11} CPU cycles/s, respectively.

This paper proposes to use the sharing factor n as an indicator of the degree of user's calculation migration, where n is the ratio of shared bits to total bits; when $n = 0$, users perform independent calculation migration; when $n = 1$, all bits are shared bits, and users perform fully shared calculation migration.

5.1. To Minimize the Delay as the Objective Function. This article first takes minimizing the delay as the objective function and simulates the sharing factor $n = 0$, that is, the situation where the user independently performs calculation migration and uses the equal power allocation method to obtain the minimum delay that the sports dance movement system can achieve when using different uplink transmission powers. The simulation result is shown in Figure 1.

It can be seen from Figure 1 that if the equal power allocation method is adopted, the uplink transmission power must meet at least 1.0 W to make the delay constrained within 0.1 s; when the delay is constrained within 0.16 s, a smaller uplink transmission power can be satisfied. Transmission requirements are as follows. In this article, we consider that the maximum uplink transmission power of the user is 0.27 W, and the maximum downlink base station transmission power P_{\max}^d is 20.05 W.

If we consider the model in [19] and take minimizing the delay as the objective function, we can get the simulation results of the minimum delay that the sports dance movement system can achieve when using different sharing factors, as shown in Figure 2.

It can be seen from Figure 2 that under the same simulation conditions, the uplink transmission power is used as the optimized parameter in the literature [20], and the bandwidth is evenly allocated. When the sharing factor is used, the delay can be constrained within 0.998 s.

5.2. To Minimize the Total Energy Consumption of Users as the Objective Function. Taking minimizing the user's total energy consumption as the objective function, on the basis of Section 3.1, simulations are carried out under the conditions of $T = 0.1$ s and $T = 0.16$ s, respectively, for the delay constraints. In the simulation, the continuous convex approximation method and the uplink equal power transmission method are used for comparison. Figures 3 and 4 are the simulation comparisons under the delay constraints of $T = 0.1$ s and $T = 0.16$ s, respectively.

It can be seen from Figure 3 that when the maximum delay is limited to 0.1 s, considering that the user's uplink transmission power is at most 0.27 W, when the uplink

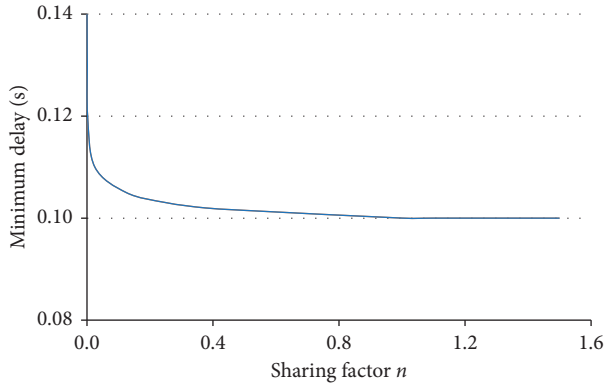


FIGURE 1: Uplink transmission power versus minimum delay.

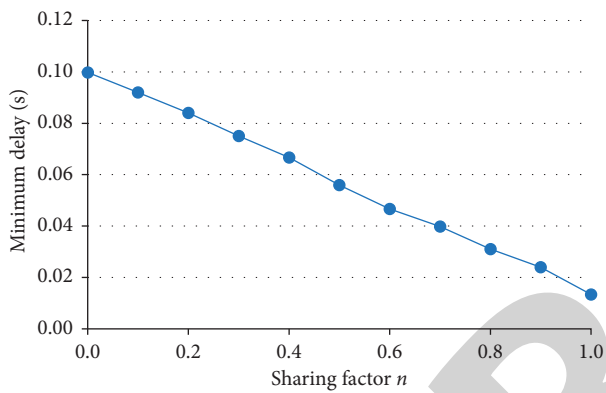


FIGURE 2: Sharing factor versus minimum delay.

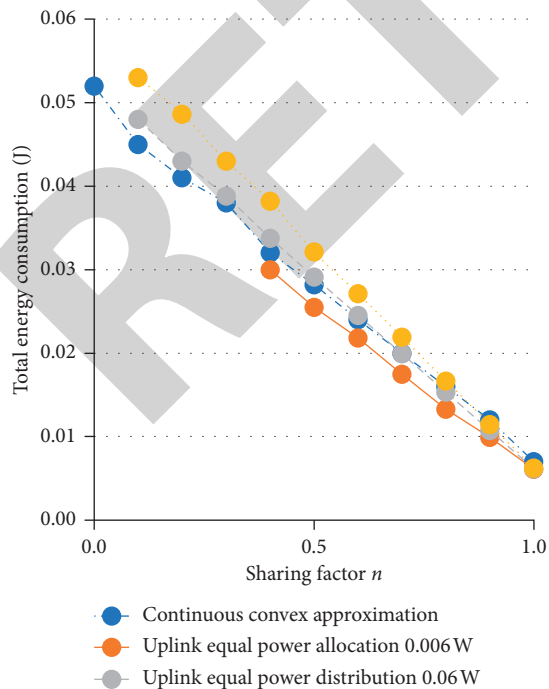


FIGURE 3: Sharing factor versus sum-energy consumption when the maximum delay is 0.1 s.

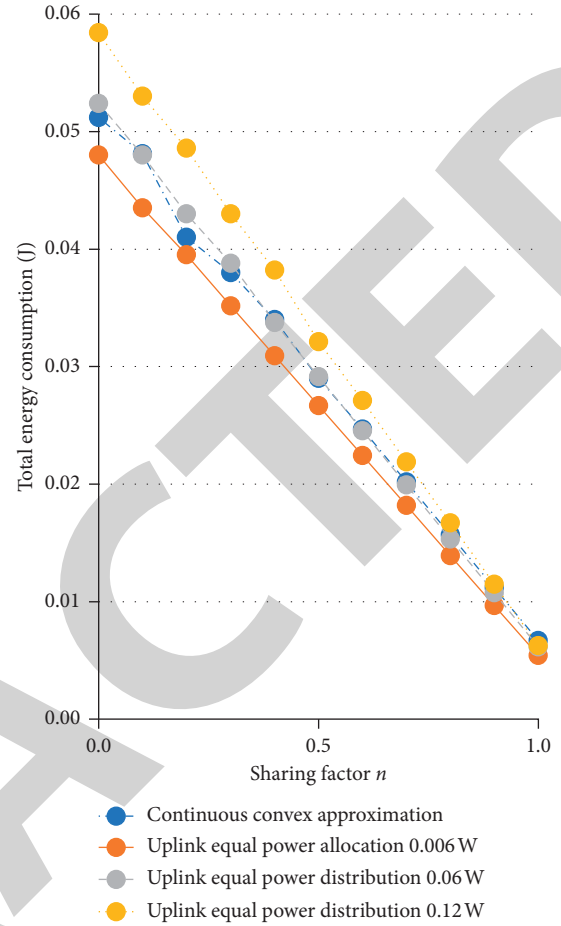


FIGURE 4: Sharing factor versus sum-energy consumption when the maximum delay is 0.15 s.

transmission power is controlled within 0.27 W, the uplink equal power allocation is used in the case of the sharing factor $n = 0$. The method cannot meet the transmission requirements. When the transmission power decreases, the total energy consumption also decreases, but under the delay constraint, when the sharing factor is small, the large amount of total dance movement data cannot meet the transmission requirements.

It can be seen from Figure 4 that when the maximum delay is limited to 0.16 s, the uplink equal power allocation method is used, and when the user transmission power is 0.06 W, the total energy consumption is relatively close to the continuous convex approximation method in [12], but when the uplink transmission power takes a small value, for example, when the uplink transmission power is 0.006 W, the total user energy consumption obtained by the uplink equal power method is less than the total user energy consumption of the continuous convex approximation method in the literature [21] and can meet the transmission needs.

From the comparison of Figures 3 and 4, it can be found that, with the increase of n , the total energy consumption gap of users obtained by using different uplink equal power transmissions is constantly

decreasing. This is due to the total energy consumption of users and the uplink transmission power and transmission. The number of bits is proportional. When the sharing factor is considered in the uplink dance movement data transmission, the number of bits that the user needs to transmit decreases with the increase of n . Therefore, when n does not change, the number of bits transmitted by the user does not change, and the total energy consumption increases; when the uplink transmission power is constant, with the increase of n , the number of bits transmitted by the user decreases, and the total energy consumption decreases; combining the two factors, as n increases, the number of transmitted bits decreases, and the increase in power brings a less obvious increase in total energy consumption.

At the same time, when the uplink power distribution method is used, the solver needs only one solution to obtain the optimal result, while the continuous convex approximation method requires multiple iterations of the solution to approximate the optimal result. It is found through simulation that in the case of accuracy $\varepsilon = 10^{-4}$, the number of iterations is usually between 1 and 35; that is, the solution time of the continuous convex approximation method is a multiple of the uplink power distribution method.

6. Conclusions and Future Research Directions

Aiming at the cooperative transmission characteristics existing in AR sports dance training scenarios, this paper combines bandwidth and channel gain based on MEC to allocate user bandwidth and uses uplink power transmission methods to minimize the total number of users. The energy consumption optimization function, by solving the convex optimization problem, obtains the optimal sports dance movement resource allocation plan. Compared with users performing MEC independently, the user cooperative transmission scheme can significantly reduce the total energy consumption of user equipment during sports dance training. At the same time, when using uplink power transmission methods, it can reduce compared with continuous convex approximation. The sports dance movement system has computing time, and under a certain delay requirement, it can meet the transmission requirements with a smaller power.

The combination of computer virtual technology, edge computing technology, and sports dance training projects has opened up a convenient channel for various types of sports training. The application of virtual technology in the field of sports reduces the danger of training and creates an objective condition that is difficult to achieve or even impossible to achieve in order to optimize the evaluation of sports dance movements based on the network architecture of edge computing and augmented reality technology, in addition to exploring the energy consumption, time optimization, and enhancement technology of MEC, the quality of user experience, and deployment research and practice on optimization and collaboration mechanisms, so as to realize the deep integration of sports dance and edge computing technology under the augmented reality technology, with a view to elevating college sports training to a whole new level.

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.

References

- [1] J. Broderick, J. Duggan, and S. Redfern, "The importance of spatial audio in modern games and virtual environments," in *Proceedings of the 2018 IEEE Games Entertainment Media Conference (GEM)*, pp. 1–9, Galway, Ireland, August 2018.
- [2] ARPost. "Key News and Events in the Augmented and AR World-September 2018" [EB/OL]. <https://arpost.co/2018/09/28/key-news-and-events-in-the-augmented-and-virtual-reality-technology-world-september-2018/>.
- [3] A. Mahrous and G. B. Schneider, "Enhancing student learning of removable prosthodontics using the latest advancements in virtual 3D modeling," *Journal of Prosthodontics*, vol. 28, no. 4, pp. 471–472, 2019.
- [4] S. Wang, X. Zhang, Y. Zhang et al., "A survey on mobile edge networks: convergence of computing caching and communications[]," *IEEE Access*, vol. 5, 2017.
- [5] M. Qin, L. Chen, N. Zhao et al., "Power-constrained edge computing with maximum processing capacity for IoT networks," *IEEE Internet of Things Journal*, vol. 6, 2019.
- [6] J. Ahn, J. Lee, S. Yoon, and J. K. Choi, "A novel resolution and power control scheme for energy-efficient mobile augmented reality applications in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 9, pp. 750–754, 2020.
- [7] J. Zillner, E. Mendez, and D. Wagner, "Augmented reality remote collaboration with dense reconstruction," in *Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Munich, Germany, October 2018.
- [8] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 398–401, 2017.
- [9] ETSI. ETSI. GS MEC 003[EB/OL]. (2020-12) [2020-12-16]. https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02_02_01_60/gsmec003v020201p.pdf.
- [10] CBR Staff writer. "Facebook bets on gaming future with \$2bn oculus acquisition" [EB/OL]. <https://www.cbronline.com/enterprise-it/software/facebook-bets-on-future-with-2bn-virtual-reality-startup-acquisition-260314-4203654/.%202019-04-13..>
- [11] J. Ahn, Y. Kim, and R. Kim, "Virtual reality-wireless local area network: wireless connection-oriented virtual reality architecture for next-generation virtual reality devices," *Applied Sciences*, vol. 8, no. 1, p. 43, 2018.
- [12] Y. Quan, H. Zhou, J. Li et al., "Toward efficient content delivery for automated driving services: an edge computing solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, 2018.
- [13] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 235–250, 2020.
- [14] D'E. Maria, "Immersive virtual reality as an international collaborative space for innovative simulation design," *Clinical Simulation in Nursing*, vol. 54, pp. 30–34, 2021.

Research Article

Positioning of Apple's Growth Cycle Based on Pattern Recognition

Wenfeng Li,^{1,2} Yulin Yuan,^{1,2} Shikang Hu,^{1,2} Mei Li,² Wenxiu Feng,² and Jiaxin Zheng^{1,2} 

¹The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Yunnan Agricultural University, Kunming 650201, China

²Engineering Center of Yunnan Colleges and Universities for Plateau Characteristic Modern Agricultural Equipment, Yunnan Agricultural University, Kunming 650201, China

Correspondence should be addressed to Jiaxin Zheng; 2017019@ynan.edu.cn

Received 20 April 2021; Accepted 3 May 2021; Published 18 May 2021

Academic Editor: Jianhui Lv

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

The positioning of the apple growth cycle plays a very important role in predicting the development of apples and guiding fruit farmers in agricultural operations. The traditional method of manually positioning the apple growth cycle has problems such as low efficiency and poor accuracy. Pattern recognition provides support for continuous and rapid positioning during the apple growth process. Under the natural conditions of the orchard, due to the large differences in the individual colors of the apples during the growth process and the influence of factors such as light changes, the photographed apple images are more complex, which brings certain difficulties to the segmentation and recognition of the apples. In this paper, pattern recognition is used to automatically identify and extract the growth stages of apples, a hue intensity (HI) color segmentation algorithm based on a Gaussian distribution model based on prior knowledge is studied, and then an active shape model (ASM) is used to identify each period of apple growth based on pattern recognition. After a series of experimental verifications, the ASM-based automatic identification method proposed in this paper is feasible and can identify the various growth periods of apples, thereby serving the mechanized production of apples.

1. Introduction

Apples have always been recognized as a fruit that can prevent a variety of diseases. Doctors said, "A study by the University of California in the United States has found that apples contain a flavonoid substance that can delay cell aging, protect the heart, and prevent cerebrovascular diseases." In 2020, China has become the world's largest apple producer, with apple planting area and output accounting for more than 50% of the world. The apple industry has brought good economic benefits and has become the main source of income for farmers. Precisely positioning the growth cycle of apples to grasp its development status is of great importance for guiding production management such as fertilization and irrigation. In the growth process of apples, it is important to accurately assess the growth of apples and perform appropriate work at the appropriate time.

Smart agriculture is the integration between agriculture and Artificial Intelligence (AI) technology and modern

information technology, and it is also a technology that is urgently needed in the development of global agriculture. Nondestructive monitoring of crop growth is one of the important links in smart agriculture. It requires real-time and accurate acquisition of plant growth information to guide the fine management of crops. In particular, the monitoring of crops in different growth periods is an important part of crop growth monitoring. First of all, understanding the growth period of crops can help people analyze the relationship between field crop growth and environmental conditions; in addition, it can effectively guide operations, thereby significantly improve the level of agricultural mechanization and crop yields [1, 2].

Pattern recognition can detect subtle changes in plants due to malnutrition earlier than human vision, providing a reliable basis for timely irrigation and supplementation of nutrients. Under the current circumstances, if apple's economic benefits and automated production are effectively improved, this will have important practical significance. In

the process of mechanized production, orchard management is a very important task. The apple's growth cycle is divided into six major periods, the first is the growth period of the seedlings; the seedlings need to be carefully maintained and provided with appropriate water and fertilizer management to make the apples grow better. Next is the flower bud differentiation period, and attention should be paid to temperature and light control. The temperature is 12~18°C; during the flowering period from April to June, stop watering to avoid falling flowers; then in the flowering period of apples, supply enough water before flowering, and stop watering at the flowering period to prevent the flowers from falling off. In the case of poor growth, it is necessary to cut the flower buds in time; cut off the bad flowers and poorly grown flowers to maintain a healthy growth state. This is followed by the physiological fruit drop period and the fruit expansion period of the apple, and the maintenance work should be done well. Finally, it should be fertilized and pruned in time during its maturity period to allow it to grow and produce better results. In actual production, the external morphological structure of the crop has undergone tremendous changes from seedling growth to maturity, and these morphological and structural changes are also very significant for the image. Therefore, it is very necessary and urgent to carry out pattern recognition research on the growth and development cycle of apples with the help of computer vision and image processing theories and methods.

The main contributions of this paper are summarized as follows: (i) we use the HI color based on Gaussian distribution, which is suitable for the image segmentation algorithm of apples at each period of the growth process in the complex environment of the orchard to perform apple image segmentation. (ii) On the basis of correct segmentation, an ASM based on pattern recognition is used to identify apples in the growth process of each period. In other words, the novelty of this paper is mainly to use the HI color based on Gaussian distribution to perform apple image segmentation, and then use ASM to identify the whole apple growth cycle from seedling stage to mature fruit.

The rest of the paper is organized as follows: Section 2 analyzes and summarizes domestic and foreign research work in crop growth and development monitoring, crop image segmentation, and crop automatic identification. Section 3 introduces the collection and segmentation of apple images at each period of the growth process. Section 4 uses ASM to identify apples at each period of the growth process. Experimental results are reported in Section 5, and finally Section 6 concludes this paper.

2. Related Work

Nowadays, labor on farms and orchards depends largely on skilled farmers. Manual labor wastes time and increases the cost of production, while uneducated and inexperienced workers make unnecessary mistakes. In [3], Zhao et al. present a review on some key vision control techniques and their potential applications in fruit or vegetable harvesting robots. In [4], the intelligent agriculture becomes a popular concept. In [5], Wang et al. develop a computer vision-based

system for automated, rapid, and accurate yield estimation. In [6], there is increasing interest in the use of image processing techniques for crop detection in intelligent weeding applications. In [7], Lu and Sang propose a recognition method for citrus fruits under varying canopy illumination based on color and contour information. In [8], Linker et al. propose a method for detecting apples in natural lighting. It details an algorithm for estimating the number of apples in color images under natural illumination. With the rapid development of AI, pattern recognition is widely used in agriculture [9]. Deep learning is a special kind of machine learning that can be used to classify crops [10–12], crop image segmentation [13, 14], crop target detection [15, 16], and other missions [17]. In [18], Zhang et al. propose a 13-layer convolutional neural network (CNN) for fruit classification with high accuracy. In [19], Chen et al. use a spot detector based on fully connected CNN to extract candidate regions in the image and segment the target region, and use subsequent CNN counting algorithm to calculate the number of fruits. In [20], Dyrmann et al. present a method for automating weed detection and use CNN to detect the weeds. Faster R-CNN [21] uses the Region Proposal Network (RPN) method to detect the Region of Interest (ROI) in an image. In [22], Inkyu et al. use the faster R-CNN method to detect a variety of fruits and produce good results. The faster R-CNN with VGG16 nets [23] is the most advanced method for fruit detection [9]. In [24], Tian et al. propose an improved YOLO-V3 model for detecting apples during different growth stages; this model is used to detect young apples, expanding apples, and ripe apples in complex backgrounds.

Relying on the continuous development of image processing technology and pattern recognition technology, we can use computer technology for crop shape recognition. At present, in crop morphology recognition, research mainly focuses on the appearance characteristics of crops, that is, the measurement of the geometric shape characteristics of the crops, for example, the classification of certain crops by different leaf shape characteristics, including the rectangularity, circularity, and circularity of the leaves. Geometric parameters such as eccentricity, leaf size and invariant moments, and crop classification and identification can also be achieved through crop flowers, fruits. Most of the work done is focused on the study of specific organ morphology of crops, such as leaves and flowers. Few studies involve the use of computer vision systems to automatically identify the growth and development stages of crops. However, the use of pattern recognition to locate the growth cycle of apples is an indispensable part of practical applications, which provides impetus for the development of this paper. Especially different from the previous work, the novelty of this paper is mainly to use the HI color based on Gaussian distribution to perform apple image segmentation, and then use ASM to identify the whole apple growth cycle from seedling stage to mature fruit.

3. Image Collection and Segmentation

3.1. Data Collection. Computer vision usually includes the following modules: light source, camera or camera image capture card, input and output module, and image

processing software. The performance of the image acquisition system will directly affect the quality of the collected data and then affect the subsequent image processing and analysis. Taking into account the actual situation of this research and the goals that need to be completed, the hardware system for apple image segmentation and for identification of each key developmental stage is mainly composed of a computer and an image acquisition system. The image acquisition is usually done by a video camera, ordinary digital or optical camera, and then the digital image in the memory card is converted to the computer. This paper will mainly use digital cameras for data acquisition. The image acquisition system mainly includes tripods, digital cameras, computers, and data transmission equipment. This paper can obtain apple images of different weather conditions and different growth stages through this system and then identify them through image processing related algorithms.

The subject of this paper is apples, and the selected variety is Fuji apple. The image collection time is mainly determined according to the growth status and growth rules of the apple. As far as this research is concerned, it is necessary to consider accurately segmenting apples from images under different weather conditions and automatically identify the four key growth stages of apple flower bud differentiation, apple flowers, physiological fruit drop, and fruit expansion in the orchard. Therefore, we need to shoot continuously at specific times every day. The shooting object is apple (the growth cycle starts from the seedling to the end of the mature period). Apple image collection is mainly concentrated in two time periods. The first time period is mainly to photograph the apples in the orchard. Its function is to analyze the relationship between the green information of the apple and the light under different weather conditions and to lay the foundation for segmentation; the latter time period is mainly to obtain a single apple, and its role is to identify later created training samples. The shooting environment changes include dark light to strong light, weather conditions (sunny, rainy, and cloudy), shooting time (morning, noon, and afternoon), and dry to humid soil. It is also necessary to adjust the height and angle of the camera in time to find a reasonable configuration that can obtain high-quality image data.

3.2. Image Segmentation. The accurate segmentation and extraction of apples in the image constitute a prerequisite for realizing apple growth monitoring. However, the completion of this work is affected by many factors, such as changes in light conditions, changes in surface soil, and complex environments. In order to achieve accurate segmentation of apple images in complex environments, this paper studies an HI color segmentation algorithm based on Gaussian distribution model based on prior knowledge of apples in different weather conditions and different growth and development stages. The algorithm only needs a small number of training samples to complete the segmentation of the apple image.

3.2.1. Existing Crop Segmentation Algorithms. In the algorithm of crop image segmentation, the role of color index is very important. The general color index algorithms mainly include ExG (Excess Green—over-green); ExR (Excess Red—over-red); ExG–ExR (Excess Green minus Excess Red—over-green minus over-red); CIVE (Color Index of Vegetation Extraction—a vegetation color index extraction); and VEG (Vegetation). These algorithms only need to use the three components of red, green, and blue in the RGB color space. The calculation is simple, and it is not sensitive to changes in light intensity; even to complex environments, change has a certain degree of adaptability. However, the prerequisite for the realization of the algorithm is to normalize the color space and finally achieve crop extraction by setting the segmentation threshold.

The method of normalizing the color space is shown in (1), and the values of R , G , and B need to be calculated firstly.

$$\begin{aligned} R &= \frac{R^*}{255}, \\ G &= \frac{G^*}{255}, \\ B &= \frac{B^*}{255}. \end{aligned} \quad (1)$$

On the basis of R , G , and B , r , g , and b are, respectively, normalized into the $[0, 1]$ interval according to (2), and $R + G + B = C$ is assumed.

$$\begin{aligned} r &= \frac{R}{C}, \\ g &= \frac{G}{C}, \\ b &= \frac{B}{C}. \end{aligned} \quad (2)$$

The calculation of the color index is based on the calculation of r , g , and b values in (2).

(i) ExG

$$\text{ExG} = 2g - r - b. \quad (3)$$

(ii) ExR

$$\text{ExR} = 1.4r - g. \quad (4)$$

(iii) ExG – ExR

$$\text{ExG} - \text{ExR} = 3g - 2.4r - b. \quad (5)$$

(iv) CIVE

$$\text{CIVE} = 0.44r - 0.81g + 0.39b + 18.79. \quad (6)$$

(v) VEG

$$\text{VEG} = \frac{g}{r^{(2/3)}b^{(1/3)}}. \quad (7)$$

3.2.2. HI Color Segmentation Based on Gaussian Distribution Model. The HSI color model expresses the way the human visual system perceives the color of an object, where hue (H) represents the difference in color, saturation (S) represents the purity of the color, and intensity (I) represents the brightness of the color. These characteristic components are used to describe the color. The use of the HSI color model in crop segmentation has the following advantages: compared with the RGB color model, its three color feature components can be directly used for segmentation separately; at the same time, it overcomes the influence of lighting conditions [25].

The color distribution of a single-color object changes with intensity on the hue-saturation plane, where a single color is different in each period of apple growth, and it is green in the growth period of the seedling and the physiological fruit drop period; in the flower bud, it is pink at the differentiation stage, yellow at the flowering stage of the apple, and red at the fruit expansion and maturity stage. Therefore, when building a single-color pixel model in the HSI color model, it is necessary to adapt to changes in external lighting conditions. Therefore, a single Gaussian distribution model can be used to represent the distribution of tone values at a certain intensity. The probability density function of Gaussian distribution is shown in

$$f_H(h|I) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(h - \mu)^2\right]. \quad (8)$$

Here, h is the hue value of a single color, μ is the expected value, and σ^2 is the variance. The solution of the parameter values $\hat{\mu}$ and $\hat{\sigma}^2$ under a certain intensity can be estimated by using the maximum likelihood estimation method on the training sample data set. The specific calculation method is shown in

$$\begin{aligned} \hat{\mu} &= \frac{\sum_{k=1}^n H_k}{n}, \\ \hat{\sigma}^2 &= \frac{\sum_{k=1}^n (H_k - \hat{\mu})^2}{n}. \end{aligned} \quad (9)$$

Here, H_k represents the k th hue value of n sample pixels under a specific intensity, and n refers to the total number of single-color pixel samples.

The HI segmentation algorithm used in this paper is based on the HSI color model, and the type of the apple image collected in the experiment is the RGB color model. Therefore, the image needs to be converted from the RGB color model to the HSI color model when building the tone intensity table and before image segmentation. The basic principle of the conversion from RGB color model to HSI color model is as follows: First, the brightness factor needs to be separated from the RGB color model; the chroma is divided into two parts, hue and saturation; and the angle vector is expressed as hue. On the whole, it is a conversion from a unit cube based on Cartesian coordinates to a double cone based on cylindrical polar coordinates [26]. The conversion formula from RGB color model to HSI color model is defined as follows:

(i) H

$$H = \begin{cases} \theta, & G \geq B, \\ \theta + \pi, & G < B, \end{cases} \quad (10)$$

$$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{2R - G - B}{\sqrt{3}(G - B)}\right).$$

(ii) S

$$S = \frac{2}{\sqrt{6}} \times \sqrt{(R - G)^2 + (R - B)(G - B)}. \quad (11)$$

(iii) I

$$I = \frac{R + G + B}{\sqrt{3}}. \quad (12)$$

The values of R , G , and B in (12) all need to be normalized to the $[0, 1]$ interval before calculation.

Set a pixel $p(x, y)$, where (x, y) represent the coordinates of the pixel in the image. According to the HI color segmentation model of Gaussian distribution, $\varphi(x, y)$ represents the deviation (distance) between the hue value of the pixel and its expected value, as follows:

$$\varphi(x, y) = \frac{|H(x, y) - \mu_I(x, y)|}{\sigma_I(x, y)}. \quad (13)$$

Here, $H(x, y)$ and $I(x, y)$ are, respectively, derived from the expected value and variance of the hue intensity look-up table. If the calculated distance is larger, the possibility that the pixel belongs to a green crop is less. In order to accurately segment the crop image, this paper sets a threshold k to judge whether a pixel is a single-color crop; if $\varphi(x, y) \leq k$, $p(x, y) \in \text{crop}$; otherwise, it belongs to the background. It is not difficult to find that the setting of the threshold k will directly affect the accuracy of crop image segmentation, and choosing different thresholds will produce different segmentation results. If the threshold is too small, some of the single-color pixels will be mistakenly divided into other-color pixels. If the threshold is too large, some other-color pixels will be mistakenly classified as single-color pixels. Therefore, it is very important to choose a suitable threshold k . Through testing, this paper finally determines the best k value suitable for image segmentation at each stage of apple growth (the experimental part will be discussed in detail).

4. Recognition and Positioning of Different Periods in the Growth Process of Apples

The identification of each period in the apple growth process is carried out on the basis of accurately segmenting the apples. After image processing, the influence of other factors on the identification effect is reduced. Researchers at home and abroad mainly use information such as shape parameters, color, and texture characteristics to identify single leaves of plants; to identify flower bud differentiation, flowering period, and fruit period, they count on the basis of

identifying single leaves and then determine which period the plant is in. In this study, ASM method was used to identify the growth cycle of apples, taking into account the overall geometric shape of the apples in each period and reducing the influence of different leaves on the identification results.

4.1. ASM. ASM is a statistical model method applied to complex shape matching proposed by scholars [27–29]. This research discusses it from three aspects, namely, the point distribution model, the local gray-scale texture model, and the matching target search process. Its basic process is to first select a series of training samples, then use manual calibration to mark the contour (feature points) of the object shape, and then perform alignment operations on these training samples. On this basis, the shape model is established, and the constraint condition of the shape parameter change is obtained. Finally, search for the target that matches the shape model in the tested image, so as to complete the recognition of the target object. The ASM algorithm has the characteristics of fast running speed and high target positioning accuracy, and it is also suitable for recognition under different illumination, complex background, and occlusion. Besides, it can statistically model complex shapes of targets to complete the recognition work. In the matching search process, the shape model is adjusted by parameters to constrain the overall shape of the model.

4.2. Point Distribution Model. We use T to denote a specific training set. The training set T contains N shapes, and each shape has n marked points. It can be expressed in another way; that is, there are N coordinate points for each shape's mark point. The training set is represented by (a_{ij}, b_{ij}) : the j calibration coordinate point of the i th shape and the n calibration point of the i th shape are represented by a vector in the training set [30].

$$a_i = (a_{i0}, b_{i0}, a_{i1}, b_{i1}, \dots, a_{i,n-1}, b_{i,n-1})^T, \quad i = 1, \dots, N. \quad (14)$$

When manually calibrating the target contour feature points, it is necessary to ensure that the calibration sequence of the feature points in each image is consistent with the calibration position. Otherwise, the alignment of the training samples in the next step will be affected, and the modeling will be unreasonable. There are different shooting angles and different shooting distances during the data acquisition of a certain key growth period of apples. Therefore, the positions and scales of the apples in the images are different, which causes the distribution of training samples in the two-dimensional space to be disordered, resulting in failures. This shows the changing law of the shape of an apple during a key growth period, so it is unreasonable to directly count and process the training samples. Therefore, it is necessary to perform standardized alignment processing on the manually labeled training samples. The main operation steps are as follows: First select

a certain sample shape as the basic shape, and other samples are as close as possible to the basic shape through the steps of scale enlargement and reduction, rotation, and translation.

If we only align two samples, their corresponding shape vectors are a_1 and a_2 , respectively. Let a_1 be the basic shape model, then enlarge and reduce a_2 by a certain scale a_2 , and rotate the angle β and translate t , so that the shape of a_2 is as close as possible to the shape of a_1 . Under the framework of least squares, the a_2 shape vector is standardized to a_1 by adjusting these parameters.

$$\begin{aligned} a_2' &= R(\beta, C)a_1 + t \\ &= \begin{pmatrix} a_1 C \cos \beta - b_1 C \sin \beta \\ a_1 C \sin \beta + b_1 C \cos \beta \end{pmatrix} + \begin{pmatrix} t_a \\ t_b \end{pmatrix}. \end{aligned} \quad (15)$$

By the least square method, one obtains

$$L = [a_1 - R(b_2) - t]^T [a_1 - R(b_2) - t]. \quad (16)$$

Partially differentiate β, C, t_a, t_b ; set their value to 0; and then solve these equations to get these parameter values.

Therefore, it can be extended to the alignment of N samples. First, select the shape vector of a sample as the basic shape vector, and then align other shape vectors to the basic shape vector according to the above calculation method. After all are aligned, find their average shape model and make it the new basic shape vector. Then, align other shape vectors to this new basic shape vector, and repeat this operation until the convergence condition is met.

The average model in the shape model is defined as follows:

$$\bar{a} = \frac{1}{K} \sum_{i=1}^K a_i. \quad (17)$$

The final shape model in the training set can be expressed by the sum of the changes of the average shape model and the weight model.

$$a = \bar{a} + Vq. \quad (18)$$

Here, \bar{a} is the average shape model, V is the transformation matrix composed of component eigenvectors, and $q = [q_1, q_2, \dots, q_t]$ is the shape weight vector, which is calculated by the following:

$$\begin{aligned} q &= [q_1, q_2, \dots, q_t] \\ &= V^T (a - \bar{a}). \end{aligned} \quad (19)$$

The change of the shape weight m will change the corresponding shape model, so the shape model of ASM is generated. Searching and matching similar targets in the image can make the shape model approximate the shape of the target image by adjusting the weight vector m of different shapes, and the m constraint changes within a certain range. The limitation on m is to avoid the change of the shape model during the matching process that does not conform to the actual situation. The value of m is calculated by the probability distribution model of the parameters.

$$-2.98\sqrt{\varepsilon_n} \leq m \leq 2.98\sqrt{\varepsilon_n}. \quad (20)$$

Here, ε_n is eigenvalue, $\varepsilon_n \geq \varepsilon_{n+1}$.

4.3. Local Gray Texture Model. The establishment of the local gray-scale texture model firstly needs to analyze the gray-scale information [31]. The calculation process for the gray information of the y model feature point in the x image is to take this point as the center and take k sampling points along the two sides of the model contour normal. Therefore, there are $2k + 1$ pixel points in total. The gray information is recorded as g_{xy} . In order to reduce the influence of the model in the process of translation and zooming, the gradient value of g_{xy} is calculated:

$$\text{Grad}g_{xy} = [g_{xy,2} - g_{xy,1}, \dots, g_{xy,2k+1} - g_{xy,2k}]. \quad (21)$$

At the same time, in order to reduce the influence of different lighting conditions, we have done a normalization process on the basis of the gradient value $\text{Grad}g_{xy}$:

$$u_{xy} = \frac{\text{Grad}g_{xy}}{\sum_{i=1}^{2k} |\text{Grad}g_{xy}|}. \quad (22)$$

Then, calculate the average gray information model \bar{u}_y corresponding to all the feature points of the model.

$$\bar{u}_y = \frac{1}{N} \sum_{i=1}^N u_{xy}. \quad (23)$$

Calculate the covariance matrix of the feature point Cov_{u_y} .

$$\text{Cov}_{u_y} = \frac{1}{N} (u_{xy} - \bar{u}_y) \sum_{i=1}^N (u_{xy} - \bar{u}_y)(u_{xy} - \bar{u}_y)^T. \quad (24)$$

According to the calculation of the above formula, we can get the local gray information characteristics of each model feature point: the average gray information model \bar{u}_y and the covariance matrix Cov_{u_y} , including $2k + 1$ model feature points.

Mahalanobis distance D_M can describe the degree of similarity between a local feature point and the calibration feature point (similarity measure): the smaller the distance value, the greater the similarity; the greater the distance value, the smaller the similarity. Therefore, in the target image search process, by calculating the corresponding Mahalanobis distance D_M , the feature point with the greatest degree of similarity is obtained, which is the best matching point of the model calibration feature point.

$$D_M = \sqrt{(S_y - \bar{u}_y)^T \text{Cov}_{u_y}^{-1} (S_y - \bar{u}_y)}. \quad (25)$$

4.4. Matching Target Search Process

4.4.1. ASM Algorithm Single Resolution Search Strategy. When searching for a target in an image, the first step is to use artificially marked feature points to obtain the initial

shape model, then use the local gray-scale texture model to select the best matching point, and combine the limitations and changes of shape parameters to obtain a matching result map. The specific process is as follows:

- (1) The model is initialized and positioned. In the target image, whether there are apple seedlings, flower buds, and fruits at various stages of the growth process, generally determine the approximate position first; on this basis, match the seedling, flower bud, and fruit stage shapes of the initial shape model to obtain the approximate position of the target image; the initial shape is X .
- (2) Discover new model shapes. Search in the vicinity of the feature point according to the normal direction of the feature, and calculate the minimum Mahalanobis distance $D_M = \sqrt{(S_y - \bar{u}_y)^T \text{Cov}_{u_y}^{-1} (S_y - \bar{u}_y)}$ between the current feature point and the candidate feature point in the current contour neighborhood; that is, the best matching point of the current marked feature point is the minimum Mahalanobis distance D_M ; then, the best matching point corresponding to each feature point of the current shape is calculated in turn, and the new shape vectors X' are established.
- (3) The shape parameters are updated. The displacement required by the new model to the characteristic target point is obtained by affine transformation.
- (4) Shape parameter constraints. From the probability distribution model, the variation range of the shape parameter is $-2.98\sqrt{\varepsilon_n} \leq m \leq 2.98\sqrt{\varepsilon_n}$, and finally the result is matched with the target contour in the image.
- (5) Repeat the steps from (2) to (4) until the new shape and the original shape no longer change significantly; we consider that the algorithm converges and the iteration ends.

4.4.2. ASM Algorithm Multiresolution Search Strategy.

When the ASM searches for the best matching point, it is often affected by internal and external factors, such as the initial target location and the noise of the image itself. In turn, deviations occur when each feature point is matched with the best candidate point corresponding to it, and the search performance is changed. The result of the search becomes worse as the number of deviation points increases or the deviation increases, which in turn affects the matching effect of the ASM algorithm. The traditional ASM algorithm faces the following contradictions: In the process of feature point matching, from one perspective, the expected search range is large enough to ensure that the best matching position of the feature point can be searched. However, if the search interval is too large, the number of nonoptimal feature points similar to ideal feature matching points in the search interval will increase, the computational efficiency of the algorithm will be greatly reduced, and the possibility of errors will continue to increase. From another perspective, in general, the smaller search interval is expected to reduce the amount of calculation and finally improve the efficiency.

In response to the abovementioned contradictions, Cootes and other research scholars proposed a new ASM search method, namely, multiresolution search strategy. The basic idea is “from low to high” multilevel search: The first step is to search for the approximate position of the target in the lowest resolution image. Then, in the image environment with the resolution gradually increasing, the precise position of the target is continuously improved until the original image that is the highest resolution image is found. This method can speed up the running speed and reduce the probability that the local structure of the image interferes with the shape model.

In the process of multiresolution search strategy, the first step is to build a pyramid of Gaussian images. The images in each layer are the same, but the difference between them is different resolutions. The lowest layer which is the original image is image layer 0, and the upper layer of lower resolution image is image layer 1. The method of acquisition is to process the lowest layer image with a 5×5 image mask and take a sample of two pixels; that is, the length and width scale of the image obtained in each layer is half of the previous layer; then, get the pyramid of the n -layer image. In this way, a pyramid of Gaussian images at different resolutions is established.

5. Experiment Design and Result Analysis

5.1. Image Segmentation Experiment Results: k Value Selection. In order to determine the appropriate threshold in the HI color segmentation of the Gaussian distribution model to accurately segment the apples in the growth process of each period, this paper selected 45 different weather conditions (sunny, cloudy, and rainy), different environments, and different growth stages. The apple image of the period was used as a sample for testing. At the same time, the relationship between the selection of k value and the segmentation result is discussed. As shown in Figure 1 (the periods are marked with orange numbers), the six images are apple in the seedling stage, flower bud differentiation stage, apple flowering stage, physiological fruit drop stage, fruit expansion stage, and mature stage. The results show that the setting of k value has a great influence on the results of apple image segmentation. When $k = 1$, the lack of apples is serious, and some pixels belonging to apples are wrongly classified as background, as shown in Figure 2. When $k = 4$, most of the pixels belonging to the background are wrongly classified as apple flower buds, apple flowers, or their fruits, as shown in Figure 3. After a lot of experimental verification, it is found that the threshold k segmentation effect is the best within $[2.4, 2.6]$, which meets the goal of correctly extracting apples. Figure 4 shows that when $k = 2.5$, the effect of image segmentation is much better than when the k values are 1 and 4, respectively. The k value at this time shows that the HI color segmentation algorithm of the Gaussian distribution model extracts the apples in the growth process of each period. It adapts to changes in different environmental conditions and has good robustness. Therefore, this paper sets the threshold k to 2.5 as the parameter of the HI color segmentation method of the

Gaussian distribution model to segment the apple image during each period of growth.

In order to further prove the excellent performance of the HI color segmentation algorithm of the Gaussian distribution model in extracting apples in the growth process of each period, this paper again selects typical data from the obtained apple image data: different weather conditions (sunny, cloudy, and rainy), different growth stages, and different levels of complexity are used as test samples.

In this study, the experimental results of the four algorithms of CIVE, ExG–ExR, ExG, and VEG are compared with the experimental results of the method in this paper. The threshold of the HI segmentation algorithm of the Gaussian distribution model is set to $k = 2.5$. It can be seen from the experimental results that the HI algorithm has the best segmentation effect and can adapt to changes in lighting and complex environments. Compared with the three algorithms of CIVE, ExG–ExR, and VEG, the ExG algorithm has better adaptability and can adapt to general light intensity changes, but the image segmentation effect is inferior to the HI algorithm. The VEG algorithm has the worst segmentation effect and is most susceptible to the influence of different light intensity; the noisy point is serious. The ExG–ExR algorithm has a high error rate. The CIVE algorithm can handle the segmentation of green crops and soil better, but when there are interference factors (such as shadows) in the image, the performance is not good. In summary, the HI algorithm selected in this article has the best effect.

5.2. The Positioning Results of Each Period in the Apple Growth Process. The ASM algorithm is used to accurately describe the overall two-dimensional shape of the apple during the growth process, and the multiresolution search strategy is used to determine the position of the apple during the growth process to identify it. The training samples are shown in Figure 1. In order to overcome the shortcomings of traditional methods, this paper directly describes the overall outline of each stage of apple growth. The specific operation is to mark the contours in each image. As shown in Figure 5, the green points are the marked contour points.

In order to improve the accuracy and efficiency of recognition at each period of the apple growth process, we should pay attention to the following issues. The outline needs to be marked in order, and the calibration points should be distributed as far as possible on the edge of the contour. Use these ordered calibration points to describe the contour, and then you can convert them from the image to the two-dimensional space and record it to describe the overall contour.

Although the contour points for calibration at each period of apple growth process are marked in order, there are still certain errors. The description of the overall geometric shape will still be affected by three aspects:

- (1) When obtaining the contour calibration points at each period of apple growth process, the original calibration points are in their respective coordinate



FIGURE 1: Each period of apple growth.

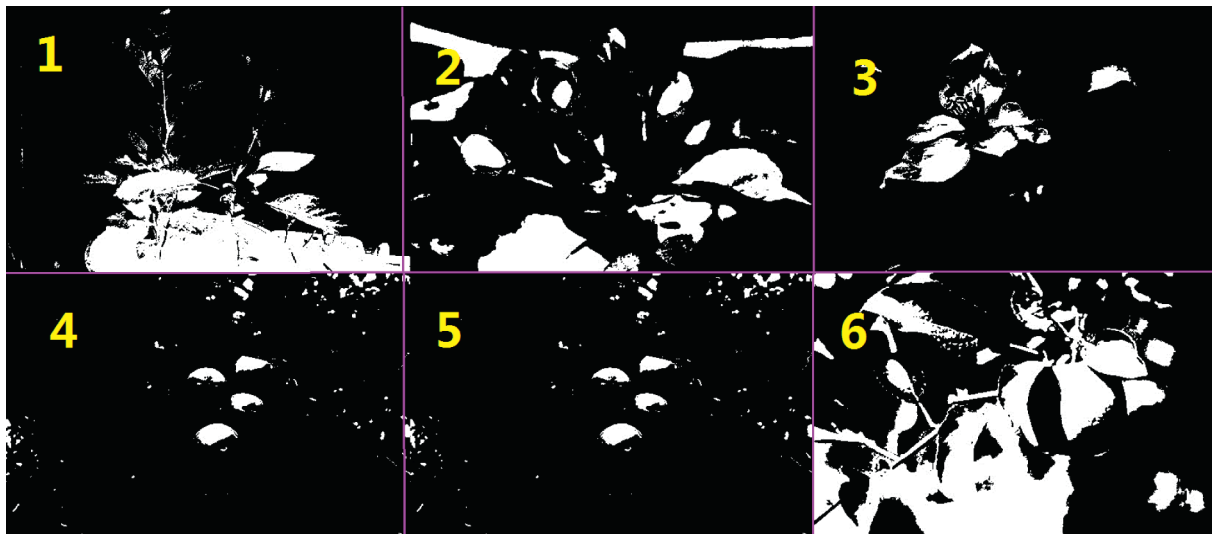


FIGURE 2: HI segmentation result of Gaussian distribution model when $k=1$.

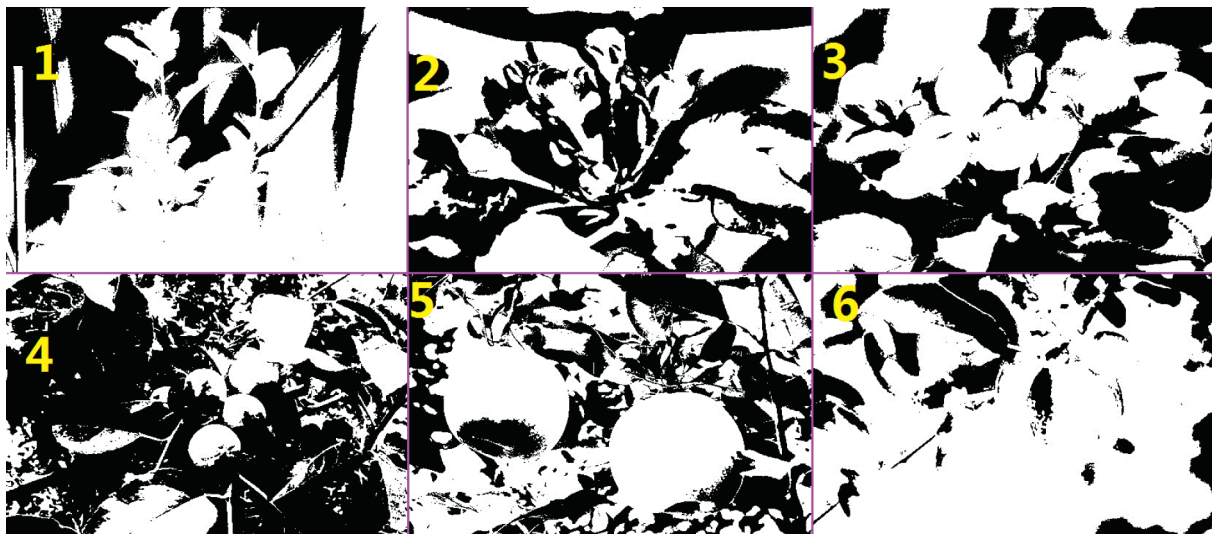


FIGURE 3: HI segmentation result of Gaussian distribution model when $k=4$.

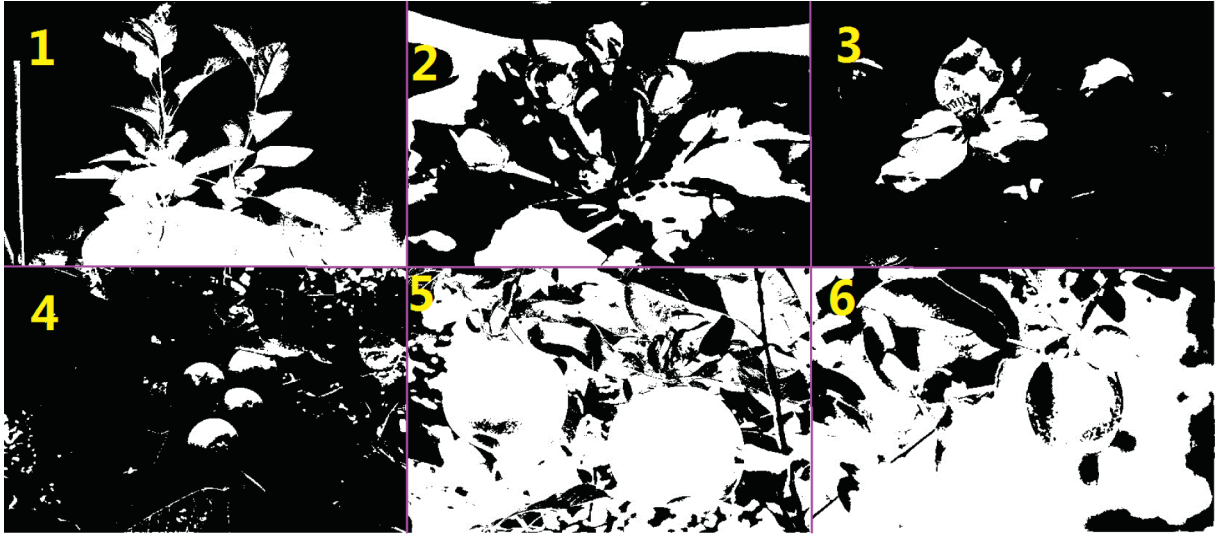


FIGURE 4: HI segmentation result of Gaussian distribution model when $k=2.5$.



FIGURE 5: Outline of markings at each period of apple growth.

systems. Therefore, the apples of the same stage and different periods have different coordinate systems, and there will be differences when they are unified to the same coordinate system.

- (2) When the data is collected, the distance between the camera and the apple is different, which causes the scales of apples in different stages and different periods to be inconsistent in the image, so they cannot be directly compared.
- (3) The apple and the imaging plane are not completely parallel, which causes distortion in the image of the geometry of the apple at each stage of the growth process. In order to overcome the influence of the abovementioned unfavorable factors, it is necessary to align the sequence points of the apple contour calibration at each stage in different images. The main purpose of alignment is to be able to compare

the same marked feature points of training samples from different coordinate systems and build a statistical shape model on this basis to objectively describe the trend and law of shape changes.

After the calibration points of the training samples are aligned, the average sequence calibration points of the training samples are obtained according to (17). The final shape model is denoted by $a = \bar{a} + Vq$. The initial model is used to search for the matching target, and the best matching effect is achieved by limiting the shape weight parameter q .

In this paper, we select 100 apple trees in the orchard and segment the HI image based on the Gaussian distribution model. The optimal k value for image segmentation in each period of apple growth is 2.5, and the weight parameter $q=0.03$ in the shape model.

The success rate of apple identification and positioning at each period of the growth process is shown in Figure 6.

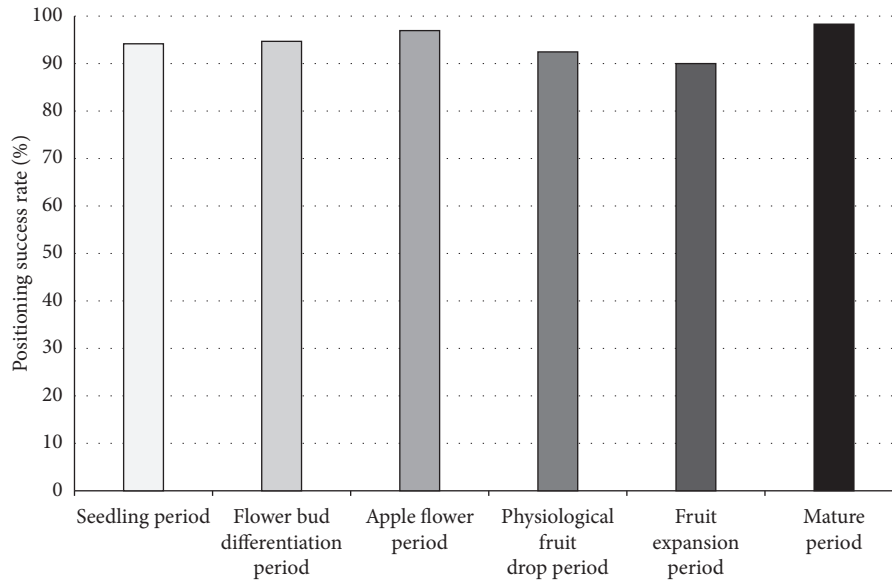


FIGURE 6: The positioning success rate of each period of apple growth.

6. Conclusions

Based on pattern recognition, this paper studies an HI color segmentation algorithm based on Gaussian distribution, which is suitable for apple image segmentation at each period of the growth process in the complex environment of the orchard. It is based on prior knowledge to segment the apple image. First, the apple image with the background removed under different weather conditions is obtained as the training sample, the HI look-up table is established, and then each pixel in the image is judged on whether it belongs to an apple. Morphological processing and contour tracking are performed on the binary image of the segmented apple image, and the appropriate threshold is determined in the HI color segmentation of the Gaussian distribution model to accurately segment the apples in the growth process of each period. On the basis of correct segmentation, the method based on pattern recognition, ASM, is used to identify the apples in the growth process of each period; the overall outline is calibrated to describe their shape; and then the training samples are counted and calculated. Finally, the initial shape model is obtained. This paper shows, through a series of experimental results, that our algorithms are effective and feasible; they can meet actual production needs, provide decision support for real-time management of orchards, and finally lay the foundation for the application of smart agriculture. Although this paper has achieved some satisfactory results, it also has shortcomings. Using the Gaussian distribution HI color segmentation algorithm to segment the apple image depends on the selection of the k value. If the k value is selected incorrectly, it will have a great impact on the segmentation result. Therefore, a clustering method should be added on this basis to achieve a better segmentation effect; in addition, the initial positioning of the ASM algorithm should be improved to make the search matching effect better.

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 (no. 31860331), YEFICRC Project of Yunnan Provincial Key Programs (no. 2019ZG009), and Yunnan Agricultural Joint Youth Fund (no. 2018FG001-091).

References

- [1] S. Z. Knezevic, S. P. Evans, E. E. Blankenship, R. C. Van Acker, and J. L. Lindquist, "Critical period for weed control: the concept and data analysis," *Weed Science*, vol. 50, pp. 773–786, 2009.
- [2] L. Quanqi, D. Baodi, Q. Yunzhou, L. Mengyu, and Z. Jiwang, "Root growth, available soil water, and water-use efficiency of winter wheat under different irrigation regimes applied at different growth stages in north China," *Agricultural Water Management*, vol. 97, pp. 1676–1682, 2010.
- [3] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Computers and Electronics in Agriculture*, vol. 127, pp. 311–323, 2016.
- [4] A. C. Tyagi, "Towards a second green revolution," *Irrigation and Drainage*, vol. 65, no. 4, pp. 388–389, 2016.
- [5] Q. Wang, S. Nuske, M. Bergerman, and S. Singh, "Automated crop yield estimation for apple orchards," in *Proceedings of Experimental Robotics*, vol. 88, pp. 745–758, Quebec, Canada, June 2013.

- [6] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, "Improved image processing—based crop detection using Kalman filtering and the Hungarian algorithm," *Computers and Electronics in Agriculture*, vol. 148, pp. 37–44, 2018.
- [7] J. Lu and N. Sang, "Detecting citrus fruits and occlusion recovery under natural illumination conditions," *Computers and Electronics in Agriculture*, vol. 110, pp. 121–130, 2015.
- [8] R. Linker, O. Cohen, and A. Naor, "Determination of the number of green apples in RGB images recorded in orchards," *Computers and Electronics in Agriculture*, vol. 81, pp. 45–57, 2012.
- [9] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [10] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognition*, vol. 71, pp. 1–13, 2017.
- [11] J. Tang, D. Wang, Z. Zhang, L. He, J. Xin, and Y. Xu, "Weed identification based on K-means feature learning combined with convolutional neural network," *Computers and Electronics in Agriculture*, vol. 135, pp. 63–70, 2017.
- [12] Y. Zhang, P. Phillips, S. Wang, G. Ji, J. Yang, and J. Wu, "Fruit classification by biogeography-based optimization and feed-forward neural network," *Expert Systems*, vol. 33, no. 3, pp. 239–253, 2016.
- [13] J. I. Arribas, G. V. Sánchez-Ferrero, G. Ruiz-Ruiz, and J. Gómez-Gil, "Leaf classification in sunflower crops by computer vision and neural networks," *Computers and Electronics in Agriculture*, vol. 78, pp. 9–18, 2011.
- [14] P. A. Dias, A. Tabb, and H. Medeiros, "Apple flower detection using deep convolutional networks," *Computers in Industry*, vol. 99, pp. 17–28, 2018.
- [15] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proceedings of Automation and Robotics*, pp. 1–8, Singapore, September 2016.
- [16] K. Yamamoto, W. Guo, Y. Yoshioka, and S. Ninomiya, "On plant detection of intact tomato fruits using image analysis and machine learning methods," *Sensors*, vol. 14, no. 7, pp. 12191–12206, 2014.
- [17] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors (Basel, Switzerland)*, vol. 17, p. 905, 2017.
- [18] Y. D. Zhang, Z. Dong, X. Chen et al., "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools Applications*, vol. 78, pp. 1–20, 2017.
- [19] S. W. Chen, S. S. Shivakumar, S. Dcunha et al., "Counting apples and oranges with deep learning: a data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, 2017.
- [20] M. Dyrmann, R. N. Jørgensen, and H. S. Midtiby, "Robo-WeedSupport—detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network," *Advances in Animal Biosciences*, vol. 8, no. 2, pp. 842–847, 2017.
- [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, realtime object detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [22] S. Inkyu, Z. Ge, D. Feras, U. Ben, P. Tristan, and M. C. Chris, "DeepFruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, p. 1222, 2016.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [24] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.
- [25] X.-m. Pang, Z.-j. Min, and J.-m. Kan, "Color image segmentation based on HSI and LAB color space," *Journal of Guangxi University (Natural Science Edition)*, vol. 36, no. 6, pp. 976–980, 2011.
- [26] Y. Sun, Y. Wang, H. Yang, Q. Zhou, A. Sun, and W. Yang, "Prediction of SPAD in rice leaf based on RGB and HSI color space," *Acta Agriculturae Zhejiangensis*, vol. 30, no. 10, pp. 1782–1789, 2018.
- [27] Y.-f. Meng, X.-b. Zhang, and X.-h. Yang, "Application of feature extraction and matching based on ASM in OCR," *Journal of Guangxi University*, vol. 42, no. 6, pp. 2183–2190, 2017.
- [28] Y. Yang, L.-l. He, Y. Shang, and Z. Li, "ASM reduction for test generation," *Computer Engineering and Science*, vol. 40, no. 6, pp. 1084–1092, 2018.
- [29] L.-X. Zhang, H.-P. Zhao, and Y. Xin, "Facial expression recognition based on refined unified model," *Computer Simulation*, vol. 29, no. 1, pp. 227–230, 2012.
- [30] S. Liu, X. Cai, L. Zhang, W. Bi, and J. Liang, "Face sharp classification based on active shape model and K nearest neighbor algorithm," *Journal of Computer Engineering*, vol. 37, no. 6, pp. 479–483, 2014.
- [31] X. Wang, B. I. Xiu-li, M. A. Jian-feng, and B. Xiao, "Nonlinear filtering algorithm based on local texture direction probability statistic model," *Journal of Computer Applications*, vol. 26, no. 12, pp. 2829–2831, 2006.

Research Article

Efficient English Translation Method and Analysis Based on the Hybrid Neural Network

Chuncheng Wang 

Tongling University, Tongling 244061, China

Correspondence should be addressed to Chuncheng Wang; chad825@tlu.edu.cn

Received 29 March 2021; Accepted 4 May 2021; Published 15 May 2021

Academic Editor: Jianhui Lv

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

Neural machine translation has been widely concerned in recent years. The traditional sequential neural network framework of English translation has obvious disadvantages because of its poor ability to capture long-distance information, and the current improved framework, such as the recurrent neural network, still cannot solve this problem very well. In this paper, we propose a hybrid neural network that combines the convolutional neural network (CNN) and long short-term memory (LSTM) and introduce the attention mechanism based on the encoder-decoder structure to improve the translation accuracy, especially for long sentences. In the experiment, this model is implemented based on TensorFlow, and the results show that the BLEU value of the proposed method is obviously improved compared with the traditional machine learning model, which proves the effectiveness of our method in English-Chinese translation.

1. Introduction

The machine translation is an approach that uses computers to automatically convert different languages, which is an important research field of natural language processing (NLP) and artificial intelligence (AI) [1]. It is also one of the most common services on the internet. Although it is still challenging to make the translation quality of machine translation reach the level of professional translators, machine translation has obvious advantages in translation speed in some cases, such as the translation quality is not concerned significantly, or in specific domain translation tasks [2, 3]. In the view of the complexity and application of machine translation, this field is regarded as a key research direction, and it has become one of the most active research fields in natural language processing.

It should be mentioned that neural machine translation (NMT) is the most popular machine translation method that loads specific algorithms into the neural model framework and uses the node-to-node method to optimize the model [4]. At present, with the conditions of large-scale corpus and

computing power, NMT has shown great potential and has developed into a new machine translation method. This method only requires bilingual parallel corpus and is convenient for training large-scale translation models. It not only has high research value but also has strong industrialization capabilities. In many language pairs, NMT has gradually surpassed phrase statistical machine translation. Junczys-Dowmunt et al. [5] used the United Nations Parallel Corpus v 1.0 to compare NMT and phrase statistical machine translation on 30 language pairs. It can be found that NMT surpassed phrase statistical machine translation on 27 language pairs. In addition, on tasks related to Chinese, such as Chinese-English translation tasks, NMT has 6–9 higher BLEU (Bilingual Evaluation Understudy) values. At the Workshop on Machine Translation (WMT) in 2016 [6], the NMT system developed by the University of Edinburgh surpassed phrase-based and syntactic-based statistical machine translation in English-to-German translation tasks. Furthermore, in the industry, Google Translate has adopted NMT instead of statistical machine translation in some languages to provide external services [7]. The well-known

commercial machine translation company SYSTRAN has also developed a corresponding NMT system, covering 12 languages and 32 language pairs.

NMT has an advantage that it can use neural networks to achieve direct translation from the source language to the target language [8]. This translation idea can be traced back to the 1990s; some scholars used small-scale corpus to implement a neural network-based translation method. Due to the limitation of corpus resources and computing power, it did not receive corresponding attention. After the rise of the deep learning boom, neural networks are often used for statistical machine translation language models, word alignment, translation rule extraction, and so on. Until 2013, Blunsom and Kalchbrenner reposed the neural network-based translation method which shows great application potential [9]. Subsequently, Sutskever, Cho, Jean, and others, respectively, realized the corresponding machine translation model based on the neural network [10]. These are classic NMT models, which are essentially sequence-to-sequence models. They can be used not only for machine translation but also for question-answering systems, text summarization, and other natural language processing tasks. Different from the discrete representation method of statistical machine translation, NMT uses a continuous space representation method to represent words, phrases, and sentences. In translation modeling, the necessary steps of statistical machine translation such as word alignment and translation rule extraction are not required, and the neural network is used to complete the mapping from the source language to the target language [11]. Additionally, the other is called the encoder-decoder model, in which the encoder reads the source language sentence and encodes it into a vector with fixed dimensions, and the decoder reads the vector and sequentially generates the target language word sequence. The encoder-decoder model is a general framework that can be implemented by different neural networks, such as long short-term memory (LSTM) neural networks [12] and gated recurrent neural networks (GRNNs) [13]. NMT has been verified that its translation effect is close to or equal to the phrase-based statistical machine translation method. It also has great advantages in some fine-grained evaluation indicators of translations. For example, focusing on the English-to-German translation evaluation task in the 2015 International Workshop on Spoken Language Translation (IWSLT), Bentivogli et al. made a detailed comparative analysis of the translations of phrase statistical machine translation and NMT [14]. As a result, in the neural machine translation, morphological errors were reduced by 19%, vocabulary errors were reduced by 17%, and word ordering errors were reduced by 50%. Among the word order errors, the verb order errors were reduced by 70%.

The current mainstream of NMT is combining with the encoder-decoder structure, and the algorithm is connected through the attention mechanism between the encoder and the decoder [1]. However, NMT based on the structure of the encoder and decoder is a general model, which is not specifically designed for the machine translation task. This leads to some problems including two aspects: firstly, although neural machine translation is a major improvement

of the attention mechanism, its disadvantage is that historical attention information is not considered when generating the target language words, and the constraint mechanism is weak. Furthermore, in some cases, it is not necessary to pay too much attention to the source language information when generating the target language words. For example, when generating the function word “The” in Chinese-English translation, much attention should be paid to the relevant information of the target language [15]. In addition to the above, overtranslation and undertranslation problems may occur in NMT, and it is also necessary to improve the existing attention mechanism [16, 17]. In summary, the attention mechanism optimization is a hot and difficult point in NMT research.

It should be noted that the attention mechanism is a classic neural machine translation model, which improves the representation of the source language and generates source language-related information dynamically in the decoding process to improve the translation effect [18]. Attention-based NMT encodes the source language sentence into a vector sequence instead of a fixed vector. When generating the target language word, it can use the source language word information related to the generation of the word, and the corresponding word can be in the source language. The bilingual vocabulary correspondence realized by the attention mechanism is called soft alignment. Compared with the hard alignment method of statistical machine translation, this method does not limit the alignment length of the target language words and the source language words and can avoid the air-to-air problem in the hard alignment method [19]. However, the attention mechanism has the problem of a large amount of calculation. In order to reduce the amount of calculation, Xu et al. [20] divided attention into soft attention and hard attention on the task of image description generation. The former refers to assigning weights to all regions of the original image, and the amount of calculation is relatively large. And the latter refers to only paying attention to part of the original image area, which can reduce the computational complexity. Based on the above ideas, Luong et al. [21] proposed a local attention model, which is an improvement on global attention. When calculating the context vector c_b , global attention needs to consider all the coding sequences of the source language. Local attention only needs to focus on a small context window in the source language coding, which can significantly reduce the complexity of computation. The core of this method is to find an alignment position related to the generated word from the source language. Local attention only focuses on a small part of the source language when generating the context vector and filters out irrelevant information, which is suitable for long sentence translation. In the WMT 2014 English to German translation, local attention increased by 0.9 BLEU value compared to global attention. In the long sentence translation experiment, as the sentence length increases, the local attention method does not reduce the translation quality. In addition, on the English-German word alignment corpus of the Aachen University of Technology, the local attention word alignment error rate was 34%, and the global attention word alignment

error rate was 39%. In particular, the supervised attention mechanism is the hot topic in this field that uses high-quality prior word alignment knowledge to guide the attention mechanism. Liu et al. proposed a method of using statistical machine translation word alignment information as a priori knowledge to guide the attention mechanism [22]. This method uses GIZA++ to obtain the word alignment information of the training corpus, and then, in the model training, statistical machine translation word alignment is used as a priori knowledge to guide the attention mechanism so that the attention-based word alignment is possible for statistical machine translation. Finally, no prior word alignment information is needed during the test. The experiment uses the Chinese-English machine translation evaluation corpus organized by the National Institute of Standards and Technology (NIST) in 2008. Compared with the attention-based neural machine translation, this method improves the BLEU value by 2.2. On the Tsinghua word alignment corpus, the word alignment error rate of GIZA++ is 30.6%, the word alignment error rate based on attention neural machine translation is 50.6%, and the word alignment error rate of this method is 43.3%. It can be seen that there is a supervision mechanism that can significantly improve the word alignment quality of the attention mechanism, but there is still a big gap compared with the statistical machine translation word alignment, and the attention mechanism still needs improvement.

Aiming at the above disadvantages of the traditional encoder-decoder model framework, this paper proposes an English-Chinese translation model based on a hybrid neural network and an improved attention mechanism. The main idea of the method is to combine the attention mechanism with the neural network to train the local attention of the translation model. Compared with traditional machine learning methods such as least square support vector machine (LSSVM) and extreme learning machine, deep learning methods, i.e., LSTM and convolutional neural network (CNN), have more powerful learning capabilities and good approximation capabilities for the text data in processing regression problems. Therefore, this paper mixes these two networks to improve the ability of the translation model to connect to the context, thereby improving the translation quality of the model.

The rest of this paper is organized as follows. Section 2 presents the detail of the encoder-decoder structure model, CNN, RNN, and attention mechanism. Section 3 presents the hybrid neural network with an improved attention mechanism proposed in our work. Experimental results and discussion are reported in Section 4. Finally, the conclusion of this paper is given in Section 5.

2. Materials and Methods

2.1. Encoder-Decoder Structure Model. The encoder-decoder structure designed in this paper is the core part of the machine translation model, which is composed of an

encoder and decoder. The encoder transforms the input data of the neural network into a fixed length of data. The decoder reversely decodes the data and then outputs the translated sentences, which is also the basic idea of the sequence model. The main process is shown in Figure 1.

The model of encoder-decoder consists of three parts: the input x , the hidden state h , and the output y . The encoder reads the input $x = (x_1, x_2, \dots, x_i)$ and transforms the code into the hidden state $h = (h_1, h_2, \dots, h_i)$ when adopting the RNN:

$$\begin{aligned} h_i &= f(x_i, h_{i-1}), \\ c &= q(\{h_1, \dots, h_i\}), \end{aligned} \quad (1)$$

where c is the sentence representation in the source language and f and q are nonlinear functions. The decoder can generate target language words with the given source language representation c and the precursor output sequence $\{y_1, \dots, y_{t-1}\}$; the definition is as follows:

$$p(y) = \prod_{t=1}^T p(y_t | \{y_1, \dots, y_{t-1}\}, c), \quad (2)$$

where $y = (y_1, y_2, \dots, y_T)$, and when using the RNN,

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c). \quad (3)$$

In this model, g is a nonlinear function that is used to calculate the probability of y_t , and s_t is the hidden state of the RNN, $s_t = f(s_{t-1}, y_{t-1}, c)$. The encoder and decoder can be trained jointly in the following form:

$$L(\theta) = \max_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(y_n | x_n). \quad (4)$$

(x_n, y_n) is the bilingual sentence pair, and θ is a parameter of the model, which can be calculated by the gradient descent method.

2.2. Convolutional Neural Network (CNN). CNN is a special deep learning neural network, which is often used to process data with known grid topology [20]. It is widely used in time series analysis, computer vision, and NLP. According to different data streams, the CNN can be divided into one-dimensional convolution, two-dimensional convolution, and three-dimensional convolution. One-dimensional convolution is widely used in time series analysis and natural language processing. The CNN structure adopted in this paper belongs to the one-dimensional convolution neural network [20], as shown in Figure 2.

Each kind of CNN consists of an input layer and output layer and the core operation part, i.e., convolution layer, pooling layer, and full connection layer. In one-dimensional convolution, the function of convolution can be understood as extracting the translation features of the data in a certain direction. In our work, the essence of the convolution

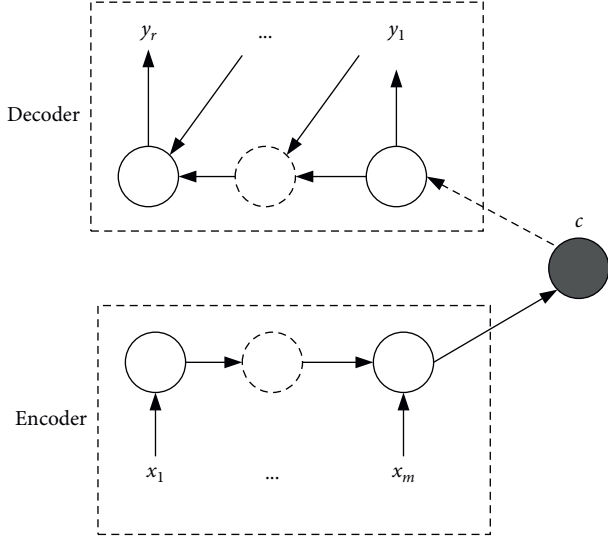


FIGURE 1: Encoder-decoder translation flowchart.

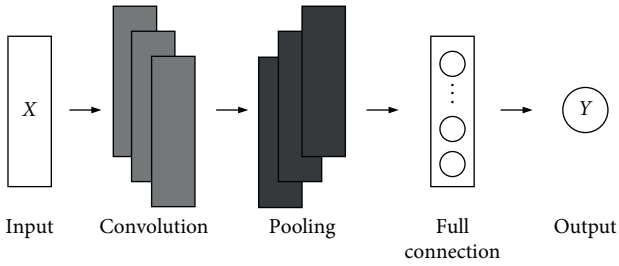


FIGURE 2: One-dimensional convolutional neural network.

operation is cyclic product and addition, and its mathematical expression is as follows:

$$y(k) = h(k) * u(k) = \sum_{i=0}^N h(k-i)u(i), \quad (5)$$

where y , h , and u are time series, k denotes the convolution number, and N is the length of u .

The basic architecture of the general CNN encoder is shown in Figure 3, and its fixed architecture consists of the following six layers:

Layer 0: input layer that uses the embedded vector form of words and sets the maximum length of the sentence to 40 words. For the sentences shorter than this vector, zero padding is often placed at the beginning of the sentence.

Layer 1: the convolution layer after layer 0, whose window size is 3. The boot signal is injected into the layer as the “boot version.”

Layer 2: the local gating layer after the first layer, and it only makes the weighted sum of the feature graph for the nonadjacent window of size 2.

Layer 3: the convolution layer after layer 2, which performs another convolution, and its window size is 3.

Layer 4: this layer performs global gating on the function diagram on layer 3.

Layer 5: fully connected weight, which maps the output of layer 4 to this layer as the final representation.

As is shown in Figure 3, convolution in layer 1 moves on the sliding window of the word, and a similar definition of the window continues to a higher level. Formally, for the source sentence input $x = \{x_1, \dots, x_N\}$, the convolution unit of F -type feature mapping on layer L is shown in the following equation:

$$z_i^{(l,f)}(x) = \sigma(\omega^{(l,f)} \hat{z}_i^{l-1} + b^{(l,f)}), \quad l = 1, 3, \quad f = 1, 2, \dots, F_l, \quad (6)$$

where $z_i^{(l,f)}(x)$ gives the output of position i in layer L whose feature map type is f ; $\omega^{(l,f)}$ is the parameter of f in layer L ; $\sigma(\cdot)$ is the activation function of sigmoid; and \hat{z}_i^{l-1} denotes the convolution segmentation at position i of layer 1, and $\hat{z}_i^0 = \{x_i^T, x_{i+1}^T, x_{i+2}^T\}$ to connect the vector of 3 words from the sentence input.

2.3. Recurrent Neural Network (RNN). The encoder-decoder framework is a part of the neural network, and it is necessary to build an appropriate neural network model to run the framework. RNN is the most widely used among many neural network models, which is a variant model of the feedforward neural network. Its main feature is to process different length data series. Figure 4 shows the structure of the RNN, in which the network has recursive property, and the state of each time has a great relationship with the previous activation state. In this figure, $x = \{x_1, x_2, \dots, x_T\}$ denotes the variable length sequence data, and at each time point t , the hidden state h_t is updated by the following formula:

$$h_t = f(h_{t-1}, x_t), \quad (7)$$

where f is a nonlinear function. We map the input x , and y is the target sequence of the model which is usually given by the training corpus. L is the loss function; U is the weight matrix input to the hidden layer; W is the weight matrix from the hidden layer to the hidden layer; V is the weight matrix from the hidden layer to the output, and t is the weight matrix from the hidden layer to the output, which has the range $[1, T]$. The whole network is updated as follows:

$$\begin{aligned} a_t &= Wh_{t-1} + Ux_t + b, \\ h_t &= \tanh(a_t), \\ o_t &= Vh_t + c, \\ \hat{y}_t &= \text{soft max}(o_t). \end{aligned} \quad (8)$$

The RNN makes different length sequences having the input vectors with the same dimension and the same transformation function, and parameters can be used at each time point, which are more suitable for processing variable length sequence data. In addition, the loop structure can capture all the precursor states in theory, which solves the problem of long-distance dependence to a certain extent.

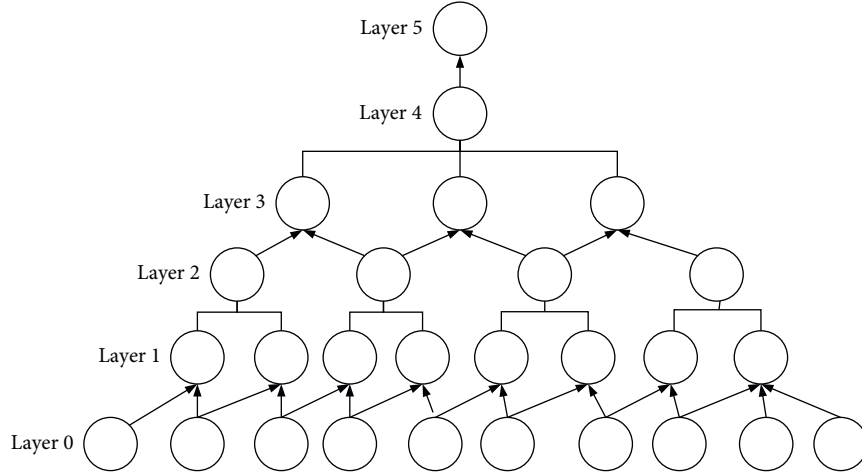


FIGURE 3: CNN coding illustration.

RNN processes variable length sequences by circulating hidden state units. However, gradient vanishing and gradient explosion may occur in RNN training because it is difficult for the RNN to capture the long-term dependence of the data. Thus, long-term and short-term memory (LSTM) network is proposed to solve the problem of RNN gradient disappearing [23]. LSTM is a kind of gated recurrent neural network, which is a special form of the RNN and can capture the long-term dependence between data.

There are three gates in the LSTM cell structure: forgetting gate, input gate, and output gate, and the structure is shown in Figure 5. In LSTM, long-term memory or forgetting information is realized through the input gate, forgetting gate, output gate, and memory unit. If the current time is t , the current input state information and the output value of the current LSTM are the memory unit state of the previous time. The calculation formula of the LSTM cell is shown in equations (9)–(14):

$$\Gamma_f = \sigma(W_f[h^{(t-1)}, x^{(t)}] + b_f), \quad (9)$$

$$\Gamma_i = \sigma(W_i[h^{(t-1)}, x^{(t)}] + b_i), \quad (10)$$

$$\Gamma_o = \sigma(W_o[h^{(t-1)}, x^{(t)}] + b_o), \quad (11)$$

$$\hat{s}^{(t)} = \tanh(W_s[h^{(t-1)}, x^{(t)}] + b_s), \quad (12)$$

$$s^{(t)} = \Gamma_f s^{(t-1)} + \Gamma_i \hat{s}^{(t)}, \quad (13)$$

$$h^{(t)} = \Gamma_o \tanh(s^{(t)}), \quad (14)$$

where W is the weight vector for each gate; b is the bias vector; σ is the sigmoid function; and \tanh is a nonlinear activation function.

Furthermore, gated recurrent unit (GRU) is the variant of LSTM, which has simpler memory units. This structure combines the input gate and forgetting gate of the long- and short-term memory cycle into the update gate and then

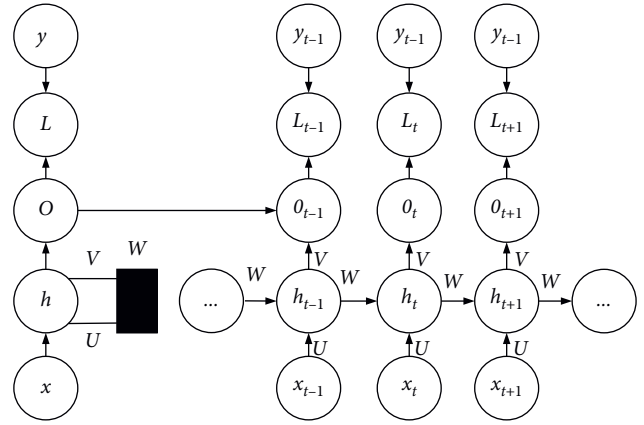


FIGURE 4: Structure diagram of the RNN.

introduces the reset gate. It uses the update gate to control the history information and new information to be forgotten in the current state and uses the reset gate to control the information quantity obtained from historical information in the candidate state. As shown in Figure 6, the GRU memory unit has only two gates: reset door and update door. The reset gate r_t controls the degree of status information of the previous moment, and the update gate z_t determines the quantity of the memory reserved in front. The simple memory unit of the GRU makes its parameters less than LSTM, and its performance is equivalent to or even better than LSTM. The calculation formula of the reset gate and update gate is as follows:

$$\begin{cases} z_t = \sigma(W_z[h_{t-1}, x_t] + b_z), \\ r_t = \sigma(W_r[h_{t-1}, x_t] + b_r), \\ \bar{h}_t = \tanh(W_a[r_t * h_{t-1}, x_t] + b_a), \\ h_t = (1 - z_t) * h_{t-1} + z_t * \bar{h}_t, \end{cases} \quad (15)$$

where W_z , W_r , and W_a are weight matrices and b_z , b_r , and b_a are deviation vectors.

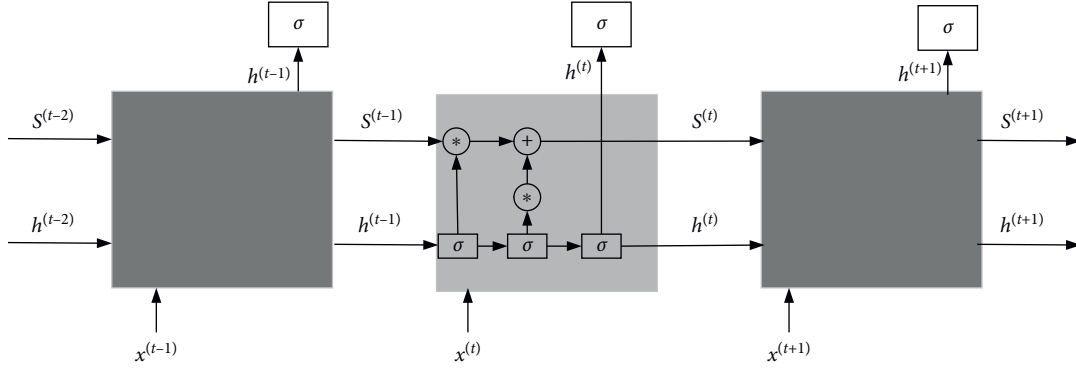


FIGURE 5: LSTM structure.

2.4. RNN with the Attention Mechanism. When the human brain observes an object, it often focuses on some parts, and these parts are also the key to obtain information from things. This information has a strong guiding role in cognition of similar things, and the attention mechanism is designed to imitate this cognitive process. The application of the attention mechanism in computer vision and natural language processing has achieved good results. This paper applies the attention mechanism to the analysis of text series.

In the analysis of the text sequence, the CNN is used to extract the spatial features of the sequence. As too many or nonkey features will affect the final prediction results after LSTM is used to extract the spatial and temporal features, the attention mechanism is used to extract the key features of the sequence. The attention mechanism is similar to a weighted summator or a key feature extractor, which mainly performs the weighted summation operation. The attention model proposed in this paper is shown in Figure 7, and the vector c is the key feature to be extracted, as is shown in the following equation:

$$c = \sum_{i=1}^m \beta_i v_i, \quad (16)$$

where m is the time step sum of the input of the LSTM network; v is the output eigenvector of the LSTM network; and β is the weight of vector v . To obtain the weight β , we add a small neural network a to the attention model, and the output layer activation function is softmax. The calculation is shown as follows:

$$\beta_i = \frac{\exp(e_i)}{\sum_{k=1}^m \exp(e_k)}, \quad (17)$$

where e_i can be computed in $e_i = a(v_i) = \sigma(Wv_i + b)$ and σ is the sigmoid function. W is the weight matrix from the input layer to the hidden layer in Figure 7, and b is the offset value matrix.

3. Proposed Method

In order to reduce the gradient problem caused by the long data sequence, a popular method is to combine the RNN with encoder-decoder. However, the operational performance of the encoder-decoder framework is limited to a

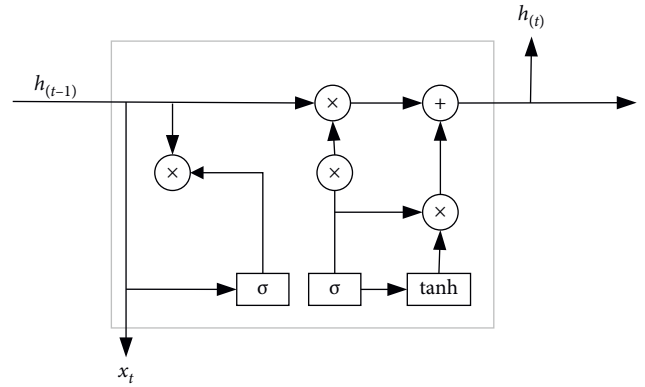


FIGURE 6: GRU structure.

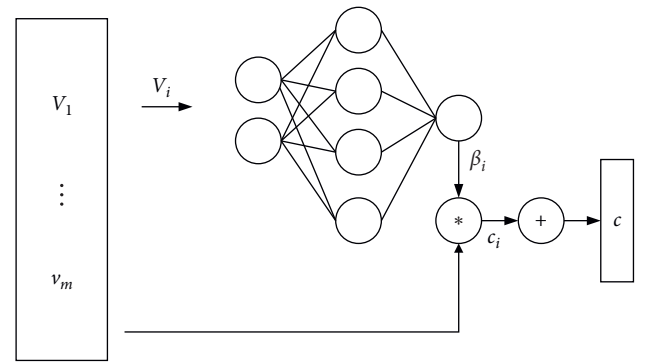


FIGURE 7: Attention mechanism model.

certain extent, which results in that the long input data have high computational complexity. The attention mechanism can solve the problem of limited operation by establishing the decoder and sentence segment transmission channel, i.e., the decoder can return to view the input data at any time. In this way, the intermediate data can be omitted; thus, the operation performance will be improved, and the translation accuracy will be improved.

The left-right sequential operation mechanism of the RNN may lead to the limitation of the parallel operation ability of the model and data module loss. The attention mechanism is helpful to solve the above problems because the data distance of any position in the translation data can

be changed to 1, which improves the parallelism of the model but depends on the influence of the previous sequence operation no longer.

The main process of the attention mechanism consists of four steps: (1) weighting the input data of the neural network and importing them into the encoder; (2) importing the data into the decoder; (3) the decoder queries the data weight in the decoding process as the reverse input data; (4) computing the weighted average value of the data in each state. The simplified implementation process of the attention mechanism is shown in Figure 8.

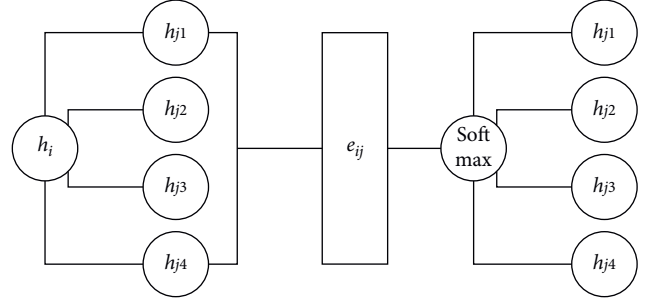


FIGURE 8: Schematic diagram of the attention mechanism.

3.1. Model Framework. In order to highlight the advantages of the attention mechanism, this paper introduces the attention mechanism into the model framework of the RNN and realizes the translation task by establishing the encoder-decoder framework. The neural connection part of the model is realized by the attention mechanism, which is helpful to take advantages of the attention mechanism.

Figure 9 shows the attention mechanism model constructed in this paper, whose overall structure is composed of the encoder and decoder. The encoder is composed of a single-layer structure and a single-layer structure of the precoding network, and the block number is N_c . The structure of the decoder is similar to that of the encoder, which is also composed of N_c blocks, but has no head attention layer. The neural network in this paper uses differential network connection, and the remarkable feature of this approach is that the network has entered the standard level for data processing.

3.2. Attention Mechanism Module. The attention mechanism module is mainly divided into the encoder module and decoder module. The input part of the encoder module is the whole data sequence, and three input matrices are used in this part, i.e., Q , K , and V . The attention mechanism function can be regarded as a mapping relationship as follows:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (18)$$

The calculation process of the attention mechanism is as follows:

- (1) The sentence data to be translated are weighted by three different matrices, i.e., Q , K , and V , and each sentence will get the three vectors
- (2) The weight distribution of the above three matrices is calculated by scaling the dot product to get a numerical value
- (3) Take the weight value in Step (2) as the activation function
- (4) Multiply the output of Step (3) with the V matrix to get the final result

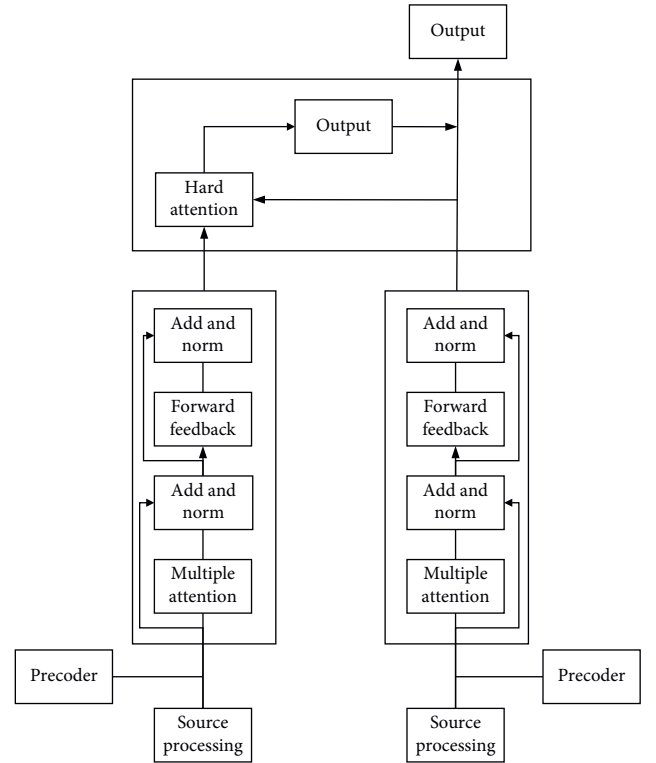


FIGURE 9: Model frame structure.

4. Experiment and Analysis

All the experiments are carried out on the Alibaba cloud server ECS, whose CPU type is Intel Skylake Xeon Platinum 8163 with 2.5 GHz, and the memory is 8 GB. All the program codes are written in Python (version 3.7.7) in TensorFlow [24].

4.1. Data Acquisition and Evaluation Methods.

- (1) Data acquisition: the training data of this paper are extracted from LDC data [25]. Only the part of the source pair which is less than 40 words in length is reserved, which covers more than 90% of the sentences. The bilingual training data consist of 221k

sentence pairs, including 5 million Chinese words and 6.8 million English words. After length-limited filtering, the development set is NIST MT03, which contains 795 sentences, and the test set is MT04 and MT05, which contain 1499 sentences and 917 sentences, respectively.

- (2) Preprocessing: use Giza++ to achieve word alignment in two directions with the “growth-determiner-final” and “balance strategy” on the corpus [23]. The improved Kneser–Ney smoothing 4-gram language model is trained on the new Chinese part of English Gigaword corpus, which contains 306 million words, by using the SRI language modeling toolkit. Then, the Chinese sentence is parsed into a mapping dependency tree using Stanford Parser.
- (3) Neural network optimization: when training the neural network, the source words and target words are limited to the most common 20k words in Chinese and English, covering 97% and 99% of the words in the two corpora, respectively. All vocabularies and words are mapped to the special token UNK. Random gradient descent is used to train the joint model, and the minimum batch size is set to 500. All joint models use 3-word target (4-gram LM). The final representation of the CNN encoder is a vector with a size of 100. The final DNN layer of the joint model is a standard multilayer perceptron, and the top layer has softmax.
- (4) Evaluation: in order to evaluate the prediction error, BLEU value is used as the evaluation index as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^N w_n \log P_n\right), \quad (19)$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r, \\ e^{1-r/c} & \text{if } c \leq r. \end{cases}$$

It should be noted that the BLEU value is calculated for a translation sample.

4.2. Experimental Test and Result Analysis. The steps of the experiment are as follows:

- (1) The corpus is processed to cut long sentences into words
- (2) The words are numbered and stored as files, and the files are stored in a PC
- (3) Normalizing the text, making up for the insufficient sentence length, and intercepting the excessive sentence length
- (4) Training the processed sentences, and then the BLEU value is evaluated

In the experiment, the comparison test is used to evaluate the error of single LSTM [26], LSSVM [27], CNN

[28], CNN-LSTM [29] without the attention mechanism, LSTM with the attention mechanism [30], and the proposed model. The comparison results are shown in Table 1.

It can be seen from Table 1 that LSTM’s translation effect has been significantly improved by 2.8 and 7.36 BLEU after combining the CNN and attention mechanism with the same settings. Furthermore, LSTM is higher than CNN’s average values of 2.5 and 0.49 BLEU in MT04 and MT05, respectively. The results indicate that LSTM and attention mechanism can provide discriminative information in decoding. It is worth noting that CNN + LSTM is more informative than LSSVM. Therefore, it is speculated that this is due to the following two facts:

- (1) CNN + LSTM avoids the spread of errors and pseudo-effects in the learned word alignment
- (2) Guidance signals in CNN + LSTM provide supplementary information to evaluate translation

In addition, the reason CNN can get high gain on BLEU is that it encodes the whole sentence, and the representation should be far away from the best representation of the joint language model. Therefore, as the CNN has a very useful summary of the sentence, the losses in the resolution and the related parts of the source words can be made up.

In other words, the pilot signals in LSTM and attention mechanism are important for the function of the CNN-based encoder, which can be seen from the difference between BLEU values obtained by CNN and LSSVM. CNN + LSTM can further benefit from the dependency structure encoded in the source language in the input. The dependency initial can be used to further improve the CNN + LSTM model. In LSTM, a marker bit (0 or 1) is added to the words embedded in the input layer as a marker whether they belong to the source word or not. In order to merge the dependency header information, we extend the tagging rule by adding another marker bit (0 or 1) to the original LSTM word embedding to indicate whether it is part of the dependency header of the adjunct. For example, if x_i is the embedding of the related source word and x_j is the dependency header of word x_i , the extended input of LSTM will contain as the following equation:

$$\begin{cases} x_i^{(\text{AFF,NON-HEAD})} = [x_i^T, 1, 0]^T, \\ x_j^{(\text{NON-AFF,HEAD})} = [x_j^T, 0, 1]^T. \end{cases} \quad (20)$$

The second part of the experiment is to cluster the sentences in the corpus according to the length and then test the translation model with different length sentences to verify the ability of the translation model. The test results are shown in Table 2. It can be seen from Table 2 that the performance of the proposed model is still the best in the case of different sentence length, which proves that this model also has strong translation performance in the aspect of long sentence translation. Moreover, with the increase of sentence length, the decline of the translation performance of this model is smaller than that of the traditional model, which indicates that it is less sensitive to the change of sentence length and more accurate.

TABLE 1: Comparison of the method with published studies.

| Model | MT04/BLEU | MT05/BLEU | Average/BLEU |
|------------------------------------|-----------|-----------|--------------|
| LSTM | 15.62 | 14.81 | 15.21 |
| LSSVM | 17.76 | 16.89 | 17.32 |
| CNN | 13.12 | 14.32 | 13.72 |
| CNN + LSTM | 18.99 | 17.03 | 18.01 |
| LSTM + ATT | 21.69 | 23.44 | 22.57 |
| Proposed method (CNN + LSTM + ATT) | 24.33 | 25.97 | 25.15 |

TABLE 2: Comparison of the method with published studies.

| Model | MT04/BLEU | MT05/BLEU | Average/BLEU |
|------------------------------------|-----------|-----------|--------------|
| LSTM | 13.31 | 12.51 | 12.91 |
| LSSVM | 15.21 | 14.35 | 14.78 |
| CNN | 11.06 | 11.53 | 11.3 |
| CNN + LSTM | 17.22 | 14.97 | 16.10 |
| LSTM + ATT | 20.01 | 21.23 | 20.62 |
| Proposed method (CNN + LSTM + ATT) | 22.16 | 24.83 | 23.50 |

5. Conclusions

Aiming at the problems of inaccurate translation and incomplete semantics in the traditional encoder-decoder translation model, we propose a hybrid neural network that combines CNN and LSTM and introduce the attention mechanism. This model can improve the performance of the context semantic connection and parallel operation and then effectively improve the translation quality of long sentences. The experimental results show that the translation performance of the improved hybrid neural network translation model is significantly better than that of the traditional translation model. In the long sentence translation test, this model also has the best performance.

In future, we will study the new English translation scenario based on the large-scale regions. In addition, an English translation system platform will be implemented.

Data Availability

The data used to support the findings of this study are available upon request to the author.

Conflicts of Interest

The author declares that there are no conflicts of interest.


References

- [1] L. Lemao, T. Watanabe, E. Sumita et al., "Additive neural networks for statistical machine translation," in *Proceedings of the International Conference on Parallel & Distributed Systems*, Seoul, Korea, December 1997.
- [2] D. Xiong, Z. Min, and H. Li, "Enhancing language models in statistical machine translation with backward N-grams and mutual information triggers," in *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, June 2011.
- [3] P. Li, Y. Liu, and M. Sun, "Recursive autoencoders for ITG-based translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, NJ, USA, October 2013.
- [4] L. Deng, M. Seltzer, D. Yu et al., "Binary coding of speech spectrograms using a deep auto-encoder," *Interspeech*, 2010.
- [5] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is neural machine translation ready for deployment? A case study on 30 translation directions," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, USA, October 2016.
- [6] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," in *Proceedings of the First Conference on Machine Translation*, Berlin, Germany, August 2016.
- [7] Y. Wu, M. Schuster, Z. Chen et al., "Google's neural machine translation system: bridging the gap between human and machine translation," 2016, <https://arxiv.org/abs/1609.08144>.
- [8] Z. Jie, C. Ying, X. Wang et al., "Deep recurrent models with fast-forward connections for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 4, no. 2, 2016.
- [9] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, NJ, USA, October 2013.
- [10] J. Brea, W. Senn, and J. P. Pfister, "Sequence learning with hidden units in spiking neural networks," in *Proceedings of the Part of Advances in Neural Information Processing Systems 24 (NIPS 2011)*, Granada, Spain, December 2011.
- [11] Z. Huang, M. Cmejrek, and B. Zhou, "Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, USA, October 2010.
- [12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, December 2014.
- [13] K. Cho, B. V. Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, <https://arxiv.org/abs/1406.1078>.

- [14] L. Bentivogli, A. Bisazza, M. Cettolo et al., “Neural versus phrase-based machine translation quality: a case study,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, November 2016.
- [15] Z. Tu, Z. Lu, L. Yang et al., “Modeling coverage for neural machine translation,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.
- [16] G. Druck, K. Ganchev, and J. V Graça, “Rich prior knowledge in learning for natural language processing,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Portland, OR, USA, June 2011.
- [17] Z. Tu, Y. Liu, L. Shang et al., “Neural machine translation with reconstruction,” 2016, <https://arxiv.org/abs/1611.01874>.
- [18] C. Yong, X. Wei, Z. He et al., “Semi-supervised learning for neural machine translation,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.
- [19] C. T. Chung, C. Y. Tsai, H. H. Lu et al., “An iterative deep learning framework for unsupervised discovery of speech features and linguistic units with applications on spoken term detection,” 2016, <https://arxiv.org/abs/1602.00426>.
- [20] K. Xu, J. Ba, R. Kiros et al., “Show, attend and tell: neural image caption generation with visual attention,” *Computer Science*, vol. 37, pp. 2048–2057, 2015.
- [21] M. T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015, <https://arxiv.org/abs/1508.04025>.
- [22] L. Liu, M. Utiyama, A. Finch et al., “Neural machine translation with supervised attention,” in *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, December 2016.
- [23] S. Maskey and B. Zhou, “Unsupervised deep belief features for speech translation,” *Interspeech*, 2012.
- [24] L. Hao, S. Liang, J. Ye, and Z. Xu, “TensorD: a tensor decomposition library in TensorFlow,” *Neurocomputing*, vol. 318, pp. 196–200, 2018.
- [25] Y. Bo, R. Liu, and D. He, “Research on remote sensing image classification based on improved decision tree classification algorithm,” *Computer Measurement & Control*, vol. 41, 2018.
- [26] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM fully convolutional networks for time series classification,” *IEEE Access*, vol. 6, no. 99, pp. 1662–1669, 2018.
- [27] Y. Zhai, X. Ding, X. Jin et al., “Adaptive LSSVM based iterative prediction method for NOx concentration prediction in coal-fired power plant considering system delay,” *Applied Soft Computing*, vol. 89, 2020.
- [28] X. Zhou, Y. Yang, H. E. University et al., “Study on evaluation of tense accuracy in CNN-based google translation from English to Chinese,” *Modern Electronics Technique*, 2019.
- [29] S. Gundapu and R. Mamidi, “Multichannel LSTM-CNN for Telugu technical domain identification,” 2021, <https://arxiv.org/abs/2102.12179>.
- [30] Y. Wang, M. Huang, X. Zhu et al., “Attention-based LSTM for aspect-level sentiment classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, November 2016.

Research Article

Artificial Intelligence-Based Joint Movement Estimation Method for Football Players in Sports Training

Bin Zhang,¹ Ming Lyu ,² Lei Zhang,¹ and Yang Wu¹

¹Anhui Normal University, Wuhu 241008, China

²Anhui Technical College of Mechanical and Electrical Engineering, Wuhu 241002, China

Correspondence should be addressed to Ming Lyu; 0127000110@ahcme.edu.cn

Received 22 March 2021; Revised 13 April 2021; Accepted 26 April 2021; Published 4 May 2021

Academic Editor: Jianhui Lv

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

Football is a product in the process of human socialization; it can strengthen the body and enhance the ability of teamwork. The introduction of artificial intelligence into football training is an inevitable trend; this trend must be bound to intensify, but how to apply artificial intelligence to solve the problem of the joint movement estimation method for football players in sports training is still the main difficulty now. The basic principle of football training action pattern recognition is to determine the type of football player's action by processing and analyzing the movement information obtained by the sensor. Due to the complex movements towards football players and the changeable external environment, there are still many problems with action recognition. Focusing on the detailed classification of different sports modes, this article conducts research on the recognition of the joint movement estimation method for football players in sports training. This paper uses the recognition algorithm based on the multilayer decision tree recognizer to identify the joint movement; the experiment shows that the method used in this paper accurately identified joint movement for football players in sports training.

1. Introduction

Football movement pattern recognition is an emerging human-computer interaction method. According to the estimation method, football action pattern estimation can be divided into vision-based estimation technology, tactile-based estimation technology, and sensor-based estimation technology. Among these technologies, vision-based football movement estimation is to complete the classification and estimation of football movement patterns by analyzing the image or video of the active target [1, 2]. Although the technology of this type of method is relatively mature, it has more stringent requirements for the test environment, such as good light and no objects between the moving target and the video sensor. Tactile-based movement pattern recognition limits the range of activities of the active target and requires a relatively high-cost touch screen device. Compared with the above two methods, sensor-based football joint movement estimation only

requires the tester to wear a compact sensor, which has almost no limitation on the range of motion and is less disturbed by the external environment. These obvious advantages make sensor-based estimation methods gain more and more attention.

In recent years, the technology of micro-electromechanical systems has become more and more mature; their size has become smaller and cheaper; their sensitivity has increased day by day. At present, micromotor systems are widely used in mobile devices, game consoles, and other products, bringing users a brand-new human-computer interaction experience and a superior user experience. In these microelectromechanical systems, inertial measurement units (IMUs) embedded with three-axis gyroscopes and three-dimensional acceleration sensors can simultaneously obtain the angular velocity and acceleration of the measured object in the three-dimensional space, thereby providing rich and comprehensive data source. This provides a powerful data guarantee for sensor-based

recognition and recognition of joint actions in football player training.

The main contributions to this paper are summarized as follows. (i) We use the recognition algorithm based on the multilayer decision tree recognizer to identify joint movement. (ii) We use Weka to generate the traditional SVM classifier and the traditional decision tree recognizer.

The rest of the paper is structured as follows. Section 2 introduces the Kalman algorithm based on data pre-processing. In Section 3, data feature extraction and selection is presented and analyzed. Section 4 introduces the recognition algorithm based on the single decision tree identifier and proposes the recognition algorithm based on the multilayer decision tree recognizer. The experimental results are reported in Section 5, and finally, Section 6 concludes this paper.

2. Kalman Algorithm Based on Data Preprocessing

2.1. Kalman Filter Algorithm. In the 1960s, Kalman first proposed the Kalman filter [3, 4], which can optimally estimate the system state based on the system state equation and a series of measurement variables with null values and containing noise, so as to achieve the purpose of noise reduction and filtering. The Kalman filter algorithm is a fast and efficient recursive estimation algorithm. Its basic calculation principle is to obtain the estimated value of the current state through the measured value of the current state and the estimated value of the previous state. The recursive estimation algorithm has a small amount of calculation, less memory, and clear steps; the Kalman filter is very suitable for real-time data processing [5] and is currently widely used in many fields such as communication, pattern recognition, navigation, and automatic control. The Kalman filter is used to estimate the state variable x of a discrete-time process, which can be described by the following difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}. \quad (1)$$

Among them, x is the state variable of the system, k represents the system time, x_k means the system state variable at time k , A is the state transition matrix, B is the gain of the system input u , and w is the excitation noise matrix. The system observation variable is defined as z , and the system's measurement equation is defined as follows:

$$z_k = Hx_k + v_k. \quad (2)$$

Among them, H represents the gain of the state variable against the measurement variable at time k and v is the observation noise matrix. Random signals w_{k-1} and v_k represent process excitation noise and observation noise, respectively; assuming that they are mutually independent, normal distribution of white noise can be obtained as follows:

$$\begin{aligned} p(w) &\in N(0, Q), \\ p(v) &\in N(0, R). \end{aligned} \quad (3)$$

The Kalman filter algorithm has three assumptions [6, 7]: firstly, the current state is a linear function of the previous state and system input, or Gaussian noise can be superimposed; then, the measured value of the state variable must be a linear function of the state variable superimposed by Gaussian noise; finally, the initial state distribution of the system is Gaussian distribution.

Five core equations of the Kalman filter algorithm are given below. These five equations can be divided into time update equations (also called prediction functions) and measurement update equations (also called calibration functions). Among them, the time update equation calculates the a priori estimated value x at the current moment (time k) according to the state estimate of the state variable x at the previous moment (time $k-1$); the measurement update equation comprehensively calculates a priori estimated value of x and the measured value of the current moment to obtain the posterior estimated value of the current moment.

Time update equations are defined as follows:

$$x_{\bar{k}} = Ax_{k-1} + Bu_{k-1}xk - 1, \quad (4)$$

$$P_{\bar{k}} = AP_{k-1}A^T + Q. \quad (5)$$

State update equations are defined as follows:

$$K_k = \frac{P_{\bar{k}}H^T}{R + HP_{\bar{k}}H^T}, \quad (6)$$

$$x_k = x_{\bar{k}} + K(z_k - Hx_{\bar{k}}), \quad (7)$$

$$P_k = (1 - K_kH)P_{\bar{k}}. \quad (8)$$

The core step of the Kalman filter algorithm is to estimate the current state of the system by combining the predicted value and the measured value: equation (4) passes the posterior estimated value x_{k-1} of the previous state and the system input u_{k-1} to obtain the a priori estimated value $x_{\bar{k}}$ of the current state; equation (5) obtains the a priori estimation agreement variance $P_{\bar{k}}$. In equation (6), the Kalman gain matrix K can be obtained by one formula, which is the intermediate calculation result of filtering. Equation (7) is used to obtain the posterior estimated value of x_k of the current state, which is also regarded as one core step of the Kalman filter algorithm. Finally, calculate the posterior covariance P_k of the current state for the next round of iteration.

For example, to estimate the attitude angle of a football player when running in training, the angular velocity sensor device can be used to measure and calculate in real time. But considering the influence of noise and interference on the system, the measured value is not accurate, so the real result cannot be completely dependent on the result collected by the sensor. Since any system state that satisfies the laws of physics should be continuous, the Kalman filter can predict the state at the current moment based on the estimated value of the previous moment. As for whether the result is biased toward the measured value or the estimated value, it is

determined by K , the ‘‘Kalman Gain,’’ which is calculated based on the measured value and the a priori estimated covariance.

The variables that affect the results of the Kalman filter mainly include initial state x_0 , initial state covariance P_0 , state transition matrix A , process excitation noise covariance Q , and measurement noise covariance R . Among them, the values of the initial state variable x_0 and the initial variance matrix P_0 have little effect on the filtering result because, no matter what their values are, the result can quickly converge. Therefore, this article focuses on the three factors of A , Q , and R .

2.1.1. State Transition Matrix A . The state transition matrix describes how the system transitions from the state at time $k - 1$ to the state at time k . The system transition matrix can estimate the state of the system at the next moment. The state transition matrix may be known or unknown.

When the conversion of the system is linear, such as uniformly accelerating motion, at this time, the state transition matrix A is a linear matrix. However, if the estimated process or the relationship between the observed variable and the process is nonlinear, the system does not satisfy the premise of Kalman filtering. In this case, we can consider using the Extended Kalman Filter (EKF), which uses the Jacobian matrix to linearize the expectation and variance, so the application of the Kalman filter can be extended to nonlinear systems. But EKF involves the expansion of Taylor series, so the amount of system calculations will increase.

When the system transition is uncertain, the motion equation of the moving target is usually unknown, so the state transition matrix cannot be determined. At this time, solving the problem generally requires alternative methods, such as using other prediction algorithms to replace the state transition matrix, such as the tracking Kalman filter method based on gray prediction, or using a combination of Kalman filters of multiple known models for hybrid prediction, such as the interactive multimodel method and so on.

2.1.2. Process Excitation Noise Covariance Q . The process excitation noise is the noise component superimposed by the system state variable x in the state transition process. It is the white Gaussian noise with a mean value 0 and a covariance Q .

For some stable processes, it can be considered that the Q value of the system is certain. At this time, it is only necessary to adjust the coefficients of the filter through experiments in order to obtain a better filtering effect. When the state transition is determined, the Q value should be as small as possible, so as to ensure the faster convergence of the filter and reduce the impact on the state variables.

2.1.3. Measurement Noise Covariance R . The measurement noise covariance is a value related to the instrument and a known condition of the filter. If the value R is too large or too small, it is not conducive to the final filtering effect, and the smaller the value R , the faster the convergence. First,

calculate the appropriate R value through experiments, and then, use this value to participate in the Kalman filter.

2.2. Data Preprocessing of Joint Movement Signals for Football Players in Sports Training. Due to the complex and changeable experimental environment, the data collected by the sensor inevitably contains missing data and noise components [8, 9]. If these interference factors are not handled properly, the accuracy and efficiency of pattern recognition will be greatly reduced. Therefore, a series of preprocessing operations must be performed on the original data.

The data preprocessing module based on human movement signals mainly includes the following four parts: removing the null value of the original data, data normalization, Kalman filter, and data windowing.

Various movements of football players are continuous in a short period of time. To judge the joint movements of football players in training, it is necessary to judge by analyzing the data signals for a period of time, and it is necessary to perform windowing processing. Generally speaking, the duration of football player joint movement is about 2 s. Therefore, the length of each window in this article is 2 s. A semi-overlapping sliding window method is used to divide the data. The length of each window is 2 seconds. The data of each window processing unit is 40.

3. Data Feature Extraction and Selection

The information collected by the sensor module is mainly three-axis acceleration and three-axis angular velocity data information, which can be expressed as $\vec{A} = (a_{x,t}, a_{y,t}, a_{z,t})$ and $\vec{\omega} = (\omega_{x,t}, \omega_{y,t}, \omega_{z,t})$, respectively, in order to reduce the complexity of multidimensional acceleration and angular velocity signal calculation, and the coordinate system of the sensor and the natural coordinate system. For the calculation between real-time conversion between the two, this paper uses the corresponding scalar, the amplitude value of the synthesized acceleration signal, and the amplitude value of the synthesized angular velocity signal; the specific calculation formula is as follows:

$$\begin{aligned} A(t) &= \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2}, \\ \omega(t) &= \sqrt{\omega_{x,t}^2 + \omega_{y,t}^2 + \omega_{z,t}^2}xyz. \end{aligned} \quad (9)$$

Among them, $a_{x,t}$, $a_{y,t}$, and $a_{z,t}$ represent the acceleration signal of the x , y , and z three-axis direction changing with time; $\omega_{x,t}$, $\omega_{y,t}$, and $\omega_{z,t}$ represent the angular velocity signal of the x , y , and z three-axis direction changing with time.

Covariance between angular velocity signals: the standard deviation and covariance are defined as follows:

$$\begin{aligned} \text{std} &= \sqrt{\frac{\sum_{i=1}^n (a_i - a_{\text{avg}})^2}{n}}, \\ \text{cov}(a, \omega) &= \frac{\sum_{i=1}^n (a_i - a_{\text{avg}})(\omega_i - \omega_{\text{avg}})}{n - 1}. \end{aligned} \quad (10)$$

Among them, n represents the number of data points in an active window, a_i and ω_i represent the synthetic acceleration and synthetic angular velocity at a certain time point, respectively, and a_{avg} and ω_{avg} , respectively, represent the average value of the synthetic acceleration and synthetic angular velocity in a sliding window.

The value of the synthetic acceleration of the football player during joint movement in training will continue to change. The point with the largest amplitude within a sliding window is the peak value. The peak value indicates the intensity of the exercise. The larger the peak value, the more intense the exercise. On the contrary, the smaller the amplitude value in a sliding window is the trough, the smaller the trough value can also indicate the stronger the signal strength. The standard deviation represents the degree of signal deviation within a period of time; the covariance represents the relationship between the change of the acceleration signal and the change of the angular velocity signal.

4. Recognition Algorithm Based on the Decision Tree Identifier

In recent years, pattern recognition has played an increasingly important role in the field of artificial intelligence; many important results have been achieved. Pattern recognition belongs to a kind of cognitive ability. The goal of the decision tree identifier is to supervise learning and to find a set of eigenvector in a given dataset to describe all elements; each element can be represented by a set of mutually exclusive eigenvector [10]. In other words, its goal is to distinguish a set of elements from different eigenvectors through a specific mapping relationship to form a decision tree identifier model, and this mapping relationship can be applied to unknown samples for classification.

4.1. Recognition Algorithm Based on the Single Decision Tree Identifier. The essential problem of the decision tree algorithm is to select the eigenvector of the corresponding node and the pruning of the decision tree [11]. When using the decision tree model to classify unknown samples, start from the root node and gradually identify them according to the corresponding feature attributes and further identify them according to the corresponding recognition results, until they reach the leaf nodes. The eigenvector of the decision tree algorithm is based on the value of its information gain ratio.

The purpose of the decision tree estimation method is to select the attribute with the largest information gain rate as a set of eigenvector for testing.

In order to prevent the established decision tree and training samples from overfitting and to improve the recognition accuracy and classification accuracy of unknown classes, the decision tree needs to be pruned [12]. Pruning is usually to delete the most unreliable branches through statistical methods to improve the efficiency of the entire system and the accuracy of classification.

Import 2/3 of the collected data as training samples into the Weka environment, and use the decision tree generated

by the decision tree estimation C4.5, and use 1/3 of the data as the sample to be tested to identify and verify its accuracy. The experimental results show that the accuracy rate has reached 95.185%, but the algorithm is more complicated. There are a total of 15 leaf nodes, and many eigenvectors are used repeatedly. The generated decision tree is shown in Figure 1.

In Figure 1, A (min) represents the trough value of the composite acceleration amplitude value; A (max) represents the peak value of the composite acceleration amplitude value; A (std) represents the standard deviation of the composite acceleration amplitude value; W (min) represents the value of the composite angular velocity amplitude value; W (max) represents the peak value of the synthetic angular velocity amplitude value; W (std) represents the standard deviation of the synthetic angular velocity amplitude value; conv represents the covariance between acceleration and angular velocity.

4.2. Recognition Algorithm Based on the Multilayer Decision Tree Identifier. The energy consumption of the data collection terminal in football joint movement recognition is limited, so it is necessary to reduce the energy consumption of the system, while meeting the energy consumption of the system. The energy consumption of the system mainly needs to consider two aspects: sensor data collection terminal and motion recognition terminal. The energy consumed by the sensor data collection terminal mainly includes the energy consumption of the sensor collecting data, the energy consumption of sending data, and the basic energy consumption. The energy consumed by the first two factors accounts for more than 80%. The energy consumed by the sensor terminal is mainly composed of three aspects: data reception, data processing, and recognition algorithms. Therefore, reducing the transmission and processing of one of the sensors' data will also reduce the system energy consumption.

Among the eight types of exercise studied in this article, we can divide them into three categories: static, slow movement, and strenuous exercise. The static movement of the football player can be distinguished from any other motion by any feature vector in the acceleration data information or the angular velocity data information. Slow movements mainly include walking, jogging, and juggling; strenuous exercise includes pass, trapping, tackle, and shoot.

Through Weka's analysis of the corresponding data, it can be obtained that slow action and violent action can be distinguished by the peak and valley value of the synthetic angular velocity and the standard deviation of the synthetic angular velocity. When the standard deviation of the synthetic angular velocity is to distinguish stationary and shooting from other actions, then the trough value of angular velocity is used to distinguish static and jogging, and the recognition rate is 98.44%.

Through the analysis of football player's movements in training, it is known that the average football player spends more than 70% of the time in static and jogging state during training. If the complexity of these two movement

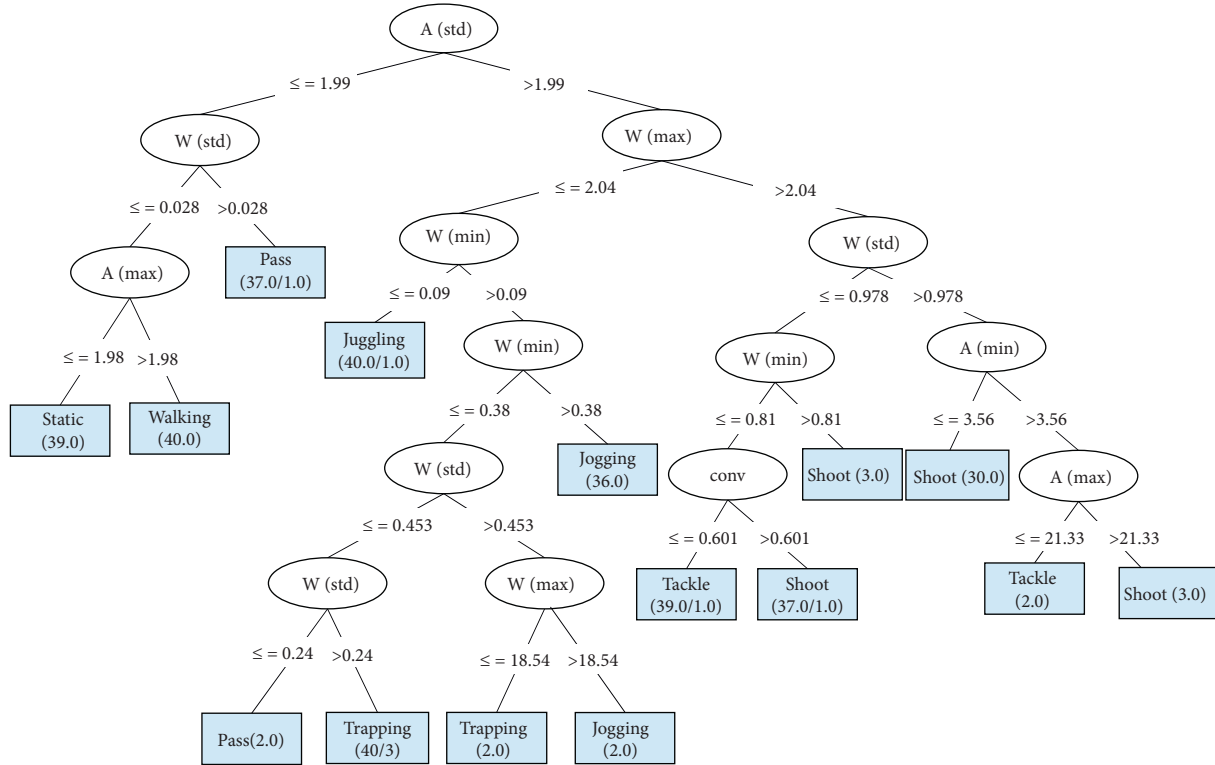


FIGURE 1: Recognizer based on the single decision tree.

recognition algorithms is reduced and the application of sensor data is reduced, then the energy consumption and complexity of the system will be reduced, so a recognizer algorithm based on a two-layer decision tree is proposed.

The specific implementation steps are as follows. First, extract the data of the gyroscope sensor, find the amplitude value of the synthetic angular velocity, calculate the trough value, peak value, and standard deviation of the synthetic angular velocity amplitude value, and use the first-level decision tree recognizer to identify stationary and walking, including other actions; if it is stationary and walking, the recognition is ended; otherwise, the data of the acceleration sensor is extracted; the acceleration amplitude value is calculated, and the peak value, trough value, standard deviation, and covariance of acceleration and angular velocity are obtained, and the second layer recognition is used in the device performs a second detailed identification.

4.3. First-Level Decision Tree Recognizer. In order to ensure the accuracy of the system, while reducing the energy consumption of the system, this paper hopes to achieve this by reducing the transmission of acceleration sensor data and the complexity of the system algorithm. Through experimental analysis, the common static and slow walking of the human body can be distinguished by the data characteristics of the angular velocity sensor, and the decision tree recognizer that is easy to move to the portable terminal is used for recognition. Figure 2 is a decision tree recognizer generated by Weka based on the peak value of the synthetic angular velocity, the trough value of the synthetic angular

velocity, and the standard deviation of the synthetic angular velocity. Its recognition accuracy reaches 98.44%.

4.4. Second-Level Decision Tree Recognizer. If the football player is not in static or walking slowly in training, it is difficult to achieve accurate recognition due to the complex movements and only relying on the time-domain characteristic signal of a single acceleration or the time-domain characteristic signal of the angular velocity. Therefore, it is necessary to further extract the data of the acceleration sensor. The feature vectors used in the second round of identification include the peak value of the synthesized acceleration signal, the bottom value of the synthesized acceleration signal, the standard deviation of the synthesized acceleration signal, and the covariance between acceleration and angular velocity. The recognizer used is still easy to implement and complicated. The decision tree recognizer has low and good real-time performance. Figure 3 shows the generation of the second-level decision tree in Weka.

In Figure 3, W (max) is the maximum value of the synthetic angular velocity, A (std) is the variance of the synthetic acceleration [13, 14], A (max) is the peak value of the synthetic acceleration, A (min) is the valley value of the synthetic acceleration, and conv is the sum of the synthetic acceleration. By using some corresponding feature vectors, other actions that cannot be accurately recognized by the first-level recognizer are further recognized. Specifically, it can recognize walking, jogging, and juggling in tardiness movements and pass, trapping, tackle, and shoot in strenuous exercise.

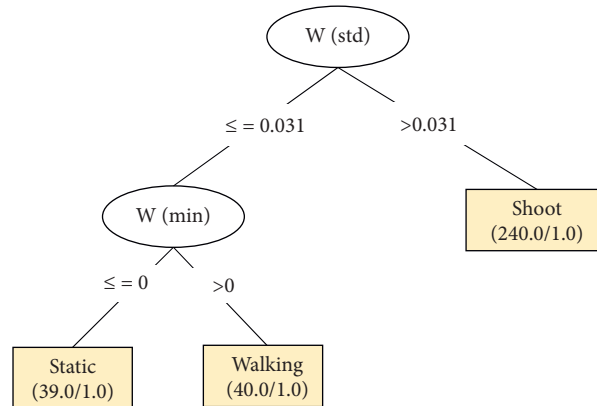


FIGURE 2: First-level decision tree recognizer.

Through the analysis of simulation results, the accuracy of the second-level decision tree's recognizer is 94.17%. From the two decision trees generated by calculation, it can be seen that the two decision trees have 9 leaf nodes, and the common static and walking states only need to pass the first decision tree and use a gyroscope.

5. Experiment Design and Result Analysis

5.1. Data Extraction. In order to train an effective recognizer, a large amount of different types of joint movement data information is needed, but there is no complete database containing various football player joint movement data information. Therefore, this research needs to collect a large number of different joint movement data information of different experimenters first to train the recognizer and verify the accuracy of the recognizer [15].

The wearing position of the sensor data collection terminal will directly affect the accuracy of the recognition. For the different wearing positions of the same motion sensor, the collected sensor data will have large differences. The ideal positions for football players to wear motion sensors are feet, waist, chest, etc.

The advantage of wearing the sensor on the joints of football players' foot is that it is more sensitive to the collection of foot joint movement data and is more sensitive to some movements that use the foot; the disadvantage is that due to the high sensitivity, it will bring more interference information, stability is not very strong, and wearing comfort level is poor.

The sensor module used in this article is the MPU6050 module (including the acceleration sensor and the gyroscope sensor). Before collecting data, the scale of the sensor needs to be set. The ranges of the acceleration sensor and the gyroscope sensor are set to $\pm 16g$ and $\pm 2000/\text{sec}$, and the frequency is 20 Hz.

5.2. Realization of Joint Movement Recognition Algorithm and Analysis of Results. After collecting a large amount of joint movement data, it is necessary to mine and process the corresponding data. In order to process the data more

professionally, this research adopts the special data processing and mining software called Weka [16].

5.2.1. Weka. In order to make better use of the collected sensor data, we use the Waikato Environment for Knowledge Analysis (Waikato Environment for Knowledge Analysis, Weka) software to process and analyze the data. Weka is an open source and free data mining platform based on the JAVA environment. It is a collection of a large number of data mining machine learning methods. This includes a large number of data preprocessing, classification, regression, clustering, association, and other rules.

The main interface of Weka mainly includes six parts, and the six parts also correspond to the corresponding functional modules:

- (1) Data preprocessing: deal with the noise of the data accordingly
- (2) Classify: training and testing-related classification methods
- (3) Cluster: learn clustering from the trained data
- (4) Correlation: learn related association rules from training data
- (5) Feature selection: select data-related attributes
- (6) Visualization: view the association rules of related data

In order to compare the validity and reliability of the recognition algorithm based on the two-layer decision tree recognizer proposed in this paper, we use Weka to generate the traditional SVM classifier [17] and the traditional decision tree recognizer to analyze each recognition performance.

5.2.2. Bayes Classifier. According to the principle of Bayes classifier implementation, this paper uses 2/3 of the collected sensor data to train the Bayes recognizer [18, 19], and the remaining 1/3 of the sensor data is used to verify the accuracy and effect of the generated recognizer. The recognition rate of the Bayesian classifier recognizer is 93.44%, but

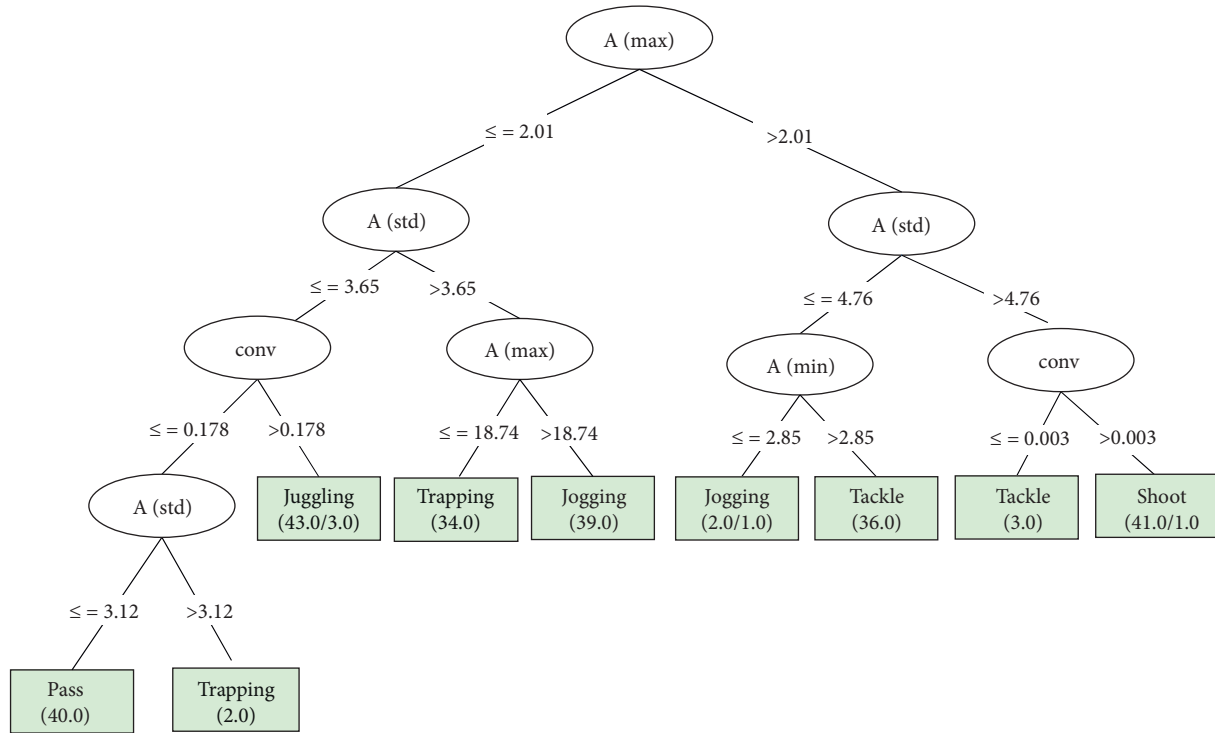


FIGURE 3: Second-level decision tree recognizer.

its algorithm is more complicated. Among them, the recognition rate of strenuous pass, trapping, tackle, and shoot is relatively low.

5.2.3. Support Vector Machine Classifier. According to the principle of the recognizer of the support vector machine, 2/3 of the sensor data collected in this study is used to train the recognizer of the support vector machine, and the remaining 1/3 of the sensor data is used to test the accuracy and effect of the generated recognizer. Due to the limited eigenvectors selected by the system and the limited training samples, its recognition accuracy is only 89.38%, which is low for some more complex movements.

5.2.4. Compare the Effects of Different Recognizers. The traditional decision tree recognizer and the two-layer decision tree recognizer have been introduced in detail above. The coefficient of the weighted average is the weighted average of the coefficients of each category, and the weight is the proportion of the actual category. By comparing the coefficients of the classifiers, it can be found that the recognition effect based on the two-layer decision tree recognizer is relatively ideal, and it is better than a single simple Bayesian network and support vector machine recognizer; from the complexity of the algorithm and the training time, it can be seen that the former is also far superior to the latter; combined with football players' exercise habits and

algorithms, the former needs to collect less than the latter two, so its system energy consumption will be reduced accordingly.

5.3. Results and Discussion. The research objects used in this paper are 20 young male football players, 2 players in a group, divided into 10 groups, aged between 18 and 26, with heights ranging from 155 cm to 180 cm.

Method: the method of collecting the training set is the same as the method mentioned above. It simulates eight actions of a football player static, walking, jogging, juggling, pass, trapping, tackle, and shoot. Each action needs to be repeated five times to upload the data to the PC. As seen from Table 1, the recognition rate is above 95%.

The verification process is to redo these eight actions on the 20 experimenters to see if there are any false positives.

Compared with other recognizers, the recognition rate of the recognition algorithm based on the multilayer decision tree recognizer proposed in this paper is higher than others, as shown in Figure 4.

From experimental results, it can be seen that the false alarm rate (meaning that one action is understood as another action) is low; the highest false alarm rate is the most vigorous tackles and shots, and the false alarms can be turned off directly. Therefore, the system in this study should also follow this principle. In general, the joint movement recognition algorithm for football players in training studied in this paper meets the goals and requirements of the system.

TABLE 1: Recognition rate of joint movement.

| Group no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------------|----|----|----|-----|-----|-----|----|-----|----|-----|
| Correct identify | 78 | 79 | 79 | 80 | 80 | 80 | 77 | 80 | 79 | 80 |
| Unrecognized | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| False positive | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |
| Recognition rate (%) | 98 | 99 | 99 | 100 | 100 | 100 | 96 | 100 | 99 | 100 |

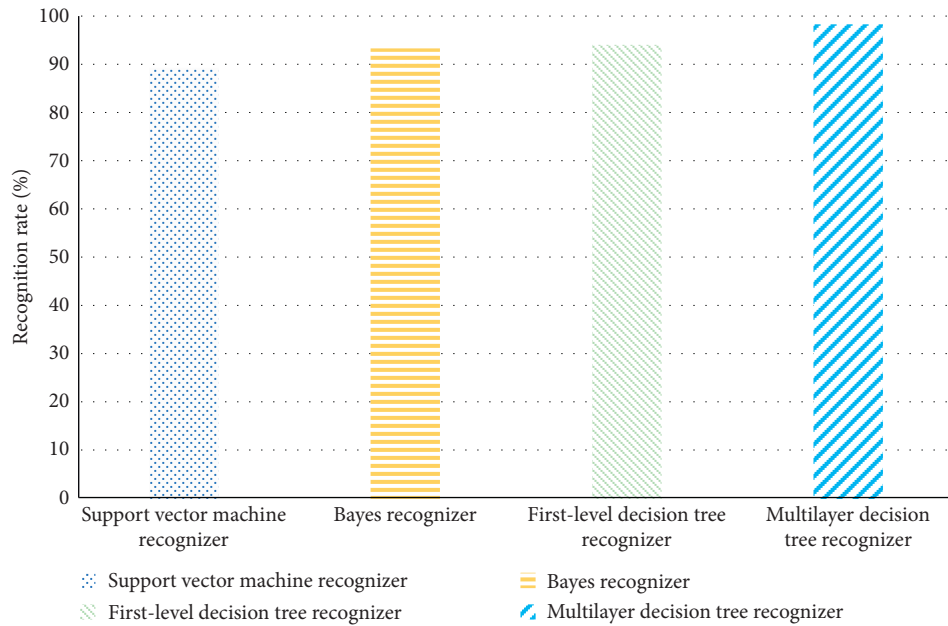


FIGURE 4: Recognition rate of different recognizers.

6. Conclusions

This paper makes an in-depth analysis of the existing pattern recognition system, analyzes the joint movement data of football players in training, and proposes a recognition algorithm based on a multilayer decision tree recognizer. Extract the data firstly, find the amplitude value of the synthetic angular velocity, calculate the peak value, trough value, and standard deviation of the synthetic angular velocity amplitude value, and use the first-level decision tree recognizer to identify stationary, walking, and also other actions; if it is stationary and walking, the recognition is ended; otherwise, the data of the acceleration sensor is extracted, the acceleration amplitude value is calculated, and the trough value, peak value, standard deviation, and covariance of acceleration and angular velocity are obtained, and the second layer recognition is used in the device performs a second detailed identification through experiments to verify that the system can meet the needs of joint movement estimation for football players in training. In this paper, the sensor module is used to collect the data information of the acceleration sensor and angular velocity sensor during the eight kinds of sports of different football players and analyze the joint movement by extracting the corresponding feature vector. In future, we will also consider various complex joint movements to better assist in sports training.

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.

References

- [1] X. Jiao, "Research on key technologies of intrusion detection based on pattern recognition," in *Proceedings of the 2019 7th International Conference on Machinery, Materials and Computing Technology*, pp. 390–393, Aizu-Wakamatsu, Japan, March 2019.
- [2] Z. Fan, S. Yu, and L. Wei, "Object recognition algorithm based on an improved convolutional neural network," *Journal Beijing Institute of Technology*, vol. 29, no. 2, pp. 139–145, 2020.
- [3] G. Shi-Luo, Y.-J. Sun, L.-M. Chang et al., "Robust Cubature Kalman Filter Method for the Nonlinear Alignment of SINS," *Defence Technology*, vol. 17, no. 2, pp. 593–598, 2021.
- [4] X. Sun, C. Wen, and T. Wen, "A novel step-by-step high-order extended kalman filter design for a class of complex systems with multiple," *Basic Multipliers*, vol. 30, no. 2, pp. 313–321, 2021.

- [5] L. Ying, J. Wang, and H. Zhongzhi, "Kalman-filter-group based aero-engine sensors fault diagnosis and verification," in *Proceedings of the 2019 Chinese Intelligent Systems Conference*, pp. 100–106, Singapore, September 2019.
- [6] X. Chenhui, H. Defeng, and S. Xiulan, "Data attack detection of connected vehicles based on adaptive kalman filter," pp. 1–6, 2019.
- [7] L. Ying, C. Lei, and Z. Zhang, "Research and simulation on pilot configuration in multi-antenna system based on kalman filter," in *Proceedings of the International Conference on Mechanical Design (ICMD)*, pp. 371–379, Huzhou, China, August 2019.
- [8] S.-Y. Hwang, L. Jang-Hyun, H. Kyu-Tack et al., "Time-series data analysis by pattern recognition and classification for fault diagnosis of plant equipment," in *World Maritime Technology Conference 2018 (WMTTC18)*, pp. 1–7, Shanghai, China, December 2018.
- [9] D. Ruijuan and C. Shilin, "The pattern recognition application research in image processing," in *2018 5th International Conference on Electrical & Electronics Engineering and Computer Science (ICEECS 2018)*, pp. 214–219, Istanbul, Turkey, May 2018.
- [10] P. Oleksiy, I. Hernandez-Bautista, N. Oscar Camacho et al., "Wavelet filter adjusting for image lossless compression using pattern recognition," in *Proceedings of the Mexican Conference on Pattern Recognition*, pp. 221–230, Springer, Cancun, Mexico, June 2014.
- [11] W. Chen, "Research on group decision making of large-scale engineering based on uncertain decision tree classification algorithms," in *Proceedings of the 2019 4th International Industrial Informatics and Computer Engineering Conference (IIIEEC 2019)*, pp. 383–387, Sanya, China, December 2019.
- [12] Y. Huiizhen, C. Xiaohang, Z. Wang et al., "Research on material identification technology of sealed relay remainder material based on decision tree," in *Proceedings of the 7th International Conference on Reliability of Electrical Products and Electrical Contacts*, pp. 237–243, Suzhou, China, November 2019.
- [13] K. Yong-Joong and C. Sung-Bae, "A HMM-based location prediction framework with location recognizer combining k-nearest neighbor and multiple decision trees," in *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*, pp. 618–628, Salamanca, Spain, September 2013.
- [14] K. Kyoung Min, P. Joon Jo, S. Myung Hyun et al., "Binary decision tree using K-means and genetic algorithm for recognizing defect patterns of cold mill strip," in *Proceedings of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004)*, pp. 341–350, Ottawa, ON, Canada, May 2004.
- [15] X. Wang, J. Zhang, N. Masafumi et al., "Efficient phoneme set design using phonetic decision tree in dialogue-based English CALL systems for Japanese students," *IEICE Technical Report*, vol. 113, no. 366, pp. 47–51, 2013.
- [16] A.-K. Koliopoulos, P. Yiapanis, F. Tekiner et al., "A parallel distributed Weka framework for big data mining using spark," in *Proceedings of the IEEE International Congress on Big Data: Institute of Electrical and Electronics Engineers*, pp. 9–16, New York, NY, USA, July 2015.
- [17] C. Chen and J. Linhai, *Stand Species Identification Using GF-2 Imagery and Airborne LiDAR Data Based on SVM Classifier*, pp. 1–13, IOP Conference Series Earth and Environmental Science, Bristol, UK, 2019.
- [18] F. Qiang, L. Yong-Bo, H.-W. Wang et al., "Aero-engine health monitoring method based on E-bayes and DNN fusion decision," in *Proceedings of the 2020 2nd International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2020)*, pp. 97–104, Wuhan, China, January 2020.
- [19] Z. Tianjiao, W. Yue, and Y. Wang, "College network forum Bayes classification system: implementation and performance evaluation," in *Proceedings of the 12th International Conference on Management of e-Commerce and e-Government (ICMeCG 2018)*, pp. 391–397, Hong kong, China, June 2018.

Research Article

Data Collection and Analysis of Track and Field Athletes' Behavior Based on Edge Computing and Reinforcement Learning

Di Han 

Jilin Agricultural University, Changchun 130118, China

Correspondence should be addressed to Di Han; handi@jlau.edu.cn

Received 18 March 2021; Revised 11 April 2021; Accepted 20 April 2021; Published 30 April 2021

Academic Editor: Jianhui Lv

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

With the development of multimedia technology, the computer auxiliary system has become an effective means of daily training in track and field. This paper designs a data acquisition and analysis system for track and field athletes. The system uses sensor modules attached to the athlete's body to collect movement data for analysis. The whole system is implemented by edge computing architecture. In order to reduce average response time, the DDPG algorithm is used to optimize the resource allocation of the edge layer. Experimental results show that the response time of the proposed algorithm can be controlled within 1 s. Meanwhile, the SVM algorithm on the edge server is arranged to classify the data, and the overall recognition accuracy is over 90%.

1. Introduction

Track and field sports is a large category, which includes a variety of subevents, such as race walking, running, and javelin throw. Athletes need to spend a bunch of time and energy developing skills and promoting endurance. A British research institute conducted a survey on the training time of more than 100 elite athletes. Result suggests that, in order to prepare for the 2012 Olympics in London, they trained six hours a day, six days a week, 12 months a year on average, and some athletes even incredibly spent 10000 hours during 4 years, consuming 1.1 million calories a year on average. For track and field athletes, scientific training methods are crucial to the improvement of their competitive level. The personal experience and teaching skills of the coach play a vital role in the training process. However, with the development of information technology, athletes can be effectively trained via the utilization of big data and software analysis [1].

There are studies on track and field sports auxiliary training system. Cucco [2] developed a system for evaluating and improving the performance characteristics of an athlete, such as speed, agility, and quickness. They took advantage of smart sensors to detect the athlete's relative position with training target and display the results to them. Ma et al. [3]

developed a C/S mode monitoring system on the athlete training process, which was based on mobile artificial intelligence technology. This system used GPS to obtain real-time position information of athletes and provided real-time guidance. Guo [4] designed a VR system to train athletes to master self-balance by detecting foot pressure data. The authors designed the whole system in the form of VR games to stimulate the enthusiasm of sports training.

Inspired by the above related works, a data acquisition and analysis system based on edge computation and reinforcement learning for track and field training is devised in this paper. The wearable sensing module in the system is used to collect activity data or physical sign of athletes and using the data for further analysis. Given that the battery life and memory resource of these devices are limited, the mobile edge computing architecture (MEC) is introduced, which is suitable for solving the problems faced by our scenario. As a complementary technology of cloud computing, MEC can make up for the shortcomings of cloud computing and effectively solve the pressure of mass data brought by the Internet of Things. MEC plays an important role in many application scenarios. For example, Mao et al. [5] studied the tradeoff between the power consumption of mobile devices and the execution delay of computing tasks in a multiuser MEC system. They proposed an online

algorithm based on Lyapunov optimization to determine whether the computation is performed locally or offloaded to the edge nodes. Li et al. [6] focused on QOE-optimized video delivery under the edge computing environment; they proposed an algorithm containing Integer Linear Programming (ILP) formula to solve this problem. Wang et al. [7] used edge computing structure to process social network data. Meanwhile, in order to optimize deployment strategy and maximize economic benefits, they designed a hybrid optimization model ITEM to reduce computing costs.

Edge computing is an extension concept of cloud computing. Its main purpose is to reduce the communication transmission costs between users and computing processing nodes. However, when the number of users increases, the computing requests from multiple users may exceed the capacity of the server, resulting in network congestion. In addition, the performance of the edge server is always fluctuating and changing under the influence of real-time changes on task requests load, power supply, and network conditions, which bring challenges to ensuring the performance of task execution. This situation can be abstractly described as a distributed resource optimization problem, and reinforcement learning is a classic solution to this issue. In 2015, Google DeepMind published a paper in Nature [8], which proposed a model that combined reinforcement learning (RL) and deep learning (DL), named deep reinforcement learning (DRL). Its outstanding performance in the field of game AI soon made DRL a new research focus. Since there are various kinds of tasks in edge computing, how to transfer the computing requests to the appropriate server so as to minimize the cost is a major optimization problem. Using simple models to solve this problem is relatively inefficient. RL algorithm can be used to optimize resource allocation under edge computing environment, which provides users with more efficient service.

The contributions of this work are summarized as follows:

- (1) A data collection and analysis system for track and field sports is devised
- (2) Deep deterministic policy gradient (DDPG) algorithm is used to reduce the service response time via edge computing structure

The rest of the paper is organized as follows: In Section 2, more related works are introduced, including information about the edge computing structure and the DDPG algorithm. Section 3 depicts the main structure of our system. In Section 4, the experiment results of the key performance indicators of the system are reported, and Section 5 gives the conclusion of this paper.

2. Related Works

2.1. Wearable Device-Based Track and Field Auxiliary Training System. With the development of information technology, the utilization of multimedia assists athletes in training. Even the real-time tactics analysis system of the game situation has been well known to people. A computer-aided system for throwing events training has emerged in the

United States since the early 1970s. This system utilized cameras to record the athletes' technical movements; then the records are analyzed to get the angle data of the athletes' hand when throwing. Finally, it compares with the standard data to improve the technical action of throwing items. With the increasing maturity of MEMS, there are more sports assistance systems choosing to collect athletes' activity data and other pieces of relevant information through wearable devices. Thomas et al. [9] developed a training monitoring system for swimming, where multiple accelerometers were placed in the key positions of the human body, and semi-Markov model (SMM) was used in the system to accurately segment and label swimming activities with high accuracy. Valkova et al. [10] asked 75 mentally handicapped athletes to wear a GT3X activity recorder and record their activity data in two days before the competition and concluded that the local Special Olympic program is beneficial for people with a mental disability. Lee and Drake [11] interviewed 20 technical athletes, they combined athletic training and performance with the collection and evaluation of personally relevant data in an effort to better understand their own abilities, and the study also examines the individual relationships that technical athletes have with their data. Wachowicz and Mrozek [12] aligned the activity data collected by smart wearable devices with meteorological data to explore the influence of weather on athletes' on-the-spot performance and fuzzy join technique was used in this work. In a sentence, as smart devices continue to evolve, as more functions are integrated into such auxiliary systems, the system will consequently become overburdened.

2.2. Edge Computing. Edge computing is a form of distributed computing in which the main processing and data storage are placed at the edge nodes of a network. Edge nodes provide services based on the principle of proximity to quickly respond to the requests from the smart devices, so as to meet the basic needs in real-time business, security, privacy protection, and so on. The edge computing layer is at the top of the physical entity. In the MEC structure, a large number of small dedicated servers are deployed on the edge of the network close to the mobile user. Users can offload computation-intensive tasks to the edge cloud close to them and effectively run the application on their own mobile terminal. A common MEC architecture is shown in Figure 1, which mainly consists of 3 layers. The bottom layer consists of IoT nodes, which includes all kinds of intelligent devices and sensing nodes for data collection, including motion data, physical sign data, and environmental state data, to monitor subjects. Simple processing of the raw data may be conducted in this layer; otherwise, the data are cached and directly transferred to the edge computing layer for further data analyzing. The edge computing layer receives the data and processes the simple tasks and feeds back results down to the smart devices to display to the users, while some complex tasks are uploaded to the cloud layer for computing.

Edge computing has been applied in various fields in recent years. For example, Luan et al. [13] developed an auxiliary system with a smart medicine box, called MEMO

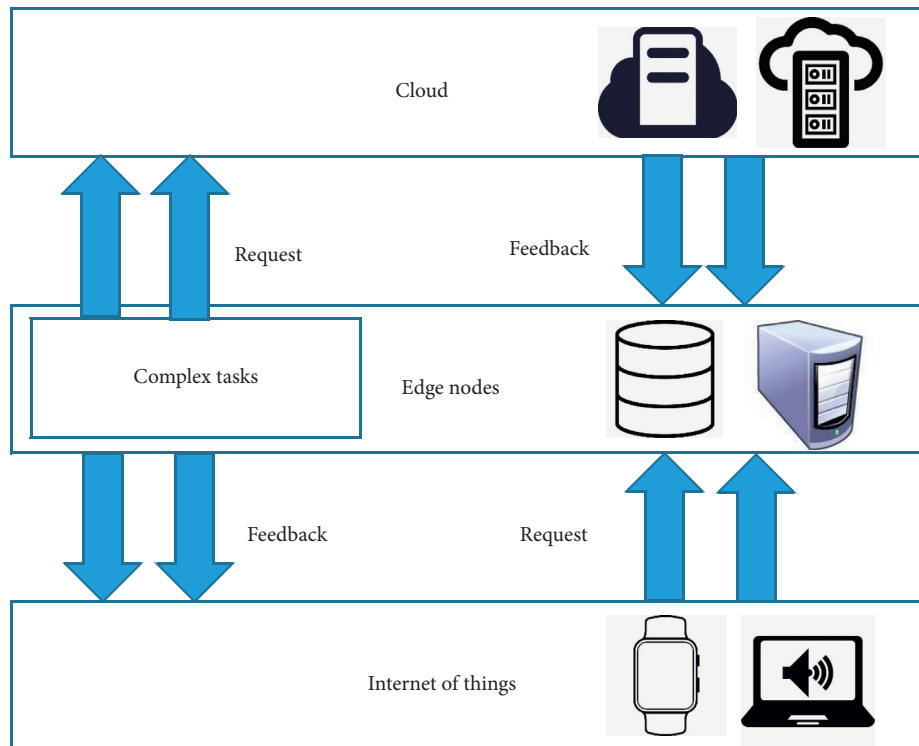


FIGURE 1: Classical edge computing architecture.

box system, in order to help the treatment of depression. The whole work is implemented by edge computing architecture, which improves the real-time performance of the system. Liu et al. [14] improved the algorithm for a patrol robot system and deployed it under the edge computing architecture, which greatly shortened the delay of the system information transmission. With the gradual popularization of 5G network, edge servers are faced with greater load, and how to properly optimize resource allocation will become a new research hotspot.

2.3. Reinforcement Learning. Reinforcement learning is a branch of machine learning methods, and its most significant feature is “learning from interaction.” Agents constantly learn knowledge according to the rewards or punishments they receive after the interaction with the environment. Since the paradigm of RL learning is very similar to how humans learn knowledge, RL is seen as an important way to achieve universal AI. Combining RL with DL, deep Q network (DQN) has achieved amazing achievements in the field of AI game robots. For example, Yoon and Kim [15] applied DQN algorithm to visual fighting games, and the game AI trained by them achieved good results in competitions. The experiment results showed the potential of the DQN approach for the two-player real-time fighting game. Shen and Kurshan [16] proposed an enhanced threshold selection policy for fraud alert systems by applying DQN algorithm to fraud detection systems. This method not only reduced the loss caused by fraud but also improved the operating efficiency of the prewarning system. In addition, DRL can also play a role in combinatorial optimization issues such as resource allocation and task scheduling.

For example, Wu et al. [17] proposed a hybrid learning strategy based on DQN for a multiuser multiserver MEC network. Simulation results showed that the hybrid approaches reach lower costs than other comparable groups. Yihang et al. [18] designed the Prioritized Memories Deep Q-Network (PM-DQN); this algorithm was applied to solve the joint routing and resource allocation problem in cognitive radio ad hoc network for minimizing the transmission delay and power consumption. Ma [19] constructed an application layout optimization strategy named min-cost to solve the layout problem of intelligent public transportation application in the city effectively. According to the simulation results, the proposed strategy can effectively reduce the total service cost of program on the basis of guaranteeing service delay.

3. Methodology

3.1. Overall Structure. This work is to design data acquisition and analysis system for track and field athletes. This paper implements the whole system using the edge computing architecture and optimizes the resource allocation of the whole system using the DRL algorithm, DDPG. The overall system architecture is shown in Figure 2.

According to Figure 2, the main operation of the system can be roughly divided into the following steps:

- (1) **Data acquisition:** collecting various kinds of data from athletes, including acceleration, angular velocity, and heart rate, through sensor modules on the athletes. The data are cached on the sensing module and ready to be forwarded to the upper edge nodes.

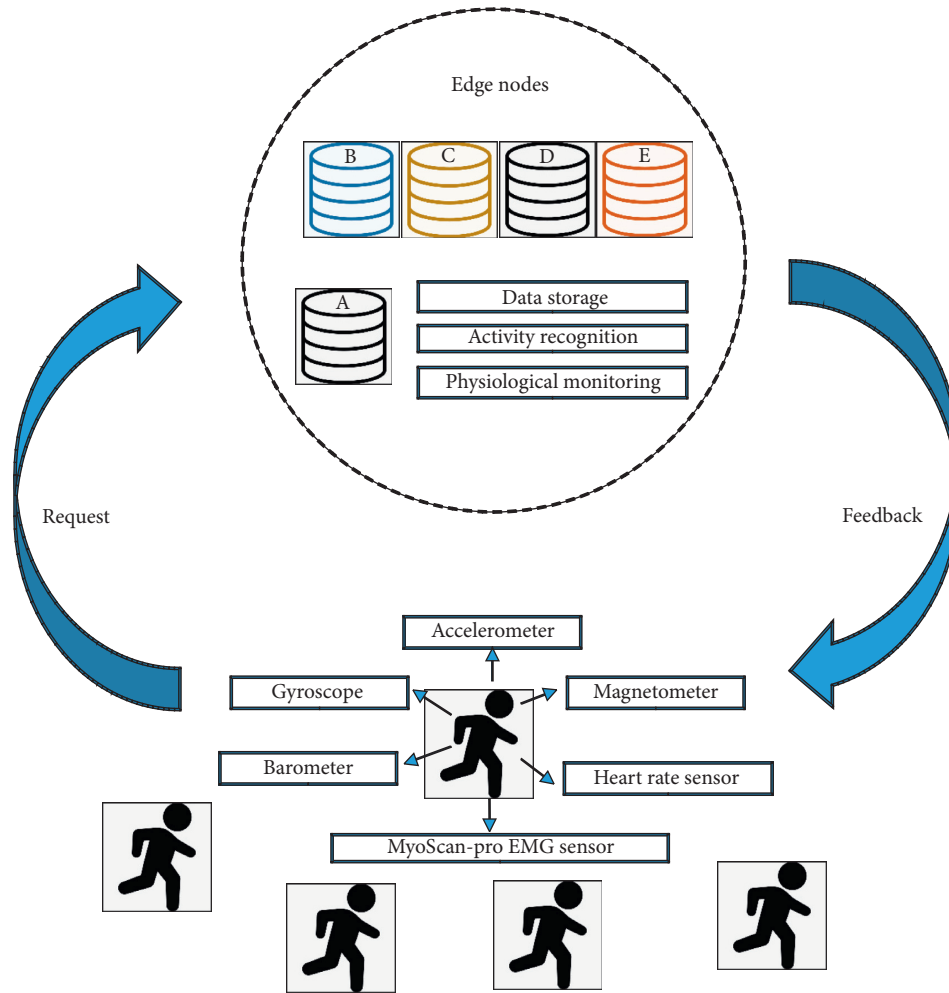


FIGURE 2: Overall system architecture.

- (2) Request submit: the sensing module transmits data and requests information to the edge node A according to the preset policy.
- (3) Request forwarding: the node A forwards subsequent tasks to other nodes after processing part of task requests. In this step, node A may find itself already overburdened and forward the requests to homogeneous node like D for processing.
- (4) Task execution: the edge node A calculates the task and generates the final result information.
- (5) Result feedback: the edge node A sends back the result down to the smart devices, which is responsible for displaying the data analysis results to the user.

According to the steps described above, the pressure on edge server is mainly from multiple users' requests. Too many requests in a period of time may cause congestion at the edge nodes. Therefore, how to optimize the resource allocation of the whole edge layer is vitally important.

3.2. Data Collection and Analysis. This section introduces details of the athletes' activity data collection and analysis,

where the MPU 9250 module is integrated into the sensing nodes as depicted in Figure 3.

The acceleration range of the module is ± 2 g, ± 4 g, ± 8 g, and ± 16 g, the gyroscope range is ± 250 , ± 500 , ± 1000 , and ± 2000 /s, and the magnetometer is up to ± 4800 UT. According to previous experiments, the node can be used for about 2 weeks if it is used intermittently and for about 24 hours if it is used continuously. The node can be charged via micro-USB interface.

The sampling frequency of the node is preset as 50 Hz. Athletes wear 5–10 of these modules for data collection and place them in body positions such as the legs and wrists. Also, sensors for recording heart rate and EMG signals are used here. All data are sent to the nearest gateway node via Bluetooth and then are forwarded to the edge computing layer.

In order to reduce the energy consumption of the sensing equipment, all further data processing is carried out on the edge computing layer. Different types of tasks have their own processing methods. Here, this paper takes the activity analysis based on inertial sensing data as an example. The edge computing node synchronizes the data by timestamp and segments the data using the sliding window

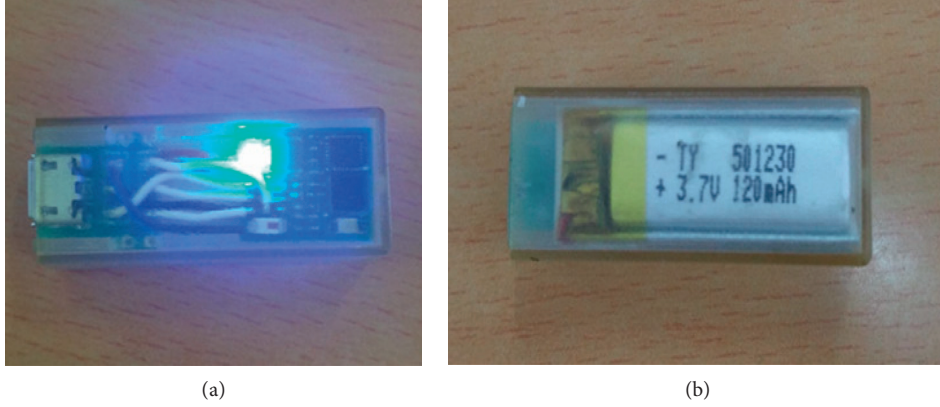


FIGURE 3: Activity data collection module. (a) Front. (b) Back.

algorithm. The data need to be feature extracted to form feature vector set in order to reduce the dimension of the inputs. In this paper, the commonly used features are extracted, including mean and variance, and their calculation formulas can be referred to in Table 1.

After the feature vector set is calculated, the pretrained machine learning model is deployed for activity analysis, such as identifying the category of the current activity. Here, this paper chooses to use support vector machine (SVM) algorithm to recognize activity and finally output the label that the activity belongs to. The whole process is summarized as Figure 4. The areas in the dashed frame are all processed by edge nodes.

3.3. DDPG-Based Dynamic Deployment Algorithm. The stable operation of the whole system depends on the reasonable system resources allocation. A dynamic resource deployment algorithm based on DDPG is proposed.

3.3.1. Problem Mapping. The problem of dynamic resource deployment is to adjust the deployment strategy of edge service resource allocation given the load of service requests at different times or on different servers. This issue with an example is shown in Figure 5.

Task loads at different time points of two edge nodes A and B are shown in Figure 5. The numbers on the left mean different types of service requests. The darker the color is, the more the service requests are. For example, there are number of requests for Service 4 on node A at time T , but there are many requests on node B at the same time. A proper allocation strategy should transmit the coming requests for Service 4 on node B to node A so as to reduce the load of B and ensure the service quality of the entire edge network.

Here, assume that the number of edge servers in the scenario is N and the number of service request types is M . In a continuous period of time, according to the request records of all services from mobile users on the edge server, the dynamic service deployment strategy σ is obtained, that is, matrix x of $N \times M$ is calculated for each period of time, as

TABLE 1: Feature extraction formulas.

| Feature | Formula |
|---------------------|---|
| Mean | \bar{a} |
| Variance | $\sum_{i=1}^n (a_i - \mu)^2$ |
| Maximum | $\max(a_i)$ |
| Minimum | $\min(a_i)$ |
| Range | $\max(a_i) - \min(a_i)$ |
| ZCR | $\sum_{i=1}^n \text{sig}(a_i > 0)$ |
| Median | $\text{median}(a_i)$ |
| MAD | $\text{median}(a_i - \text{median}(a_i))$ |
| Information entropy | $-\sum_{i=1}^m (p_i * \log(p_i))$ |
| Kurtosis | $E[(a_i - \mu)^4 / \sigma^4]$ |
| Skewness | $E[(a_i - \mu)^3 / \sigma^3]$ |
| Coefficient | $\text{cov}(X, Y)$ |

ZCR: zero crossing rate; MAD: absolute median difference.

shown in formula (1), and each element p_{ij} in X represents the ratio of service resources for deploying service s_j on the edge server node i .

$$x = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N1} & p_{N2} & \cdots & p_{NM} \end{bmatrix}. \quad (1)$$

Based on x , the whole service layer adjusts service deployment through a certain policy; then y is defined as shown in

$$y = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}. \quad (2)$$

Y represents the service allocation adjustment of a certain kind of service. For example, a_{ij} represents the number of requests forwarded by the i th node to the j th node. The function of system deployment strategy σ is to establish the relationship between x and y ; that is, when the system state is x , the probability of taking action y is

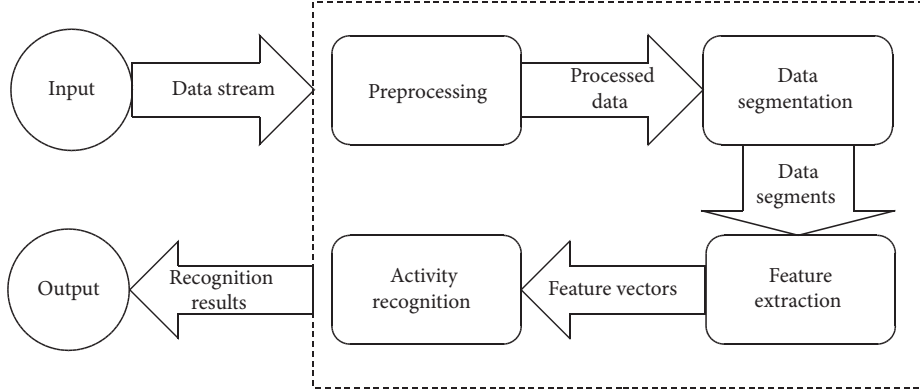


FIGURE 4: Process of activity recognition.

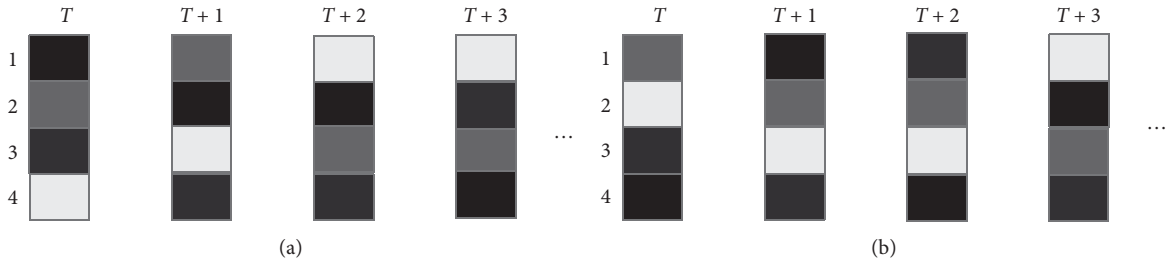


FIGURE 5: Example of load conditions on edge servers.

$$\sigma(y|x) = p(y|x). \quad (3)$$

That is, σ represents the conditional probability distribution of the system performing y operation in x state. Meanwhile, variable r is defined to evaluate the system behavior, as shown in

$$r = \begin{cases} 1, & \text{if } ct(x_t, y_t) \leq 0, \\ -1, & \text{if } ct(x_t, y_t) > 0, \end{cases} \quad (4)$$

where ct is the mean response time of the edge layer. When $ct \leq 0$ means that the strategy adopted at time t can shorten the response time, then r is positive. Otherwise, r is negative. The ultimate goal of r is to minimize the average service response time.

To simplify the model, the following assumptions are presented:

- (1) There is no interaction or dependency between services
- (2) In network transmission, the length of request message and response message is approximately same
- (3) The unit transmission delay between the sensing module and the edge node is fixed, and so is the delay between the edge node and other edge nodes

3.3.2. Algorithm Description. RL methods mainly have two branches, namely, policy-based approach and value-based approach. The DDPG algorithm is a combination of the two branches, which has the advantages of them.

As shown in Figure 6, DDPG algorithm is mainly composed of environment, experience replay memory unit, actor network module, and critic network module. The

environment is the interaction space of the agent. During the interaction, the agent obtains the interaction samples and stores them in the experience replay memory unit for training the networks. In order to optimize the learning process, DDPG algorithm takes the idea of DQN algorithm and builds a pair of artificial neural networks with identical structure for the networks, namely, the online part and the target part. Online network is used to train and update network parameters, while target network uses periodic soft update strategy to follow online network and assist online network in training. The ultimate goal is to optimize the deployment strategy σ discussed in the previous section.

According to the process shown in Figure 6, the overall algorithm steps are described as follows:

The system starts to run and conducts initialization. The network parameters in actor and critic are initialized, and the online network parameters are copied to the target network, that is, $\xi'_a \leftarrow \xi_a$, $\xi'_c \leftarrow \xi_c$. Set the size of experience replay memory as U and the size of minibatch observation data as V .

For each period, we have the following:

According to strategy σ and the current system state x_t , select an action y_t and instruct the system to execute the action. Here, $y_t = \sigma(x_t | \xi_a) + N_t$, N_t is the random noise.

The system executes y_t and obtains the reward value r_t through calculation and then enters a new resource allocation state x_{t+1} .

A state transition data set of size V is randomly sampled from experience replay memory as a small-

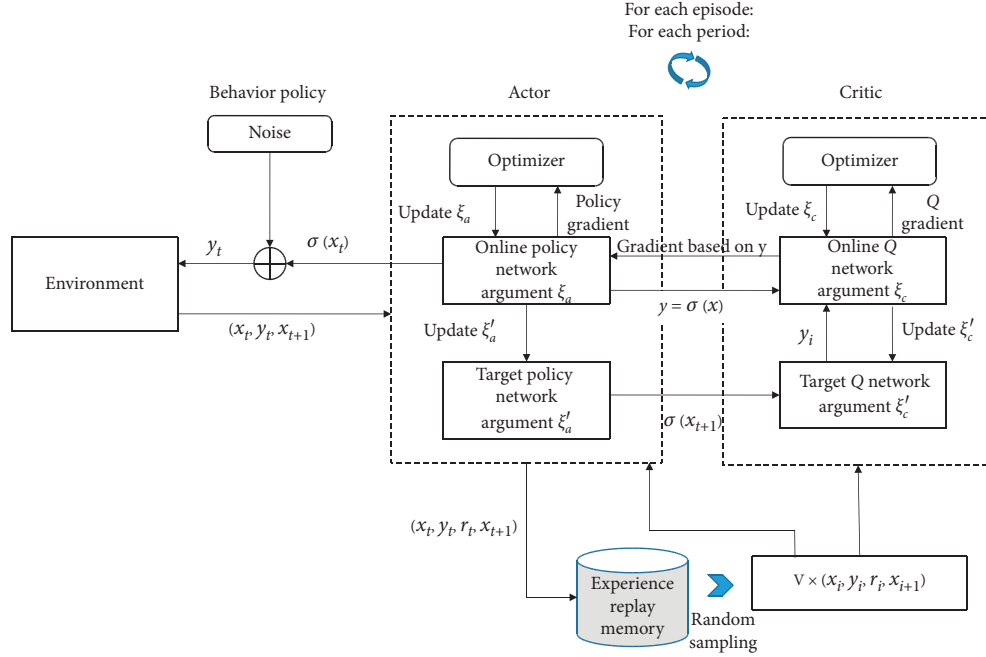


FIGURE 6: Process of the DDPG.

batch training data set for online policy network and online Q network. Each element in the set is expressed as (x_i, y_i, r_i, x_{i+1}) , $i + 1 \leq V$.

Calculate the gradient of online Q network, which is represented as $\nabla_{\xi_c} \text{Loss}$. Similar to the supervised learning method, Mean Square Error (MSE) is used here to calculate Loss. The formula is as follows:

$$\text{Loss} = \frac{1}{N} \sum_i (y_i - Q(x_i, y_i | \xi_c))^2, \quad (5)$$

where Q is the state action value function, and its expression is

$$Q(x, y) = (1 - \gamma)c(x, y) + \gamma \sum_{x' \in X} P\{x' | x, y\} V(x'), \quad (6)$$

where c represents the cost of executing action y when the system state is x . $P\{x' | x, y\}$ is the probability that the system becomes x' after action y . V is the optimal state value function obtained by the Behrman optimization equation, and the formula is

$$V(x) = \min_{y \in Y} \left\{ (1 - \gamma)c(x, y) + \gamma \sum_{x' \in X} P\{x' | x, y\} V(x') \right\}, \quad \forall x \in X. \quad (7)$$

Finally, based on the standard backpropagation method, ∇_{ξ_c} is obtained.

Update the value of ξ_c and critic network through optimizer.

Calculate the gradient of policy network, which is represented as $\nabla_{\xi_a} J$. The formula is

$$\nabla_{\xi_a} J \approx \frac{1}{N} \sum_i \nabla_y Q(x, y | \xi_c) |_{x=x_i, y=\sigma(x_i)} \nabla_{\xi_a} \sigma(x | \xi_a) |_{x_i}. \quad (8)$$

Update the value of ξ_a and actor network through optimizer.

Soft update the Target Policy Network and Target Q Network; that is,

$$\begin{aligned} \xi'_c &\leftarrow \tau \xi_c + (1 - \tau) \xi'_c, \\ \xi'_a &\leftarrow \tau \xi_a + (1 - \tau) \xi'_a, \end{aligned} \quad (9)$$

where τ is a small value, which is beneficial to the stability of target network in the update.

4. Experiment and Results

4.1. Introduction of Experiment Data. This paper plans to test the effectiveness of the resource deployment algorithm and tests the analysis ability of the system through the recognition results of athletes' activity data. First, the data from 40 track and field athletes in different events are collected, including sprint, race walk, javelin throw, pole vault, high

TABLE 2: Configuration of SVM.

| Parameter | Value |
|-------------------------|----------|
| Kernel function | RBF |
| Class weight | Balanced |
| Decision function shape | ovr |
| Gamma | 0.2 |
| C | 0.8 |

jump, and long jump. Data are collected for 3 min from each athlete, and the obtained data set is used as training data for the classifier. After training, the classifier SVM is deployed on the edge node waiting to process the task request. Table 2 gives the parameter configuration of SVM.

Then, three edge servers around the training site are deployed to handle service requests. The server configuration is shown in Table 3.

Finally, DDPG algorithm on the edge server is deployed, and the parameter settings of the algorithm are shown in Table 4.

Meanwhile, we set up two control groups for comparison with DDPG-based algorithm. One is the average-based deployment algorithm, the idea is that, for each service, the proportion of resources deployed on the server is the same. The other is the frequency-based algorithm, which allocates resources according to historical service requests.

4.2. Performance of DDPG-Based Dynamic Deployment Algorithm

4.2.1. Algorithm Convergence. This experiment mainly tests the convergence of the algorithm. We add a hidden layer with 128 neurons to the actor network and critic network of DDPG algorithm. Figure 7 shows the function loss trend after 10,000 episodes of training.

According to Figure 7, loss value of the network fluctuates greatly in the first hundreds of episodes. As the number of network training episodes increases, the loss value decreases gradually. After reaching 5000 episodes, the loss value basically converges to (0, 1), and the fluctuation becomes small. It suggests that the convergence of the algorithm based on DDPG proposed in this paper can be guaranteed.

4.2.2. Algorithm Comparison Experiment. In order to compare resource allocation algorithms, we collect user request data for several days and used the above three algorithms to carry out resource allocation on edge nodes.

Average-based method does not consider the historical service request information, so it directly sets the resource ratio of the edge server service to be the same. Frequency-based method determines the current resource allocation based on the configuration of the previous period. DDPG-based method allocates the resource based on the preset network parameters. The three algorithms have experimented with the same test set, and the average response time of the server was recorded, as shown in Figure 8.

TABLE 3: Configuration of edge nodes.

| Parameter | Description | Value |
|-----------|--|------------|
| N | Number of edge nodes | 3 |
| R | Resource capacity | [500, 800] |
| F | CPU cycle frequency | 3.2 GHz |
| CMI | Average number of cycles per instruction execution | 600 |

TABLE 4: Parameter settings.

| Parameter | Description | Value |
|-------------|---------------------------------|-------|
| U | Experience replay memory size | 300 |
| V | Minibatch size | 32 |
| Γ | Reward attenuation factor | 0.9 |
| λ_a | Learning rate of actor network | 0.001 |
| λ_c | Learning rate of critic network | 0.002 |

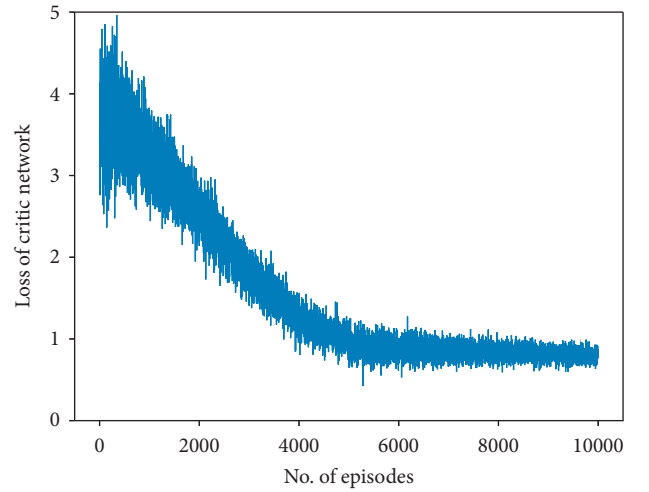


FIGURE 7: Loss trend of critic network.

Results in Figure 8 suggest that the average response time of the DDPG algorithm is the lowest, which can be controlled within about 1 s. Moreover, the user requests during the whole observation period are relatively stable. Relatively, there are more requests and the response time is relatively high only during 8:00–11:00 in the morning and 13:00–15:00 in the afternoon. To conclude, the DDPG algorithm can achieve a better deployment strategy through learning and exploring. It performs well in the track and field scenario in our work.

4.3. Performance of SVM. In order to test the system's ability on activity analyzing, the collected data are identified by the classification algorithm deployed on the edge nodes. In addition to SVM algorithm, we also introduce two other classifiers: decision tree (DT) and k -nearest neighbor (KNN) algorithm. They are frequently used in similar pattern recognition works, which are typical machine learning classifiers. We put the classification results as confusion matrix in Figure 9.

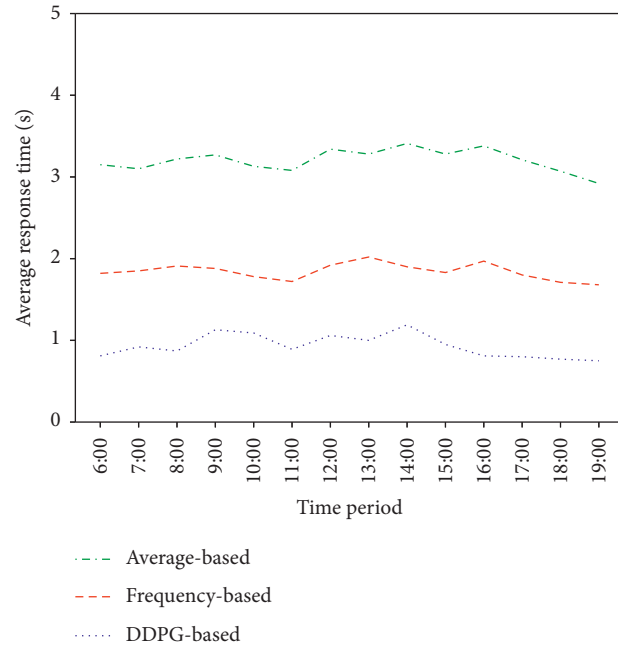


FIGURE 8: Comparison of 3 methods on average response time.

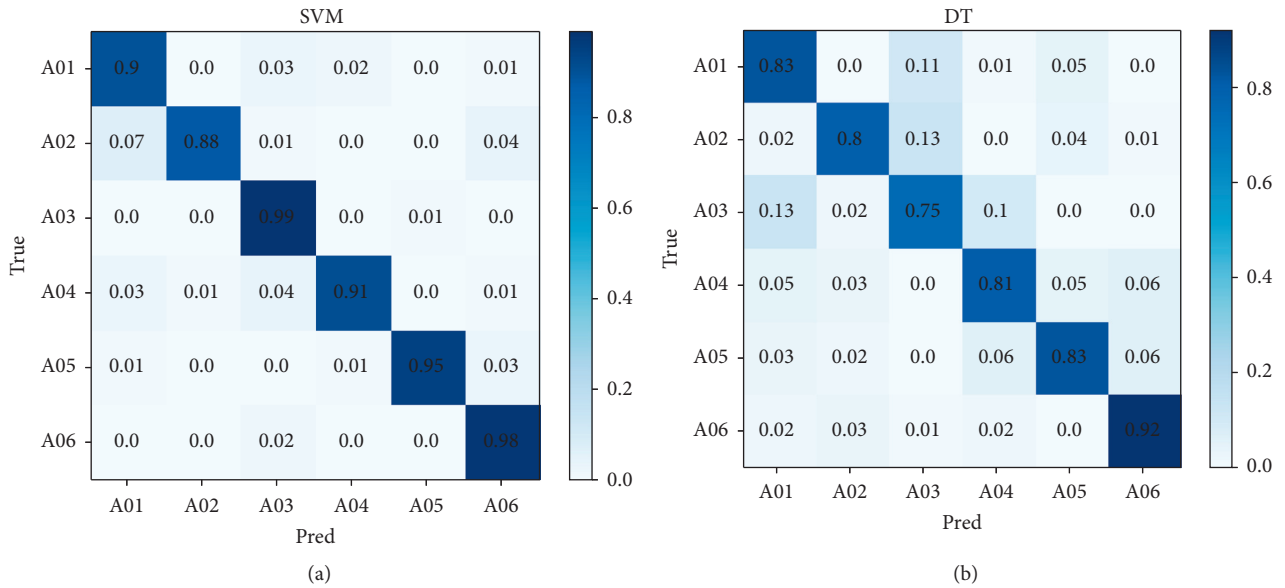


FIGURE 9: Continued.

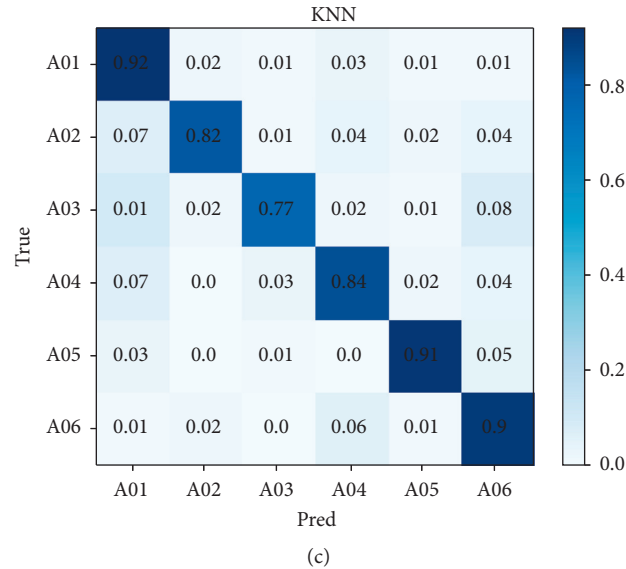


FIGURE 9: Recognition accuracy of SVM (a), DT (b), and KNN (c).

In Figure 9, A01-A06 represents the 6 kinds of activities: sprint, race walk, javelin throw, pole vault, high jump, and long jump. The darker the color block on the diagonal of the matrix is, the higher the recognition accuracy of the activity is. The recognition accuracy of SVM for all categories reached more than 88%, higher than DT and KNN. Accuracy of KNN for A01 is slightly higher than that of SVM. Given the overall recognition accuracy, SVM (93.5%) is better than DT (82.3%) and KNN (86.0%). It is better to complete the data analysis task using the SVM in our scenario.

5. Conclusion

This paper uses edge computing technology to implement data acquisition and analysis system for track and field athletes. For optimizing resource allocation and reducing the response latency, this paper introduces the DDPG algorithm to implement a dynamic resource allocation algorithm. Compared with other algorithms in the control group, the proposed algorithm has a lower average response time, within 1 s under the test environment. Meanwhile, the SVM algorithm is deployed in the edge server for activity analysis. However, the types of activities discussed in this work are relatively small. Therefore, the activity sets will be expanded in future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] E. Bobkova and E. Parfianovich, "Neural networks for forecasting and modeling training in track-and-field athletics," *Human Sport Medicine*, vol. 18, no. 5, pp. 115–119, 2018.
- [2] M. Cucco, "Smart athletic training system," US20160271447A1, 2016.
- [3] B. Ma, S. Nie, M. Ji, J. Song, and W. Wang, "Research and analysis of sports training real-time monitoring system based on mobile artificial intelligence terminal," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8879616, 10 pages, 2020.
- [4] S. Guo, "The design and application of system software about test evaluation and training of the athlete's movement function," in *Proceedings of the 5th International Conference on Computer, Automation and Power Electronics, CAPE 2017*, pp. 110–113, Dalian, China, October 2017.
- [5] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, December 2016.
- [6] Y. Li, P. A. Frangoudis, Y. Hadjadj-Aoul, and P. Bertin, "A mobile edge computing-assisted video delivery architecture for wireless heterogeneous networks," in *Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 534–539, Heraklion, Greece, July 2017.
- [7] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, Honolulu, HI, USA, April 2018.
- [8] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [9] O. Thomas, P. Suneag, G. Dror et al., "Wearable sensor activity analysis using semi-Markov models with a grammar," *Pervasive and Mobile Computing*, vol. 6, no. 3, pp. 342–350, 2010.
- [10] H. Valkova, L. Qu, and F. Chmelik, "An analysis of the physical activity of special olympic athletes with the use of an accelerometer," *Journal of US-China Medical Science*, vol. 11, no. 4, pp. 176–187, 2014.
- [11] V. Lee and J. Drake, "Physical activity data use by technoathletes: examples of collection, inscription, and identification," in *Proceedings of the 10th International Conference of*

- the Learning Sciences: The Future of Learning*, vol. 2, pp. 321–325, Sydney, Australia, July 2012.
- [12] A. Wachowicz and D. Mrozek, “Fuzzy join as a preparation step for the analysis of training data,” *Beyond Databases, Architectures and Structures. Paving the Road to Smart Data Processing and Analysis*, Springer, Berlin, Germany, 2019.
 - [13] L. Luan, W. Xiao, K. Hwang, M. S. Hossain, G. Muhammad, and A. Ghoneim, “MEMO box: health assistant for depression with medicine carrier and exercise adjustment driven by edge computing,” *IEEE Access*, vol. 8, pp. 195568–195577, 2020.
 - [14] X. Liu, B. Dong, P. Li, B. Yuan, and K. Wang, “Research and application of image recognition of substation inspection robots based on edge computing and incremental learning,” *ROBOMECH Journal*, 2021.
 - [15] S. Yoon and K. Kim, “Deep Q networks for visual fighting game AI,” in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 306–308, New York, NY, USA, August 2017.
 - [16] H. Shen and E. Kurshan, “Deep Q-network-based adaptive alert threshold selection policy for payment fraud systems in retail banking,” 2020, <http://arxiv.org/abs/2010.11062>.
 - [17] Y.-C. Wu, T. Q. Dinh, Y. Fu, C. Lin, and T. Q. S. Quek, “A hybrid DQN and optimization approach for strategy and resource allocation in MEC networks,” *IEEE Transactions on Wireless Communications*, 2021.
 - [18] D. Yihang, Z. Fan, and X. Lei, “A kind of joint routing and resource allocation scheme based on prioritized memories-deep Q network for cognitive radio ad hoc networks,” *Sensors*, vol. 18, no. 7, p. 2119, 2018.
 - [19] X. Ma, “Optimal deployment of applications based on reinforcement learning in edge computing scenarios,” *Computer Engineering and Design*, vol. 42, no. 1, pp. 15–23, 2021.

Retraction

Retracted: Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities

Mobile Information Systems

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

Copyright © 2023 Mobile Information Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] N. Hong, X. Wang, and Z. Wang, "Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities," *Mobile Information Systems*, vol. 2021, Article ID 9977736, 9 pages, 2021.

Research Article

Abnormal Access Behavior Detection of Ideological and Political MOOCs in Colleges and Universities

Ni Hong ¹, Xuefeng Wang,¹ and Zhonghua Wang²

¹Anhui Finance & Trade Vocational College, Hefei 230601, China

²Anhui University, Hefei 230039, China

Correspondence should be addressed to Ni Hong; hn@afc.edu.cn

Received 15 March 2021; Revised 5 April 2021; Accepted 13 April 2021; Published 21 April 2021

Academic Editor: Jianhui Lv

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

In many colleges and universities, MOOCs have been applied in many courses, including ideological and political course, which is very important for college students' ideological and moral education. Ideological and political MOOCs break the limitations of time and space, and students can conveniently and quickly learn ideological and political courses through the network. However, due to the openness of MOOCs, there may be some abnormal access behaviors, affecting the normal process of MOOCs. Therefore, in this paper, we propose a detection method of abnormal access behavior of ideological and political MOOCs in colleges and universities. Based on deep learning, the network behavior detection model is established to distinguish whether the network behavior is normal, so as to detect the abnormal access network behavior. In order to prove the effectiveness and efficiency of the proposed algorithm, the algorithm is compared with the other two network abnormal behavior detection methods, and the results prove that the proposed method can effectively detect the abnormal access behavior in ideological and political MOOCs.

1. Introduction

With the rapid development of information network technology and the popularization of its application, MOOCs have emerged as a new teaching mode based on Internet technology. Ideological and political MOOCs in colleges and universities apply ideological and political courses to MOOCs, breaking through limitations of time and space and greatly enhancing the pertinence and flexibility of education, which is very helpful for students to study ideological and political courses quickly and effectively. In addition to normal students learning MOOCs, there may be some abnormal network access behaviors, such as advertising and stealing students' information. Abnormal network access will affect the normal progress of the course and pose a threat to students' information security. Therefore, it is very necessary to detect the abnormal access behavior in ideological and political MOOCs to ensure the normal course and protect the information security of student users.

Throughout the traditional network security technology, it roughly includes identity authentication, encryption technology, and firewall technology. Authentication mainly verifies the actual identity of the current entity by verifying passwords, SMS, fingerprints, etc. Encryption technology mainly encrypts data through certain technical means, then transmits the encrypted data to the destination through the network, and finally decrypts the data. The technical methods mainly rely on encryption algorithms and keys [1]. On the one hand, the encryption technology combines the original data with the key through the algorithm to generate the ciphertext; on the other hand, it also relies on the algorithm and the key to restore the encrypted file to the original data. Firewall technology is the creation of a network transmission protection barrier between networks that contains software and hardware devices [2]. The barrier allows to block the intranet and extranet, thus preventing interference and attacks on the system by abnormal network behavior [2]. It is not difficult to find that these techniques

are mainly aimed at static prevention. They mainly secure the system by controlling the access rights of users or by mathematically encrypting their access rights [3]. However, on the one hand, it cannot defend against attacks from within the system and monitor the attacks in real time. Therefore, the network abnormal behavior detection technology based on dynamic defense makes up for the shortages of traditional network security in these two aspects and becomes the focus of research on computer network security. Network abnormal behavior detection is the process of identifying and monitoring behaviors that attempt to access the network [4]. It is a positive and dynamic way to protect the security behavior of a computer or network. By comparing the access network with the previously collected normal network access data, it determines whether the current access network is normal. The whole process of network abnormal detection is divided into three steps: first, collecting data, filtering the data, and sorting out the required information; second, building a model and establishing a discriminative model by training the collected data; third, abnormal behavior detection, apply the network data into the established model to detect whether the network is normal [5]. At present, the development of deep learning, which is at the forefront of research, provides a new direction and breakthrough for the study of network anomaly detection.

Traditional network abnormal behavior detection methods mainly depend on features and rules and cannot automatically use historical data to detect the latest attacks on time. In the deep learning, unsupervised or semi-supervised feature learning methods can be used instead of the traditional manual feature acquisition methods [6]. The training process of deep learning is to abstract and generalize the initial data, form more abstract high-level features by combining low-level features, and then use these features to perform a distributed feature representation of the data. Deep learning observations can be represented in a variety of ways, such as vectors of pixel intensity values in images, and more abstract points can be represented as edges or specific shape regions, etc. In this paper, based on deep learning, we propose a deep learning-based network abnormal behavior detection method to detect the abnormal access behavior of university ideological and political MOOCs.

The rest paper is organized as follows. Section 2 gives the related work. Section 3 builds the models for the proposed problem. In Section 4, the deep learning-based abnormal network behavior detection is proposed. Section 5 reports the simulation results. This paper is concluded in Section 6.

2. Related Work

The intrusion detection system was first proposed by Denning in 1987 [7]. This system described a real-time abnormal behavior detection system model that can detect computer misuse in the form of intrusion, infiltration, etc. Currently, although users have high expectations for intrusion detection systems, in practical applications, the current network abnormal behavior analysis model cannot

solve the problem of high false alarm and leakage rates, and it also has problems such as the system is not easy to manage, has poor detection performance, has lack of automation, and has poor security performance. Therefore, the design of a truly effective intrusion detection system by improving network abnormal behavior detection algorithms has a central role in the field of network security.

Network abnormal behavior detection is mainly based on network traffic classification and network topology analysis, and research studies of network abnormal behavior are mainly divided into the following categories:

- (1) Research on abnormal detection of network traffic based on measurement learning

Network traffic refers to the data transmitted on the network, which records a large number of user information. It plays an important role in the field of network security and is an important feature of network behavior detection. Jin and Yeung [8] proposed an algorithm model based on covariance analysis to detect abnormal traffic. In this algorithm, all packets in unit time were counted and classified according to different network protocols. The covariance matrix of each type was obtained, and the anomaly was detected according to the matrix. However, the algorithm needs a lot of complex mathematical calculations, which is easy to affect the use of normal network communication in anomaly detection. Barford et al. [9] proposed a detection algorithm based on multiscale analysis and wavelet transform, taking the time correlation between traffic data as the breakthrough point. However, this algorithm can only analyze the traffic data captured in a single link and can only complete the anomaly detection in that link. Due to the shortcomings of the above algorithm, Rubinstein et al. [10] proposed a network abnormal behavior detection method based on component analysis (PCA). This method transforms the original data into a traffic matrix and, based on the spatial difference of traffic data between different links, detects and processes the data in the subspace established by the algorithm. However, these algorithms have common shortcomings. On the one hand, when the data is large, the calculation of these algorithms is more complex and time-consuming. On the other hand, it is difficult to avoid the deviation caused by subjectivity in the feature extraction of network abnormal data, and it cannot completely rely on the algorithm to complete the feature extraction, classification, and detection of abnormal data.

- (2) Research on network abnormal behavior based on alarm transmission

In the current environment of increasing network devices and network data, it has become an important method to quickly analyze the alarm transmission through monitoring the network to maintain the quality of network communication,

especially for mass devices to conduct high-precision correlation analysis at the level of network topology to locate the alarm, which is very important to reduce the impact of the fault and improve the quality of network maintenance. Hu et al. [11] held that static alarm analysis was not enough to realize alarm location and used time sequence alarm information and corresponding dynamic time matrix to realize alarm correlation analysis, so as to complete alarm location. Karoly and Abonyi [12] took multiperiod alarm sequences to mine alarm data and complete the compression of alarm data. However, the above research studies have not focused on the network management data, especially when the types of terminal nodes are increasing and the alarm messages are massive and sparse, the redundancy of the algorithm mining results is high, and the alarm transmission analysis is difficult.

- (3) Research on network propagation model based on cascading failure

Network abnormal behaviors are often accompanied by the movement and diffusion of abnormal points in the network, which spreads the failure of a single node to multiple nodes and causes greater losses. Cascading failure refers to that, after a node fails, the load on it is redistributed to its adjacent nodes according to the coupling relationship between nodes, which leads to an overload of the load on the adjacent nodes and then failure, which leads to a large area of network paralysis. Cascading failure may occur in any actual network system, such as communication network, computer network, and transportation network, which will lead to a sharp decline in network performance or even network system collapse. Motter and Lai [13] and Jin et al. [14] held that the overload of the network node or the edge was one of the most important reasons. They proposed a model to analyze the cascading overload fault caused by dynamic traffic redistribution. The results show that the network is vulnerable to this kind of cascading anomaly. Parshani et al. [15] established a network composed of connections and dependent links, introduced the model and analysis framework for the study of interdependent networks, and concluded that the wider the distribution, the more vulnerable the network becomes to the random failure of interdependent networks. Alessandro [16] suggested that, in addition to the topological connectivity links, the dependency relationship between network nodes also accelerates the propagation of network failures and affects the mechanism of cascading exceptions, and the relationships can be functional or logical.

Different from the above, this paper applies the abnormal access behavior detection into the field of ideological and political MOOCs. Besides, the simulation of this paper also depends on the real dataset. In particular, the abnormal

access behavior detection method is improved by considering deep learning.

3. Problem Modeling

3.1. Network Access Behavior System Model. In the university ideological and political MOOCs network access system, as shown in Figure 1, each student can request network access from the server through the campus network to obtain video resources. In the system, the form of vector is used to represent the access of network users, and the network access behavior of each user is expressed as $N = (ID, IP, URL, time)$, where ID is the ID of the requesting user, IP is the IP of the requesting user, URL is the requested URL, and time is the time of the requesting user browsing the website.

3.2. Network Abnormal Behavior Detection Model. Network abnormal behavior detection is the process of qualitatively identifying and detecting attempted network intrusions. It is an active and dynamic security behavior to protect the computer or network, rather than a passive emergency measure to work when the intrusion occurs. Network abnormal behavior detection works as follows. On the one hand, everything has common behavioral characteristics. On the other hand, abnormal behavior generally has serious differences compared with normal behavior. Then, to detect whether a network subject is attacked by the abnormal behavior, the current behavior can be compared with the previously collected normal behavior in a comprehensive manner to determine whether the current behavior is normal. Therefore, based on the common characteristics of normal behaviors and the difference of abnormal behaviors, we can quantitatively analyze these normal behaviors and thus find out the established rules of normal behaviors through the results.

The object of our algorithm is to identify the abnormal network access behavior. In this paper, the characteristics of different types of traffic are identified for multiclassification through the collection of network abnormal traffic and feature extraction. The whole process of the algorithm is as follows: first, collect the network abnormal traffic; then divide the traffic by tuple, the traffic contains different categories; and label the corresponding categories. Because the storage format of traffic data itself is hexadecimal code, it can be directly converted into a decimal value, and the fixed position value can be extracted from each traffic to form a high-dimensional vector to complete vectorization. The vectorized samples are sent to the traffic classification network composed of CNN for training. The detection process of network abnormal access behavior is shown in Figure 2.

4. Abnormal Network Behavior Detection Based on Deep Learning

Multilayer perceptron (MLP) is the foundation of deep neural network [17]. However, it is difficult for multilayer perception to deal with the weight problem of hidden layers, and this problem can only be solved after the development of

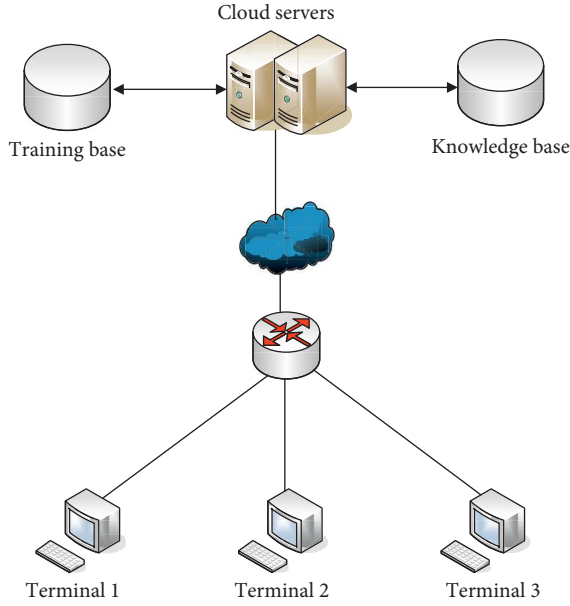


FIGURE 1: University ideological and political MOOCs network access system.

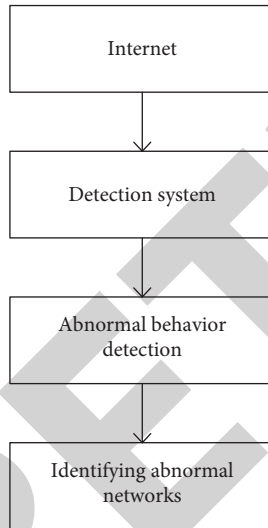


FIGURE 2: Network abnormal behavior detection model.

back propagation algorithm [18]. With the rapid development of big data and cloud computing, deep neural networks have gradually become a hot topic in the field of machine learning and have achieved certain achievements in many fields. Convolutional neural network (CNN) is a representative algorithm of deep learning. In this paper, a network behavior detection model is built using CNN.

The essence of CNN is to use the nonlinear deep network structure to approximate the function, so as to express the mapping relationship from input to output. Compared with the traditional neural networks, CNN has changed the original way of maintaining the complete connection between neurons in each layer, but adopted a way of

maintaining the partial connection between neurons in each layer. Therefore, CNN can learn the essential characteristics of the training set from a small number of samples.

In the algorithm proposed in this paper, the CNN network structure used is the network structure of Lenet-5 [19] modified according to the data of abnormal behavioral features of the network. Lenet-5 is a convolutional neural network designed by Yan Lecun in 1998. It is mainly used for handwritten digit recognition and is one of the representative structures of CNN. Lenet-5 consists of 7 layers, as shown in Figure 3. In general, it is mainly divided into convolutional layer, pooling layer, and fully connected layer. Each layer includes a different number of training parameters. Meanwhile, each layer of Lenet-5 has multiple feature maps, and the input features are extracted by convolutional filters with multiple neurons per feature map. The whole neural network is trained by a backpropagation algorithm using the original image as data. Since the purpose of this paper is to detect whether the network access is normal or not, the structure of the model designed in this paper is divided into two categories: normal and abnormal.

4.1. Convolution Layer. In the convolution layer of the network, the weight represents the proportion of neurons connected to each other, which is usually expressed as a convolution kernel. In this layer, each neuron is connected to the upper layer by partial connection rather than full connection. Each feature map output by convolution layer is convoluted with the processed convolution kernel and the feature map of the previous layer, and the corresponding elements are accumulated and then an offset value is added. Finally, the result is processed by a sigmoid function. The convolution process is shown in Figure 4.

We assume that the number of convolution layers is 1, the convolution kernel is k , the feature map of the upper layer is M_j , and the offset value is b ; the formula of the convolution layer is expressed as follows:

$$x_j^l = f \left(\sum_{i \in M_j} x_j^{i-1} k_{ij}^l + b_j^l \right). \quad (1)$$

4.2. Pooling Layer. The pooling layer is the abstraction of local information. In the neural network, the pooling layer is used to process the feature map output from the upper layer. In this layer, the sampling operation will not change the number of the feature map, but will process the size of the map and reduce the feature map to a certain extent. According to the convolution layer, the mapping formula of the pooling layer is expressed as

$$x_j^i = f(\beta_j^i \text{pool}(x_j^{i-1}) + b_j^i), \quad (2)$$

where the pool function represents the sampling function, which generally includes two methods: mean pooling and maximum pooling. In this paper, we use the maximum pooling method, which is shown in Figure 5.

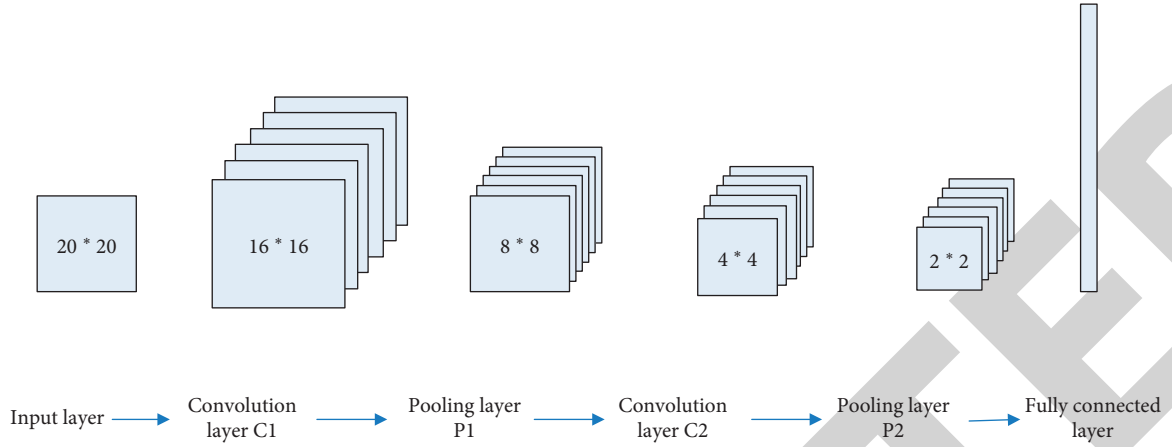


FIGURE 3: Neural network structure.

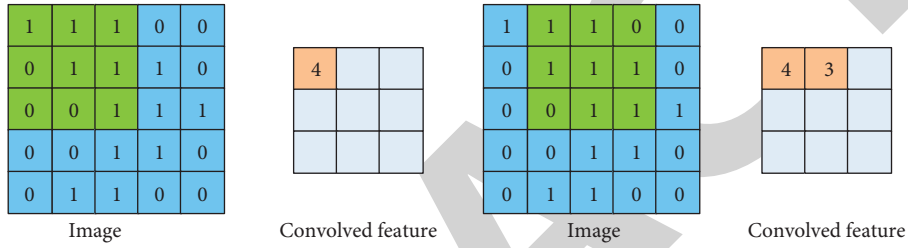


FIGURE 4: Convolution process.

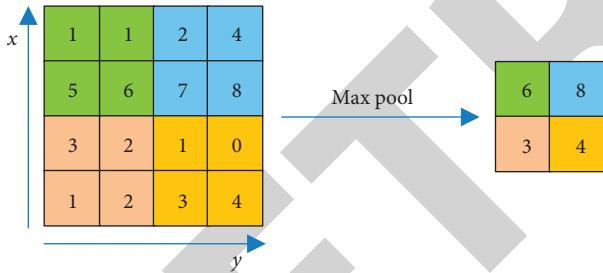


FIGURE 5: Maximum pooling method.

4.3. *Fully Connected Layer.* The full connection layer is mainly used to calculate the data processed by the former convolution layer and sampling layer, and its calculation result is the output value of the network.

There are two independent convolution cores in CNN network. After the operation of each layer mentioned above and the softmax layer, each accessed network behavior will have a type of scoring output. Suppose there are N training segments, we take the n th training sample. Then, after passing through the network, the output will be $O_n(t)$, and then, through the softmax layer, according to the data at time t of the sample, the score of category i is computed as follows:

$$P_N^i(t) = \frac{e^{O_N^i(t)}}{\sum_{j=1}^{K+1} e^{O_N^j(t)}}. \quad (3)$$

The final loss function is computed as follows:

$$L = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^L (-\log(P_n^{Z_n}(t))). \quad (4)$$

5. Simulations and Analysis

In this section, we conduct simulations on the proposed mechanism and analyze simulation results by comparing our mechanism with other two algorithms to show that our mechanism is effective and efficient to detect the abnormal access behavior in ideological and political MOOCs.

5.1. Experimental Preparation

5.1.1. *Experimental Environment.* The simulations are conducted on *Python* with a computer configured with Intel(R) core (TM)i5-9400F, CPU 2.90 GHZ, 8 GB RAM.

5.1.2. *Dataset.* In this paper, experiments are based on real dataset to verify the effectiveness and efficiency of the proposed algorithm. The real dataset comes from the statistical ideological and political MOOCs website of a university. The website uses Apache as the server, including an abnormal sample dataset and a normal sample dataset. The abnormal sample dataset includes 3964 abnormal requests, and the normal sample includes 55059 normal requests. The detailed information of the experimental dataset is shown in Table 1.

TABLE 1: The detailed information of the experimental dataset.

| Id | Attribute | Description |
|----|---------------|---|
| 1 | Duration | Length of connection time(s) |
| 2 | Protocol type | Type of protocol (e.g., TCP) |
| 3 | Service | Network service (e.g., HTTP) |
| 4 | Logged in | Login status (1 or 0) |
| 5 | Count | The number of connections to the same destination |
| 6 | src_bytes | Number of bytes from source to destination |
| 7 | dst_bytes | Number of bytes from destination to source |

5.1.3. Data Preprocessing. Before detecting the web log data, it is necessary to preprocess the log data. Data preprocessing is a very important step in log analysis. The purpose is to transform the collected original data into a data format suitable for analysis and delete the noise data which is easy to cause interference to the experiment. In this paper, data preprocessing mainly includes data cleaning and data normalization. Data cleaning includes log filtering, log parsing, and antiobfuscation.

(1) Data Cleaning. The network abnormal behavior analysis based on Web log is to obtain the original data information from the web server and then analyze the web log. However, because users are random in the process of browsing the website, the web log will generate a lot of useless information, that is, the log records not related to intrusion detection; only for the relevant and accurate log information and only through modeling and analysis can we get accurate detection results. For example, the log with GIF and JPEG suffixes is the special data returned by the server to the user. Therefore, if the HTTP request repeatedly contains the log with JPEG, GIF, and CSS suffixes, it needs to be cleared. These log files are actually documenting embedded in the website and not the web pages actually requested by users. When attackers launch attacks, they usually use js obfuscation, URL encoding, and Unicode encoding to hide the intrusion behavior. Before intrusion detection on the request attribute of Web log, it is necessary to perform antiobfuscation, URL decoding, and Unicode decoding on the HTTP request to expose attack mode and eliminate all interference factors of subsequent intrusion detection. Through the data cleaning process, Web logs are converted into reliable data for security analysis. In addition, we need to de-duplicate the HTTP requests of Web logs and delete the duplicate HTTP request records, which can prevent the deviation of detection results caused by overlearning certain types of samples.

(2) Data Normalization. All feature attributes of the dataset processed by feature engineering are digital vectors. However, the dimension level of each attribute is inconsistent, so it is necessary to normalize all feature attributes. The purpose is to unify each feature attribute to the same dimension level according to the statistical distribution of the original feature attributes so that the features of different dimensions can be compared, and the contribution of each dimension to the classification results is the same. In this experiment, the

range transformation method is used to standardize the feature data so that all the data are classified into [0, 1]. Through this method, the influence of different dimension levels between the original data on the classification results is solved.

5.2. Evaluation Indicators. To evaluate the performance of abnormal detection algorithms, we consider the effectiveness, detection efficiency, and availability of the algorithm. The effectiveness mainly shows the detection accuracy and reliability of the algorithm, which is regarded as the main index to evaluate the intrusion detection system, and is also the purpose of the design and development of intrusion detection system. The detection efficiency considers the speed of data processing in the detection system, including the training stage and detection stage. Usability measures the stability and error recovery ability of the detection system itself.

In essence, the abnormal detection model is a classifier for normal requests and abnormal requests, so we can use the evaluation index of the classifier to evaluate the performance of the anomaly detection model. In the field of machine learning, the most commonly used evaluation indicators are accuracy, recall, and *F1*-measure. These evaluation indicators are used to evaluate the effectiveness of the detection algorithm proposed in this paper. The accuracy represents the number of samples correctly classified divided by the total number of samples. In general, the higher the accuracy, the better the classifier effect. The precision represents how many of the samples predicted as positive samples are true positive samples. The recall refers to the ratio of the number of positive samples correctly predicted to the total number of correct samples predicted. The *F1*-measure is a weighted summed average of precision and recall. When *F1*-measure is higher, the method is more effective. Accuracy, precision, recall, and *F1*-measure are computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (7)$$

$$F1 - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \quad (8)$$

- (1) True positive cases (TP): the number of samples that are predicted to be positive cases and are actually positive cases, that is, the prediction is correct
- (2) True negative cases (TN): the number of samples predicted to be negative cases, which is also negative cases, that is, the prediction is correct
- (3) False positive cases (FP): the number of samples that are predicted to be positive cases but actually are negative cases, that is, the prediction is error

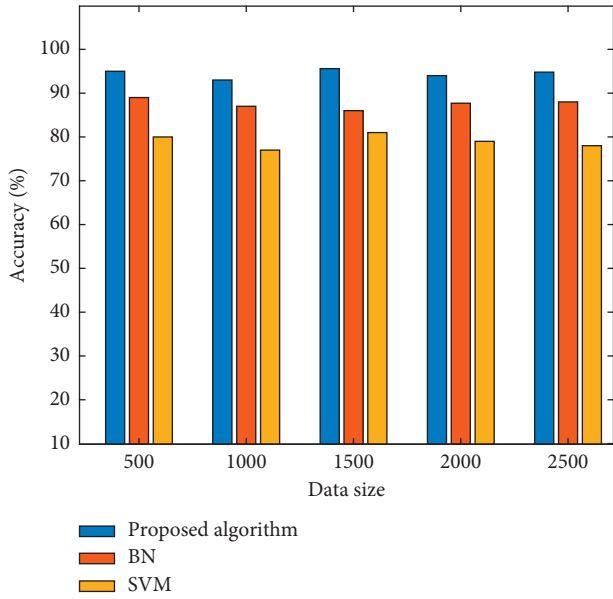


FIGURE 6: Comparison results of accuracy under different sizes of data.

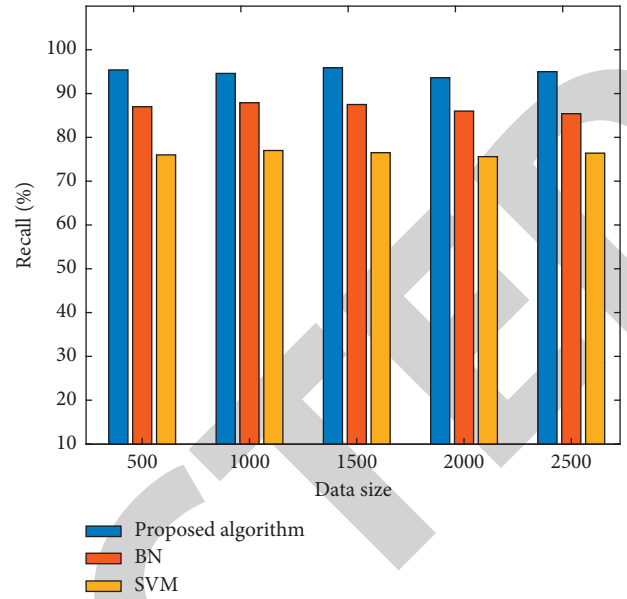


FIGURE 8: Comparison results of recall under different sizes of data.

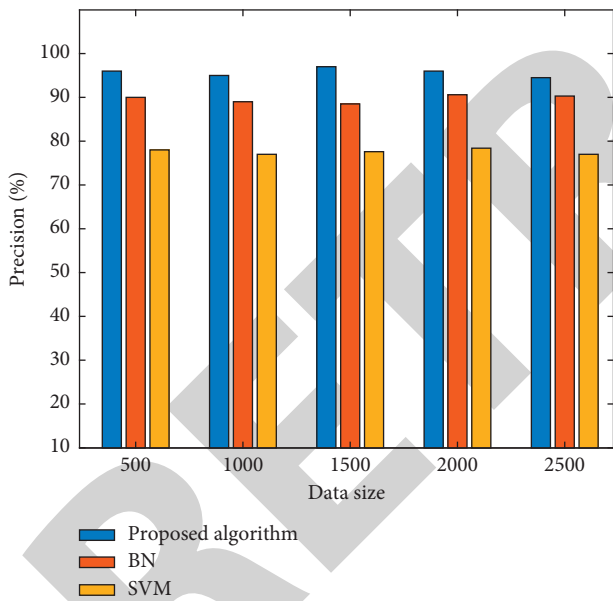


FIGURE 7: Comparison results of precision under different sizes of data.

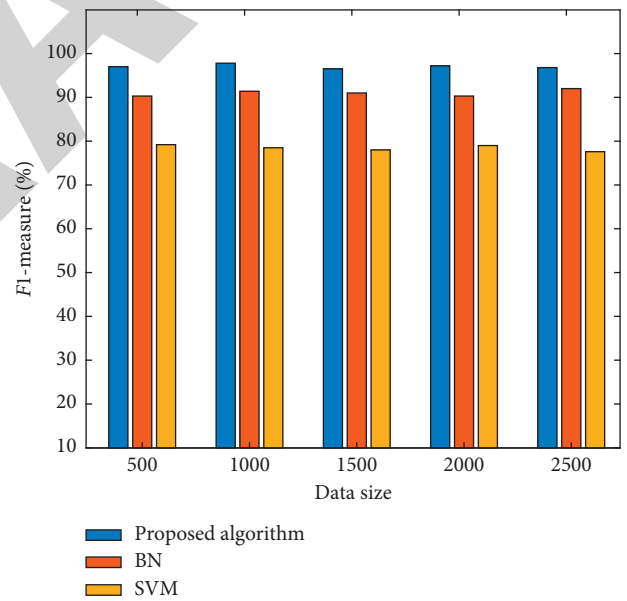


FIGURE 9: Comparison results of F1-measure under different sizes of data.

- (4) False negative cases (FN): the number of samples predicted as negative cases but actually are positive cases, that is, the prediction is error

5.3. Experimental Results. In order to prove the effectiveness and efficiency of the proposed algorithm, this paper compares the proposed algorithm with detection algorithms and with Belief Network (BN) and SVM under different data volumes, aiming at accuracy, precision, recall, and F1-measure.

5.3.1. Accuracy. Accuracy is computed by formula (5), and the comparison experimental results of accuracy are shown in Figure 6. As shown in the figure, it can be seen that compared with the other two methods, the accuracy of this algorithm is always the highest under different sizes of experimental data, which proves that the proposed algorithm performs best in terms of accuracy.

5.3.2. Precision. Precision is computed by formula (6), and the comparison experimental results of precision are shown

in Figure 7. As shown in the figure, it can be seen that compared with the other two methods, the precision of this algorithm is always the highest under different amount of experimental data, which proves that the proposed algorithm performs best in terms of precision.

5.3.3. Recall. Recall is computed by formula (7), and the comparison experimental results of recall are shown in Figure 8. As shown in the figure, it can be seen that compared with the other two methods, the recall of this algorithm is always the highest under different amount of experimental data, which proves that the proposed algorithm performs best in terms of recall.

5.3.4. F1-Measure. F1-measure is computed by formula (8), and the comparison comparative experimental results of F1-measure are shown in Figure 9. As shown in the figure, it can be seen that compared with the other two methods, the F1-measure of this algorithm is always the highest under different amount of experimental data, which proves that the proposed algorithm performs best in terms of F1-measure.

6. Conclusion

With the rapid development of information network technology and the popularization of its application, MOOCs have been used in many courses in many domestic colleges and universities, including the ideological and political education course, which is very important to the ideological and moral education of college students. Aiming at the possible network access behavior in ideological and political MOOCs in colleges and universities, we construct the abnormal network behavior detection model based on deep learning to detect whether the network behavior is normal. In order to prove the effectiveness and efficiency of this algorithm, the algorithm proposed in this paper is compared with the other two network abnormal behavior detection methods. The experimental results show that the proposed method can effectively detect the abnormal access behavior in ideological and political MOOCs. Compared with the anomaly detection method based on traditional classification algorithm, the method based on deep learning can adaptively detect the abnormal network access behavior and does not need to manually mark features. However, there are some limitations in this paper. The datasets used in the simulations are not complex enough, which makes the application of the system very limited. In the future work, we apply the abnormal network behavior detection model based on deep learning to more data by adjusting the network structure and parameters and further study different complex scenarios.

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.

References

- [1] E. Biersack, C. Callegari, and M. Matijasevic, *Data Traffic Monitoring and Analysis*, Springer Berlin Heidelberg, Heidelberg, Germany, 2013.
- [2] C. Xue and X. Yan, "Software defect prediction based on improved deep forest algorithm," *Computer Science*, vol. 45, no. 8, pp. 160–165, 2018.
- [3] S. M. Ghaffarian and H. R. Shahriari, "Software vulnerability analysis and discovery using machine-learning and data-mining techniques," *ACM Computing Surveys*, vol. 50, no. 4, pp. 1–36, 2017.
- [4] H. Wang, H. Chen, and S. Liu, "Intrusion detection system based on improved Naive Bayes algorithm," *Computer Science*, vol. 41, no. 4, pp. 111–115, 2013.
- [5] Z. Chen, C. Du, and L. Huang, "Improving image classification performance with automatically hierarchical label clustering," in *Proceedings of the 24th International Conference on Pattern Recognition*, pp. 1863–1868, Beijing, China, 2018.
- [6] Z. H. Zhou and J. Feng, "Deep forest: towards an alternative to deep neural networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.
- [7] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.
- [8] S. Jin and D. S. Yeung, "A covariance analysis model for DDoS attack detection," in *Proceedings of the IEEE International Conference on Communications*, pp. 1882–1886, Paris, France, 2004.
- [9] P. Barford, J. Kline, D. Plonka et al., "A signal analysis of network traffic anomalies," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002*, vol. 1, no. 1, pp. 71–82, New York, NY, USA, 2002.
- [10] B. I. P. Rubinstein, B. Nelson, L. Huang et al., "Stealthy poisoning attacks on PCA-based anomaly detectors," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 2, pp. 73–74, 2009.
- [11] W. Hu, T. Chen, and S. L. Shah, "Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7290–7300, 2018.
- [12] R. Karoly and J. Abonyi, "Multi-temporal sequential pattern mining based improvement of alarm management systems," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 2016.
- [13] A. E. Motter and Y. C. Lai, "Cascade-based attacks on complex networks," *Physical Review E*, vol. 66, no. 6, Article ID 065102, 2003.
- [14] L. Jin, X. Wang, Y. Zhang et al., "Cascading failure in multilayer networks with dynamic dependency groups," *Chinese Physics B*, vol. 27, no. 9, Article ID 098901, 2018.
- [15] R. Parshani, S. V. Buldyrev, and S. Havlin, "Critical effect of dependency groups on the function of networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 3, pp. 1007–1010, 2010.
- [16] V. Alessandro, "Complex networks: the fragility of interdependency," *Nature*, vol. 464, no. 7291, 2010.

Research Article

High-Performance Server-Based Live Streaming Transmission Optimization for Sports Events in Smart Cities

Si Chen and Gang Zhao 

Shenzhen University, Shenzhen 518060, China

Correspondence should be addressed to Gang Zhao; zhaogang@szu.edu.cn

Received 4 March 2021; Revised 22 March 2021; Accepted 1 April 2021; Published 10 April 2021

Academic Editor: Jianhui Lv

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

Smart cities allow cities to run more efficiently and have been approved by a lot of cities. During the process of building smart cities, a large amount of data is generated. Particularly, live sports events have been regarded as the inalienable part of smart cities. However, with the improvement in the quality of life, people tend to obtain better watching experience in terms of sports events. For such purpose, this paper proposes the live streaming transmission optimization method based on high-performance server, called HPTO, including two main modules, that is, high-performance server optimization and transmission optimization. Specifically, for the server optimization, this paper devises a distributed storage strategy to avoid producing the internal disk fragments and improve the writing efficiency of sports videos. For the transmission optimization, this paper devises a deep-learning-based video compression strategy to save the storage space of server and accelerate the transmission of sports videos. In addition, this paper makes simulation experiments based on PyCharm. The experimental results show that HPTO has higher storage efficiency, smaller transmission time, and lower packet loss rate than benchmarks, which indicates that the proposed two aspects of optimization strategies (server optimization and transmission optimization) are efficient.

1. Introduction

Smart cities [1–3] usually apply modern network infrastructures and information techniques to improve the citizens' living quality and make cities run more efficiently. Recently, there have been some cities committed to building smart cities in the world, including Chicago, Milton Keynes, Busan, and Shanghai. With the continuous improvement of life standard, citizens like to watch the live sports with the satisfactory quality of experience. However, these behaviors will generate a large amount of data, which is expected to reach ZB-level. In fact, the live streaming transmission of sports events usually shows the concurrency feature [4, 5], which has very high requirements on storage system and transmission path. In order to guarantee the sustained and steady work of smart cities during the process of watching sports events, two aspects of optimizations have been approved, that is, server optimization and transmission optimization.

For the server optimization, it refers to improve the storage system. The current storage system usually relies on file system management or raw disk design [6, 7]. The file-management-based storage system will produce lots of internal disk fragments as the magnetic head moves frequently. Besides, the file system needs to maintain index information and attribute information, which handles the redundant information and goes against the storage of sports video data. Different from the file-management-based storage system, the raw-disk-based storage system directly performs reading and writing operations by the application programs, which greatly improves the efficiency of I/O. However, the live streaming of sports events has the concurrency feature, which causes the data storage spaces to be relatively scattered and thus produces the internal disk fragments. Especially when the malfunction happens, the storage system exits the large failure probability; that is to say, the raw-disk-based storage system has no high reliability. Different from the above, in this paper, the storage

system of server is optimized based on the distributed storage method.

For the transmission optimization, it is consisted of two kinds of methods. One is the selection of transmission path, that is, routing [8]. However, smart cities usually depend on the backbone network, and the transmission path is usually assigned by service provider in advance. Thus, in terms of live streaming transmission of sports events, the optimization of transmission path has no obvious improvement effect on the transmission performance. The other one is the content compression; that is, the sports video is compressed into that with a smaller size, but all attributes are not changed. Video compression has two advantages. On one hand, the sports video can be transmitted in an effective way. On the other hand, the storage space of high-performance server can be saved to reduce its pressure so that the high-performance server works smoothly. The principle of video compression is to compile the original video through encoder [9, 10], where the encoder has two functions, that is, (i) reading the signals of video and (ii) discerning and counting the residual signals. In this paper, the deep learning is used to complete the compression of sports video.

In smart cities, this paper proposes high-performance-server-based live streaming transmission optimization method, named HPTO. HPTO's main contributions are summarized as the three following aspects:

A distributed storage strategy is devised to avoid the generation of internal disk fragments and improve the writing efficiency, that is, server optimization

A deep learning method is used to compress the sports video to save the storage space and accelerate the transmission

The rich experiments are made, including neural network verification, video compression verification, and live streaming optimization verification

The rest of the paper is organized as follows. Section 2 introduces the server optimization. Section 3 presents the transmission optimization. Section 4 reports the experiment results. This paper is concluded in Section 5.

2. High-Performance Server Optimization

2.1. Storage Structure of Sports Video. To avoid producing the internal disk fragments resulting from the concurrent and stochastic writing of live streaming, this paper handles the sports video by designing high-speed buffer structure and disk logic storage structure. In addition, this paper also uses the buffer mapping policy, which belongs to the bcache-based hybrid storage technology to complete the connection between high-speed buffer and disk logic storage. To be specific, solid state disk (SSD) [11] does the special high-speed caching operation for the written live streaming, where the group of pictures (GOP) is used as the basic unit to allocate the fixed buffer segment for each channel's live streaming. The whole storage structure of sports video is shown in Figure 1.

The high-speed buffer structure consists of superblock, buffer bitmap, and buffer segments. Among them, the superblock is a superfield that reports some parameters' information including creation time, buffer size, buffer number, and allocation condition. In particular, the file that superblock corresponds to is assigned as "0xEF53" completed when the formatting order is started. The buffer bitmap field is used to describe the service condition regarding the subsequent buffer segments. The remaining fields are buffer segments regarded as the basic unit to allocate and recycle the live streaming of sports video. In this paper, the size of buffer segment is set as 16 MB. In particular, when the remaining space cannot allocate a complete buffer segment, it is reserved. Besides, when the last GOP of live streaming is completely written, the corresponding buffer space is recycled.

The disk logic storage structure consists of superblock, data block bitmap, primary index, secondary index, and buffer segments. In particular, the file that superblock corresponds to is assigned as "0xEF53" completed when the formatting order is started. The primary index covers several parameters' information including ID of live streaming, starting and ending time, bit rate type, and GOP. The secondary index is specially used to cover the detailed information of GOP.

The buffer mapping strategy supports that multiple hard disk drives (HDDs) use the same SSD as the caching disk. To be specific, the "echo" statement is used to attach the "cset.uuid" of caching disk from high-speed buffer structure to disk logic storage structure; at the same time, the writing order is marked as "writeback."

2.2. Storage Management. The raw disk usually refers to such special character-driven device without the formatting operation and it cannot be managed by Unix/Linux's file system [12]; thus the space management is very inflexible, and the corresponding on-demand enlargement requirement is very difficult to be satisfied. Given this, this paper leverages the logical volume (LV) to complete the enlargement of high-performance server. The whole storage management of high-performance server is shown in Figure 2.

To be specific, when the unallocated space of LV group (LVG) can satisfy the enlargement requirement, the required enlargement size is divided. Then, the "rawdevice" installed and used upon it is bound to "/dev/raw/raw[*]." On the contrary, when the enlargement requirement cannot be satisfied, the additional physical disk is added and then such installation and binding operations can be started.

If the enlargement requirement cannot be satisfied by the above-mentioned enlargement method, the current file system will laterally increase the external high-performance server to reach the purpose of enlargement. In particular, the increased high performance submits its condition to the state manager through the heartbeat protocol.

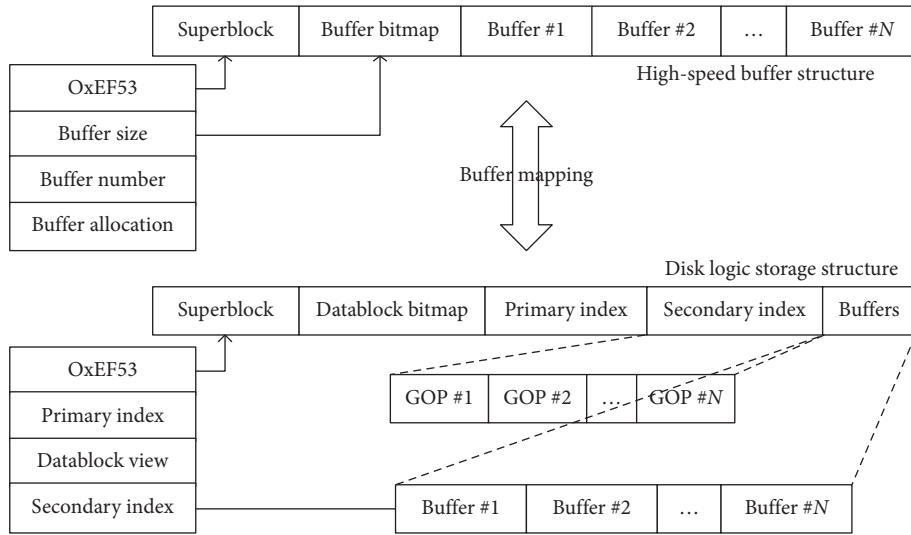


FIGURE 1: The whole storage structure of sports video.

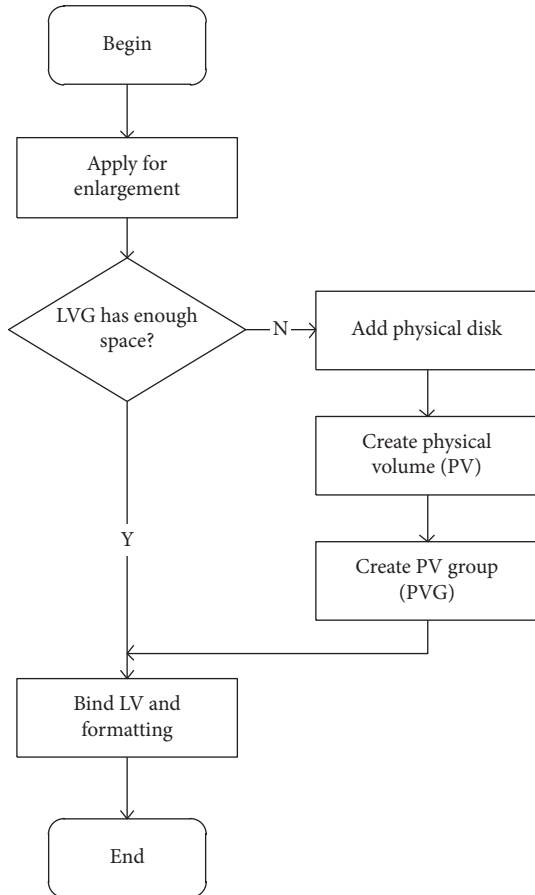


FIGURE 2: The whole storage management of high-performance server.

2.3. *Writing Method of Sports Video.* In this paper, the writing method of sports video is the single thread. At first, the live streams of sports video are arranged according to the time of request storage. Then, they are scheduled concurrently via these buffers, where the single thread coding way is

used to handle these concurrent live streams. The single thread coding way does the consecutive storage via the hugepage, which effectively avoids the time consumption due to waiting for addressing of multithreaded concurrency; therefore, the transmission efficiency is improved, and the storage time is decreased. In particular, a buffer is only used to store one channel’s live streaming, which guarantees the consecutiveness of the physical storage space.

3. Transmission Optimization Based on Video Compression

3.1. *Methodology.* In smart cities, higher video compression proportion is needed as the transmission of live streaming is subjected to the limited transmission rate and storage space. In particular, the live streaming of sports video involves lots of images; that is to say, video compression is also called image compression. The current video image compression mainly compresses the intraframe data, that is, storing the previous video image information with less data [13].

This paper proposes a video segment compression method to realize the transmission optimization, where two key frames are extracted and used to store one video segment. In addition, on the video decoder side, the frame interpolation method is used to recover the previous video segment data. In particular, three-dimensional convolutional neural network (3DCNN) (a deep learning method [14, 15]) is used to make the classification for these sports video segments by analyzing temporal information and spatial information of video frame sequences, in which there are three kinds of video segments, that is, radical change, gradual change, and ordinary change. In fact, 3DCNN has attracted attention for video information processing, since it introduces the time dimension innovatively on the basis of spatial dimensions to capture the contextual information between the different frames in the sports video.

In order to guarantee the efficient and accurate classification, a large enough surface dataset is very necessary.

Otherwise, it is considerably difficult or even impossible to train a high-precision 3DCNN. Given this, this paper presents a large enough dataset with 249621 sports video segments, including 135202 radical change segments, 103146 gradual change segments, and 11263 ordinary change segments. Meanwhile, each sports video segment consists of 16 image sequences, and the overlapping number of frames between two sports video segments is set as 8 in order to prevent the leak detection phenomenon from happening.

3.2. 3DCNN-Based Video Compression. The used 3DCNN has five convolution layers, three pooling layers, and three full connection layers. Among them, each convolution layer includes a rectified linear unit (ReLU) as the activation function with the local response normalization (LRN) operation. The first two full connection layers include 2048 neurons and the last full connection layers includes three neurons, where each neuron corresponds to one sports video segment. The detailed information of 3DCNN in this paper is shown in Table 1.

Furthermore, the ordinary change video segment's head frame and tail frame are used to express the whole sports video segment, and the whole length cannot exceed 32 frames.

4. Experiment Results

4.1. Experiment Method. In smart cities, the proposed HPTO is implemented based on Intel (R) Core (TM) i5-8500 CPU @3.00 GHz, RAM 8.00 GB, running on the Ubuntu16.02 64-bit operation system. The programming language is Python, running on PyCharm. The verification of HPTO includes three aspects. At first, the 3DCNN-based deep learning method at the transmission optimization part is verified. In particular, two benchmarks [16, 17] regarding CNN are used for the comparisons with evaluating recall ratio, precision ratio, and four F values ($F1/F2/F3/F4$). Then, the video compression method based on 3DCNN is verified, in which one benchmark [18] regarding video compression is used for the comparison with bit rate evaluation. Finally, the whole live streaming transmission optimization scheme including high-performance server optimization and transmission optimization is verified. Meanwhile, two benchmarks [19, 20] regarding live streaming optimization are used for the comparisons with evaluating storage efficiency, transmission time, and packet loss rate. In total, the five above-mentioned benchmarks are denoted by KumarB, LiuB, RahaB, HeB, and LiB, respectively, and they are introduced as follows:

Kumar et al. [16] did a comparative study on CNN for any real-time image classification and object recognition, where CNN had that much of ability to create optimized video image classifications and object recognitions

Liu et al. [17] proposed a memristor-based 3DCNN to recognize and classify the behaviors of human in the video with 6 main actions

TABLE 1: The detailed information of 3DCNN.

| Current layer | Core parameter | Connection layer |
|---------------|----------------------------|------------------|
| Data | — | — |
| Conv1 | $(3 * 3 * 3 * 3) * 96$ | ReLU + LRN |
| Pool1 | — | — |
| Conv2 | $(3 * 3 * 3 * 96) * 256$ | ReLU + LRN |
| Pool2 | — | — |
| Conv3 | $(3 * 3 * 3 * 256) * 384$ | ReLU |
| Conv4 | $(3 * 3 * 3 * 384) * 384$ | ReLU |
| Conv5 | $(3 * 3 * 3 * 384) * 256$ | ReLU |
| Pool3 | — | — |
| Fc1 | $(8 * 7 * 7 * 256) * 2048$ | ReLU |
| Fc2 | $2048 * 2048$ | ReLU |
| Fc3 | $2048 * 3$ | — |
| Softmax | Label | — |

Raghavendra et al. [18] devised different image compression techniques without any data loss

He et al. [19] proposed an uncoded multiuser video streaming system by exploiting the diversities of video contents and channel conditions of multiple users

Li et al. [20] presented a joint optimization method for conversational HD video service, taking into account the linkage between video coding and transmission

Furthermore, the seven above-mentioned performance evaluation metrics are introduced as follows:

The recall ratio rec is defined as

$$rec = \frac{TP}{TP + FN} \quad (1)$$

where TP indicates true positives and FN indicates false negatives.

The precision ratio pre is defined as

$$pre = \frac{TP}{TP + FP} \quad (2)$$

where FP indicates false positives.

F value is defined as

$$F\alpha = \frac{(\alpha^2 + 1) * rec * pre}{\alpha^2 * (rec + pre)} \quad (3)$$

When $\alpha = 1$, $F1$ value is obtained.

The bit rate is defined as the transmitted number of bits per second (kbps).

The storage efficiency is defined as the utilization rate of high-performance server's storage space.

The transmission time is defined as the time difference between the timepoint when the first video segment of live streaming is sent from the high-performance server side and that when the last video segment of live streaming arrives at the decoder side.

The packet loss rate is defined as the ratio of the lost number of video segments to the total number of video segments.

4.2. 3DCNN Verification. The experiment results on recall ratio based on six time simulations are shown in Table 2. We can observe that the proposed HPTO has the best recall ratio, followed by LiuB and KumarB. In particular, the recall ratio of HPTO can reach about 99%, increasing about 2% and 4.5% compared to LiuB and KumarB, respectively. HPTO and LiuB have higher recall ratio than KumarB, which results from the fact that they use 3DCNN structure to recognize and classify these live streams of sports video. Here, we emphasize that 3DCNN introduces the time dimension innovatively on the basis of spatial dimensions to capture the contextual frame information in the sports video and it has better performance than those traditional CNN structures. For HPTO and LiuB, the former presents the deep training based on Table 1, while the latter has no further improvement on 3DCNN structure. Therefore, HPTO has higher recall ratio compared to LiuB.

The experiment results on precision ratio based on six time simulations are shown in Table 3. We also find that the proposed HPTO has the highest precision ratio, followed by LiuB and KumarB. Similar reasons are found from the above statements.

Based on Tables 2 and 3, the average experiment results on F values including $F1$, $F2$, $F3$, and $F4$ are shown in Table 4. We can observe that, with the increase of α , the corresponding F value becomes smaller and smaller. As a matter of fact, the evaluation based on $F1$ has the highest reference value. In particular, the larger $F1$ value means that the corresponding strategy has better performance. As can be seen from Table 4, the proposed HPTO has the largest $F1$ value, which means that HPTO has the best classification effect on the live streaming of sports video.

4.3. Video Compression Verification. This section considers six kinds of sports videos (NBA, CBA, German Bundesliga, Serie A, World Cup, and AOTC) and two kinds of encoding structures (H.264/AVC and HEVC). The average experiment results on bit rate are shown in Table 5. The improvement degrees on bit rate are shown in Table 6. We can observe that the proposed HPTO has an obvious advantage in terms of increasing the bit rate. In particular, the improvement rate of bit rate compared to the benchmark can reach 65.52% (Serie A) based on H.264/AVC encoding structure and 85.29% (World Cup) based on HEVC encoding structure, respectively. This further indicates that the proposed video compression optimization scheme is efficient.

4.4. Live Streaming Optimization Verification. The experiment results on storage efficiency based on six time simulations are shown in Table 7. We can observe that HPTO has the highest storage efficiency, followed by HeB and LiB. In particular, the storage efficiency of HPTO can reach about 86%, but those of HeB and LiB can only reach about 79% and 72%, respectively. Different from two benchmarks, HPTO makes two aspects of optimization on live streaming, that is, server optimization and transmission optimization. To be specific, the server optimization improves the storage structure, storage management method, and writing

TABLE 2: The experiment results on recall ratio based on six-time simulations.

| Strategy | 1 (%) | 2 (%) | 3 (%) | 4 (%) | 5 (%) | 6 (%) |
|----------|---------|---------|---------|---------|---------|---------|
| HPTO | 99.1218 | 98.9072 | 99.0638 | 98.9291 | 99.2275 | 98.8809 |
| KumarB | 94.5038 | 94.3911 | 94.4927 | 94.4335 | 94.2908 | 94.3266 |
| LiuB | 97.2787 | 97.3053 | 96.9806 | 97.1913 | 97.2327 | 97.3371 |

TABLE 3: The experiment results on precision ratio based on six-time simulations.

| Strategy | 1 (%) | 2 (%) | 3 (%) | 4 (%) | 5 (%) | 6 (%) |
|----------|---------|---------|---------|---------|---------|---------|
| HPTO | 98.7056 | 98.8561 | 98.6632 | 98.7354 | 98.5962 | 98.6911 |
| KumarB | 92.7118 | 92.6822 | 92.7386 | 92.6136 | 92.5893 | 92.6357 |
| LiuB | 96.1923 | 96.2083 | 95.8804 | 96.1353 | 96.2607 | 95.9186 |

TABLE 4: The average experiment results on F values including $F1$, $F2$, $F3$, and $F4$

| F values | $F1$ (%) | $F2$ (%) | $F3$ (%) | $F4$ (%) |
|------------|----------|----------|----------|----------|
| HPTO | 98.8646 | 61.7903 | 54.9248 | 52.5218 |
| KumarB | 93.5260 | 58.4538 | 51.9589 | 49.6857 |
| LiuB | 96.6568 | 60.4105 | 53.6982 | 51.3489 |

TABLE 5: The average experiment results on bit rate.

| Live streaming | HPTO (H.264/AVC) | RaghaB (H.264/AVC) | HPTO (HEVC) | RaghaB (HEVC) |
|-------------------|------------------|--------------------|-------------|---------------|
| NBA | 2.7 kbps | 2.1 kbps | 4.1 kbps | 2.7 kbps |
| CBA | 2.9 kbps | 1.8 kbps | 5.3 kbps | 3.1 kbps |
| German Bundesliga | 3.6 kbps | 2.5 kbps | 4.7 kbps | 2.9 kbps |
| Serie A | 4.8 kbps | 2.9 kbps | 5.8 kbps | 3.5 kbps |
| World Cup | 3.7 kbps | 2.4 kbps | 6.3 kbps | 3.4 kbps |
| AOTC | 4.2 kbps | 3.1 kbps | 5.6 kbps | 3.8 kbps |

method. In addition, 3DCNN is also employed to optimize the video compression at the transmission optimization part. In fact, 3DCNN structure and video compression scheme have presented the efficient experiment results, which can be found in Section 4.2 and Section 4.3.

The experiment results on transmission time based on six-time simulations are shown in Table 8. We can find that the proposed HPTO has the smallest transmission time, followed by LiB and HeB. In particular, HPTO has more than twice as transmission time as HeB; this is because HPTO increases the storage time and saves more storage space to accelerate the transmission of live streaming. Different from HeB, LiB presents a joint optimization method for conversational HD video service by considering the linkage between video coding and transmission; thus it has smaller transmission time than HeB.

The experiment results on packet loss rate based on six-time simulations are shown in Table 9. We can find that the proposed HPTO has the lowest packet loss rate, followed by HeB and LiB, which means that HPTO will present the best watching experience in terms of the sports video. In particular, the packet loss rate of HPTO almost reaches 0%, which indicates that server optimization and transmission

TABLE 6: The improvement degrees on bit rate.

| Live streaming | H.264/AVC (%) | HEVC (%) |
|-------------------|---------------|----------|
| NBA | 28.57 | 51.85 |
| CBA | 61.11 | 70.97 |
| German Bundesliga | 44.00 | 62.07 |
| Serie A | 65.52 | 65.71 |
| World Cup | 54.17 | 85.29 |
| AOTC | 35.48 | 73.68 |

TABLE 7: The experiment results on storage efficiency based on six-time simulations.

| Strategy | 1 (%) | 2 (%) | 3 (%) | 4 (%) | 5 (%) | 6 (%) |
|----------|-------|-------|-------|-------|-------|-------|
| HPTO | 86.37 | 86.29 | 86.34 | 86.95 | 86.78 | 86.43 |
| HeB | 79.62 | 79.16 | 79.68 | 79.49 | 79.09 | 79.27 |
| LiB | 72.58 | 72.47 | 72.63 | 72.11 | 72.56 | 72.38 |

TABLE 8: The experiment results on transmission time based on six-time simulations.

| Strategy | 1 | 2 | 3 | 4 | 5 | 6 |
|----------|------------|------------|------------|------------|------------|------------|
| HPTO | 16.2845 ms | 17.2219 ms | 16.8246 ms | 17.0662 ms | 16.9034 ms | 17.2254 ms |
| HeB | 36.8526 ms | 36.1944 ms | 36.5356 ms | 36.7113 ms | 36.2876 ms | 36.5071 ms |
| LiB | 22.3638 ms | 21.9873 ms | 22.0643 ms | 21.6555 ms | 22.1967 ms | 22.3897 ms |

TABLE 9: The experiment results on packet loss rate based on six-time simulations.

| Strategy | 1 (%) | 2 (%) | 3 (%) | 4 (%) | 5 (%) | 6 (%) |
|----------|-------|-------|-------|-------|-------|-------|
| HPTO | 0 | 0 | 0.10 | 0 | 0.12 | 0 |
| HeB | 0.18 | 0.24 | 0.18 | 0.20 | 0.22 | 0.20 |
| LiB | 0.34 | 0.28 | 0.32 | 0.28 | 0.30 | 0.34 |

optimization can guarantee that all live streams of sports video arrive at the user end.

5. Conclusions

In smart cities, the live streaming optimization of sports video is very important because it has a direct influence on the watching quality. In this paper, a live streaming transmission optimization method based on server optimization and transmission optimization is proposed. Meanwhile, the server optimization includes storage structure optimization, storage management optimization, and writing optimization method. The transmission optimization mainly depends on the video compression based on 3DCNN structure.

The experiments include 3DCNN verification, video compression verification, and live streaming optimization verification, with evaluation of seven metrics, that is, recall ratio, precision ratio, four F values, bit rate, storage efficiency, transmission time, and packet loss rate. All experiment results show that the proposed HPTO has better performance to optimize the live streaming of sports video.

In the future, we will test more metrics and more applications. In addition, we also plan to make a real demo for the proposed live streaming transmission optimization mechanism by connecting some high-performance servers.

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.

References

- [1] L. Quijano-Sanchez, I. Cantador, M. E. Cortes-Cediel et al., "Recommender systems for smart cities," *Information Systems*, vol. 92, pp. 1–22, 2020.
- [2] A. Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, "Future trends and current state of smart city concepts: a survey," *IEEE Access*, vol. 8, pp. 86448–86467, 2020.
- [3] E. Ismagilova, L. Hughes, Y. K. Dwivedi, and K. R. Raman, "Smart cities: advances in research-an information systems perspective," *International Journal of Information Management*, vol. 47, pp. 88–100, 2019.
- [4] K. Takayama, T. Fujimoto, R. Endo et al., "Neighbor selection based on transmission bandwidth on P2P live streaming service," in *Proceedings of 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA'12)*, pp. 105–110, Fukuoka-shi, Japan, March 2012.
- [5] M. Liu, P. Hong, J. Li et al., "LM-MCM: a new layered multicast transmission protocol for live streaming," in *Proceedings of 14th IEEE International Conference on Networks*, pp. 1–6, Singapore, September 2007.
- [6] R. Salunkhe, A. D. Kadam, N. Jayakumar et al., "In search of a scalable file system state-of-the-art file systems review and map view of new Scalable File system," in *Proceedings of 2016 International Conference on Electrical, Electronics, and*

- Optimization Techniques (ICEEOT)*, pp. 364–371, Chennai, India, March 2016.
- [7] S. Kim, J. Han, H. Eom et al., “Improving I/O performance in distributed file systems for flash-based SSDs by access pattern reshaping,” *Future Generation Computer Systems*, vol. 115, pp. 365–373, 2020.
 - [8] V. M. Goncalves, E. M. B. Bolonhez, G. E. M. Campos et al., “Transmission line routing optimization using rapid random trees,” *Electric Power Systems Research*, vol. 194, pp. 1–11, 2021.
 - [9] R. J. Clarke, “Image and video compression: a survey,” *International Journal of Imaging Systems and Technology*, vol. 10, pp. 1–13, 1999.
 - [10] J. A. Jacob and N. S. Kumar, “FPGA implementation of optimal 3D-integer DCT structure for video compression,” *The Scientific World Journal*, vol. 2015, Article ID 204378, 9 pages, 2015.
 - [11] S. Kim, S. Lee, S. Kim et al., “Environmental effects of the technology transformation from hard-disk to solid-state drives from resource depletion and toxicity management perspectives,” *Integrated Environmental Assessment and Management*, vol. 15, pp. 1–7, 2019.
 - [12] M. Wang and Q. Zhang, “Optimized data storage algorithm of IoT based on cloud computing in distributed system,” *Computer Communications*, vol. 157, pp. 124–131, 2020.
 - [13] D. R. Bhojani, V. J. Dwivedi, and R. M. Thanki, *Hybrid Video Compression Standard*, pp. 1–58, Springer Briefs in Applied Sciences and Technology, Berlin, Germany, 2020.
 - [14] H. Yin, Y. Wei, H. Liu et al., “Deep Convolutional Generative Adversarial Network and Convolutional Neural Network for Smoke Detection,” *Complexity*, vol. 2020, pp. 1–12, Article ID 6843869, 2020.
 - [15] Z. Xie, Z. Guo, and C. Qian, “Palmprint gender classification by convolutional neural network,” *IET Computer Vision*, vol. 12, pp. 1–8, 2018.
 - [16] K. K. Kumar, M. D. Kumar, C. Samsonu et al., “Role of convolutional neural networks for any real time image classification, recognition and analysis,” *Materials Today: Proceedings*, 2021.
 - [17] J. Liu, Z. Li, Y. Tang et al., “3D Convolutional Neural Network based on memristor for video recognition,” *Pattern Recognition Letters*, vol. 130, pp. 116–124, 2018.
 - [18] C. Raghavendra, S. Sivasubramanian, and A. Kumaravel, “Improved image compression using effective lossless compression technique,” *Cluster Computing*, vol. 22, no. S2, pp. 3911–3916, 2019.
 - [19] C. He, Y. Hu, Y. Chen, X. Fan, H. Li, and B. Zeng, “MUCast: linear uncoded multiuser video streaming with channel assignment and power allocation optimization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 1136–1146, 2020.
 - [20] H. Li, W. Lei, W. Zhang, and Y. Guan, “A joint optimization method of coding and transmission for conversational HD video service,” *Computer Communications*, vol. 145, pp. 243–262, 2019.

Research Article

Intelligent Forwarding Strategy for Congestion Control Using Q-Learning and LSTM in Named Data Networking

Sanguk Ryu ¹, Inwhee Joe ¹, and WonTae Kim ²

¹Department of Computer and Software, Hanyang University, 222 Wangsimni-ro, Seoul 04763, Republic of Korea

²Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 31253, Republic of Korea

Correspondence should be addressed to WonTae Kim; wtkim@koreatech.ac.kr

Received 7 February 2021; Revised 9 March 2021; Accepted 23 March 2021; Published 31 March 2021

Academic Editor: Jianhui Lv

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

Named data networking (NDN) is a future network architecture that replaces IP-oriented communication with content-oriented communication and has new features such as cache, multiple paths, and multiple sources. Services such as video streaming, to which NDN can be applied in the future, can cause congestion if data is concentrated on one of the nodes during high demand. To solve this problem, sending rate control methods such as TCP congestion control have been proposed, but they do not adequately reflect the characteristics of NDN. Therefore, we use reinforcement learning and deep learning to propose a congestion control method that takes advantage of multipath features. The intelligent forwarding strategy for congestion control using Q-learning and long short-term memory in NDN proposed in this paper is divided into two phases. The first phase uses an LSTM model to train a pending interest table (PIT) entry rate that can be used as an indicator to detect congestion by knowing the amount of data returned. In the second phase, it is forwarded to an alternative path that is not congestive via Q-learning based on the PIT entry rate predicted by the trained LSTM model. The simulation results show that the proposed method increases the data reception rate by 6.5% and 19.5% and decreases the packet drop rate by 7.3% and 17.2% compared to an adaptive SRTT-based forwarding strategy (ASF) and BestRoute.

1. Introduction

The rapid development of the Internet has led to a significant increase in the amount of content transmitted every year. However, these changes have been difficult to adapt to because the current Internet architecture depends on IP addresses and is designed for end-to-end communication. This drawback causes problems such as transport efficiency and security.

Information-centric networking (ICN) was proposed as a solution to the problem caused by the rapid increase of content. The goal was to change the communication paradigm from a content-oriented model to the IP-based model [1]. Named data networking (NDN), one of the most well-known ICN architectures, is attracting attention as a hotspot for research [2–4].

NDN is a future network architecture that alters the current IP-based Internet as the Internet environment

changes, replacing IP addresses with named content for communication. Compared with traditional TCP/IP, it has the following new features in the transmission method. First, NDN communication is a consumer-driven pull mode and is connectionless. The consumer sends an interest packet to request the content, and the producer with the requested content returns the matched data. The second is a multi-source feature. NDN has a content store (CS), where the returned content can be temporarily stored in the intermediate nodes in a network. Therefore, the consumer can receive the requested data from multiple sources, including the CS of the intermediate node and the producer with the original data. Third, NDN has multipath features, so it supports dynamic multipath forwarding. The NDN node provides multiple paths from the consumer to sources via a forwarding information base (FIB) that stores the interface information where the packet can go to next. It then decides

how to use the path provided through a forwarding strategy. Although this change solves the limitations of the current Internet to some extent, if NDN is applied to a service such as video streaming, congestion may occur at a node, where data is concentrated when people are crowded during a certain period. Therefore, congestion control is one of the major research tasks of NDN.

Congestion control of NDN has been proposed by applying the TCP/IP method. TCP/IP congestion control detects congestion via the retransmission timeout (RTO) and adjusts the sending rate via an additive increase/multiplicative decrease (AIMD) window-based mechanism. However, congestion detection through RTO is not a reliable indicator in NDN, where different round-trip times (RTTs) are measured for each source as it has a multisource feature. Furthermore, the window control method targeting a single path of TCP/IP is not suitable for NDN due to its characteristics of multiple sources and multiple paths. The reason is that when a consumer receives data from two sources through different paths, if one path is congested and the consumer reduces the window size, the throughput of the other path that is not congested also decreases. As such, the direct application of existing solutions does not adequately consider the characteristics of NDN, so network congestion control methods must also change. Therefore, it is necessary to propose a new congestion control method for NDN.

In this paper, we propose an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking (IFS-QLSTM) using a dynamic forwarding method to utilize multiple paths. First, the IFS-QLSTM uses the LSTM model to train the number of entries that change due to packets added to the pending interest table (PIT) in the NDN node (we use the term PIT entry rate interchangeably in the rest of the paper). Second, the PIT entry rate predicted by the trained LSTM model is used for the reinforcement learning to judge the congested node. The node is then bypassed and the packet is forwarded.

The rest of this paper is organized as follows. Section 2 explains the background of NDN and related research. Section 3 describes an intelligent forwarding strategy for congestion control using Q-learning and LSTM in NDN. Section 4 presents the performance evaluation and analysis of the results through simulation. Finally, Section 5 concludes the paper.

2. Related Works

In recent years, NDN has been studied as a future network architecture that will replace the current Internet. One of the core technologies of NDN architecture is congestion control. We survey related studies in two aspects: (1) studies on control of the interest sending rate for congestion control and (2) studies on adaptive forwarding strategy [5–7].

Researches on the interest sending rate for congestion control include a receiver-based window control method and a hop-by-hop interest sending rate control method. In [8], the authors describe a receiver-based window control scheme that controls the interest sending rate by adjusting the congestion window using a TCP-like mechanism in the

receiver of RTT. Similarly, both ICTP and CCTCP use a method of adjusting the congestion window based on RTT [9, 10]. However, the NDN caches data through the CS added to the router, which causes the RTT to change irregularly. In addition, when a consumer requests data from multiple sources and one source is congested, the consumer reduces the window size. This means that the amount of transmission to the source where congestion does not occur is also reduced. Therefore, the traditional receiver-based window control congestion control method is not suitable for NDN. In [11], the authors demonstrate a representative hop-by-hop method that detects congestion in intermediate nodes and adjusts the interest sending rate using interest shaping. Wang et al. [12] proposed a method that improves [11] by adding NACK feedback to inform the downstream nodes of congestion.

A forwarding strategy can dynamically select one or more interfaces in the FIB to forward the interest packet. The BestRoute strategy forwards the interest packet using the path available at the lowest routing cost [13]. In [14], the authors propose a forwarding strategy based on calculating the weight value of the number of pending interests corresponding to each output interface of FIB. In [15], the authors design adaptive forwarding to retrieve data through optimally performing paths, quickly detect, and recover from packet transmission problems. In [16], the authors propose an adaptive SRTT-based forwarding strategy (ASF). The ASF periodically measures the SRTT of an adjacent node at each node, arranges the transmittable nodes based on this, selects the node with the lowest SRTT, and transmits the interest packet. If a problem such as a timeout occurs, the node in which the problem occurs is penalized and sent to the end of the sequence.

In this paper, we design an intelligent forwarding strategy for congestion control using Q-learning and LSTM in NDN. In the first phase, we predict the change in the PIT entry rate in the next time step through time series prediction based on a pretrained LSTM model. In the second phase, based on the predicted PIT entry rate, an appropriate alternative route is selected through Q-learning in congestion situations.

3. Proposed Method: IFS-QLSTM

3.1. Basic NDN Forwarding Mechanism. The NDN node is composed of three elements: PIT, CS, and FIB. The PIT records where the interest packet originated from when it came into the node and tells where to return the data packet when it comes in. CS is a place to temporarily store data and is a feature of the NDN nodes. FIB is a place where nodes that can go to each prefix are recorded, and when an interest packet comes in, it searches for the prefix and informs the path to go next.

Figure 1 illustrates the forwarding process of NDN. When an interest packet arrives, the NDN node first searches for the CS and then returns it to the incoming interface if there is matching data. If not, it goes to PIT and lookup. If duplicated data is already requested in the PIT, the path on which the interest packet came in is added. However, if not,

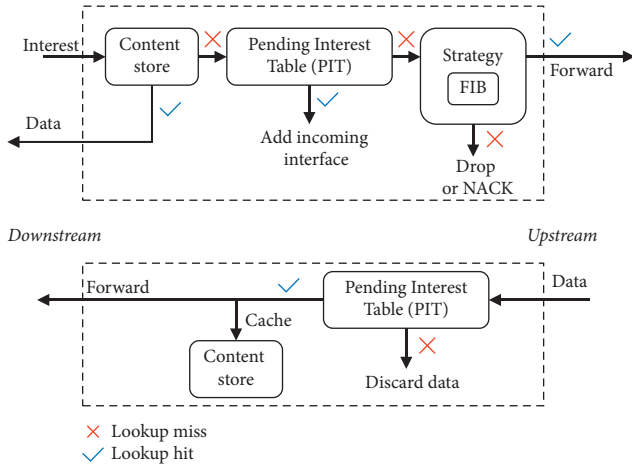


FIGURE 1: Interest and data processing in NDN node.

it is recorded in the PIT and sent to the FIB. Finally, in FIB, if there is a node that can search for the name of a received packet and transmit it, it transmits it to the optimal path according to the forwarding strategy. However, if there is no transmittable node, the packet is discarded. Next, when a data packet arrives at the NDN node, it first searches the PIT and checks whether there is a request for the received data. If there is a request, it returns through the recorded reverse path, and if there is no request, the incoming data packet is discarded. Data that comes in before being transmitted over the reverse path is retrieved from the CS, and if there is no cached data, it is stored in the CS so that it can quickly respond to the next request.

3.2. Proposed System Model. The system model of an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking is shown in Figure 2. When the NDN node receives the interest packet, it checks whether there is a matching name in CS and PIT, and if not, the FIB searches the outgoing interface and forwards it to the interface chosen by the forwarding strategy. As shown in Figure 2, the IFS-QLSTM proceeds in the same way up to the PIT but shows the difference in the forwarding strategy to bypass the congested nodes. First, the PIT entry rate of neighboring nodes is predicted through pretrained LSTM using the PIT entry rate of the nodes obtained from the data packet. After that, congestion is detected using the predicted value as the state of Q-learning, and an appropriate alternative path is selected as the action and forwarded.

3.3. Pretrained LSTM. NDN’s PIT is a place to record the incoming interface of the received interest packet, so it can predict the amount of returned data. Since it changes with time, it can also be viewed as time-series data. Thus, if we train using the LSTM model, a deep learning that is widely used for predicting time series data, we can predict the new PIT entry rate in the next time interval. Based on this data, it is possible to know the arrival rate of data packets, and the congestion can then be forecast in a timely manner.

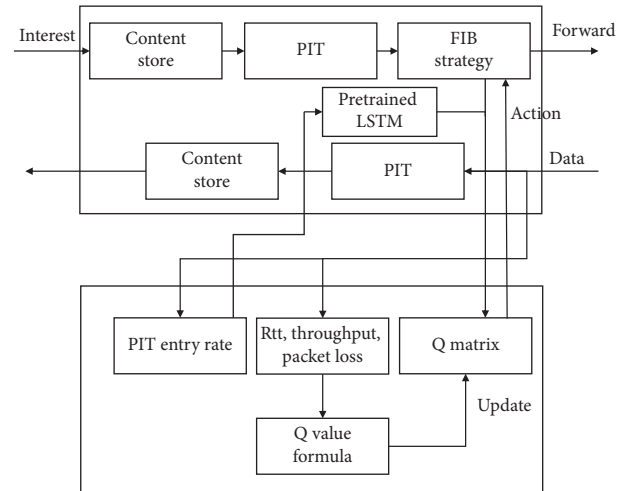


FIGURE 2: System model of the IFS-QLSTM.

In advance, the PIT entry rate for each node is measured and normalized to use as an input to the LSTM model. Then, as shown in Figure 3, we train the LSTM model to predict time $t + n + 1$ by inputting time t through $t + n$. Finally, the trained model is saved and used to predict the next time step PIT entry rate of the neighboring nodes.

3.4. Q-Learning Structure

3.4.1. State. Reinforcement learning agents must be given enough information to accurately know their current state. However, in the case of the Q-learning used in this paper, if you use too much state to generate the q-table consisting of states and actions, it may cause problems with the q-table by becoming too complicated. Therefore, it has to choose an appropriate state variable that can represent the current state. In this paper, it shows the two following state variables: First, it is necessary to know where to make a decision, so the current node that has received the interest packet is set as the state. Second, to know the congestion condition of the nodes that can be transmitted by the current node, the predicted value of the PIT entry rate of the transmittable nodes using a pretrained LSTM model is set as the state. Based on these two states, it is possible to know where the agent is currently located and the congestion condition of the neighboring nodes.

3.4.2. Action. Since the IFS-QLSTM is a method of transmission by selecting an appropriate path for a congestion condition, when the NDN node receives an interest packet, one of the neighboring nodes that can be transmitted is selected as an action.

3.4.3. Reward. Since the reward is an indicator of the direction of training, the definition of reward is important in reinforcement learning. Therefore, to train in the desired direction, it is necessary to define a reward suitable for the training direction. Thus, the reward is defined as follows:

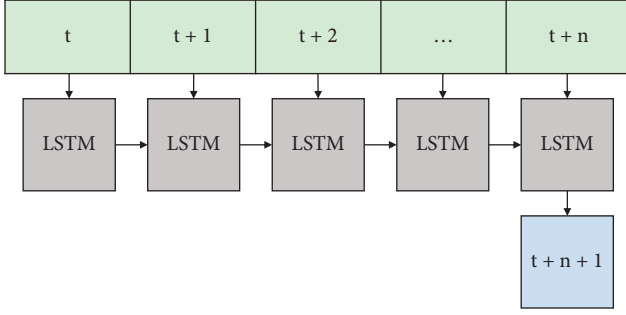


FIGURE 3: Pretrained LSTM

$$\text{reward} = \alpha \cdot \text{throughput}_n - \beta \cdot \text{packet loss}_n - \gamma \cdot \text{RTT}_n, \quad (1)$$

where N represents a node, and α , β , and γ are the weight values for controlling the throughput, packet loss, and RTT, respectively. Throughput represents the number of packets processed per second by node N , and packet loss represents the number of packets discarded per second by node N . RTT represents the time when a packet is transmitted and received by node N . We thought that if we set only the packet loss as a reward, it may be trained not to consider packet transmission time or throughput, although congestion paths were well avoided. Therefore, we designed the reward in the direction of increasing packet throughput and reducing packet loss and RTT while avoiding congested paths.

3.4.4. Q Value Update. In this paper, the Q value was updated every second. The update formula of the Q value is the general Q-learning update formula as shown in Equation (2). $Q(s, a)$ represents the Q value when action A is performed in state S . The value of r is the reward when action A is taken in state S . The discount factor, γ , is a number between 0 and 1 which has the effect of valuing rewards received earlier as higher than those received later.

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right). \quad (2)$$

3.5. Q-Learning-Based Forwarding Strategy. Figure 4 shows the Q-learning packet transmission process when an interest packet is received by the NDN node. When the interest packet arrives, the NDN node first checks the CS and PIT for a matching name; if a matching name does not exist, it looks up the name in the FIB. If there is a matching name in the FIB, the PIT entry rate of the nodes corresponding to the matching name (transmittable nodes from the current node) is predicted using the pretrained LSTM. If not, the interest packet is discarded. After that, it is forwarded to the most optimal path through Q-learning. Specifically, the predicted PIT entry rate and the current node are used as the state of Q-learning to obtain the Q values of the transmittable nodes

from the q table. Next, a random value between 0 and 1 is selected, and if it is less than the current epsilon value, the reinforcement learning agent selects the exploration method. The exploration method selects a random node among the remaining nodes except for the node with the highest Q value and forwards the interest packet. The reason for the exploration is that as the path that was not good in the past may improve, always making the optimal decision may not be good for reinforcement learning training, it is a method used to gain various experiences. Next, if the random value is greater than the epsilon value, the exploitation method is selected. This method selects the node with the largest Q value among the transmittable nodes in the q table and forwards the interest packet. In this way, exploration and exploitation are performed according to the epsilon value, but if the exploration is excessive, the performance is reduced, so the epsilon value is set to decrease over time.

4. Simulation and Analysis

4.1. Simulation Environment. In this section, we implemented by using the open-source ndnSIM [17, 18], an NS-3 based simulator that was developed for NDN. We then evaluated the performance of the IFS-QLSTM through simulation results. Two evaluation metric criteria were selected to quantitatively evaluate the effectiveness of our method. The first criterion was the rate of InData as an indicator for evaluating the utilization of the bottleneck links and alternate links. InData represents the amount of incoming data in the node and guarantees that this amount of data packets was actually transmitted during the congestion. The second criterion is the packet drop rate. If the packet drop rate of IFS-QLSTM is low, it can be seen that IFS-QLSTM effectively mitigates packet dropping.

The topology used in the experiment is shown in Figure 5. In the topology, the consumer (Node0) forwards an interest packet, and the producer (Node8) returns data matching the requested interest packet. The link bandwidth and delay in this topology are set to 10 Mbps and 10 ms, respectively. In our experiment, we cause congestion by setting a specific link bandwidth as low as 1 Mbps according to the requirements of various congestion scenarios.

Next, the Q-learning parameters of the IFS-QLSTM are as follows. First, a random variable (between 0.0 and 1.0) was assigned for comparison with epsilon. The epsilon value, which determines exploration and exploitation, decreased with time until it reached 0.01. The discount factor, which is the weight to control the future compensation compared to the current compensation, was set to 0.9. In the case of LSTM, Adam was used as the optimizer, and the learning rate was set to 0.001. We chose BestRoute and ASF because BestRoute is a basic NDN forwarding method used as a comparison algorithm in many papers, ASF is a more advanced forwarding algorithm, and the main reason is that both methods are verified algorithms. Therefore, we simulated them and compared them with the IFS-QLSTM.

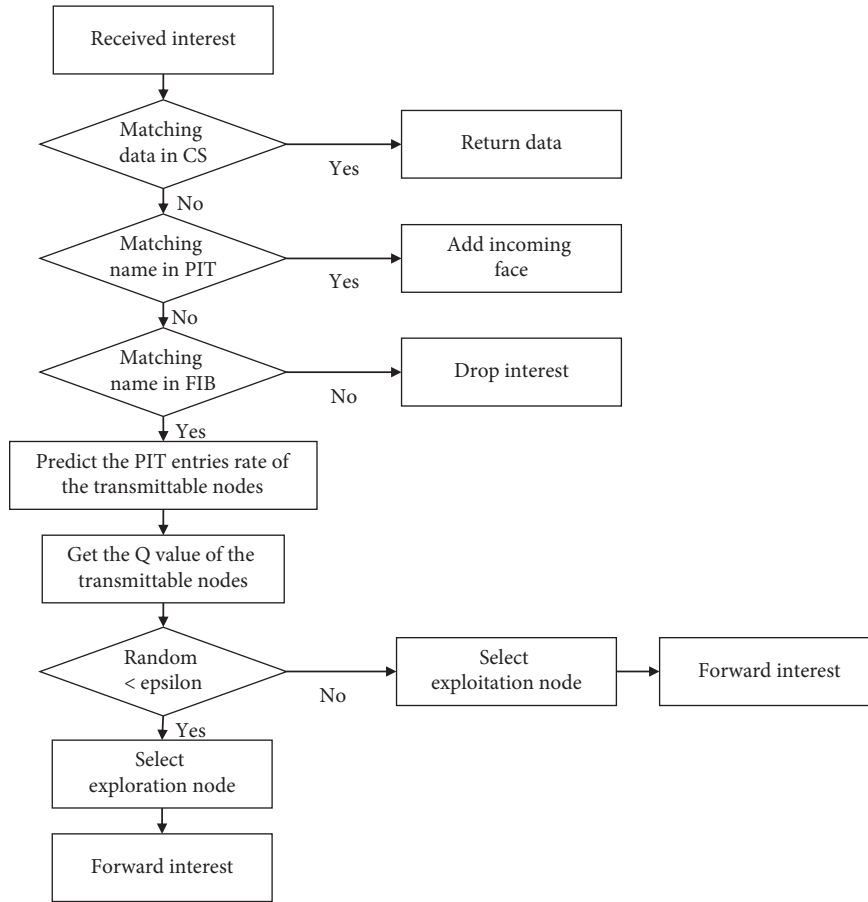


FIGURE 4: Q-learning forwarding strategy when the NDN node receives the interest packet.

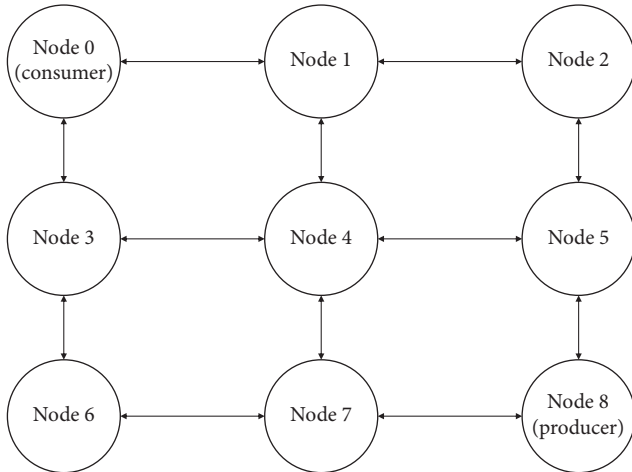


FIGURE 5: Topology used in the experiment.

4.2. Performance Analysis

4.2.1. Low-Level Congestion. We designed a 3 x 3 grid topology as shown in Figure 5. N5-N8 in Figure 6(a), N1-N2 and N4-N5 in Figure 6(b), and N1-N4, N4-N5, and N5-N8 in Figure 6(c) have a bandwidth of 1 Mbps, while the rest of the link bandwidth was connected at 10 Mbps. The link delay is commonly set to

10 ms. Therefore, as shown in Figures 6(a)–6(c), there are paths without bottleneck links: N0-N3-N6-N7-N8.

The graph in Figure 7 shows the average of the data packets received per second from the consumer in the three cases of Figures 6(a)–6(c). The IFS-QLSTM showed almost similar performance to that of ASF and a 17.3% higher data receiving rate than the BestRoute. The graph in Figure 8 is the average of the total packet drops in Figures 6(a)–6(c). Since there are 35,750 packets transmitted, ASF, BestRoute, and the IFS-QLSTM show packet drop rates of 0.07%, 15.9%, and 0.09%, respectively. Like the data receiving rate, the packet drop rate is similar to ASF and is 15.81% lower than BestRoute.

In detail, looking at the data rates in Figures 6(a)–6(c), you can see how each method transmits the packet. In the case of ASF, the SRTT of the adjacent nodes is measured periodically, so it quickly detects bottleneck links, finds alternate links, and sends packets to show a high InData rate. In the case of BestRoute, an alternative route is selected only when the FIB is updated, but because the update is not performed frequently or is not performed at the optimal time, packets are transmitted through the bottleneck link to show a low InData rate. Finally, the proposed method has a slightly lower initial InData rate because it transmits even paths with a low Q value due to exploration at the beginning. However, through reward, the model trains the PIT entry rate that does not cause the packet drop and the appropriate

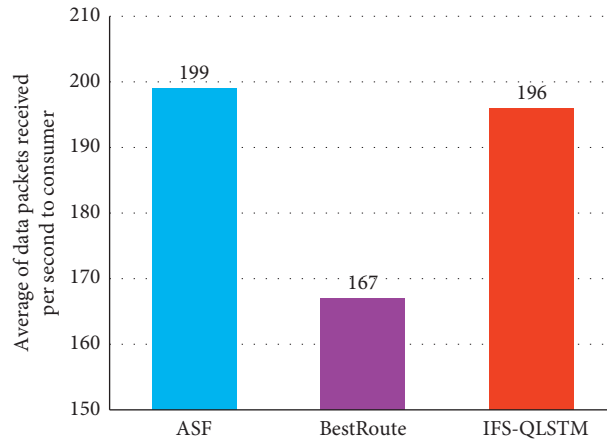


FIGURE 6: Comparison of the InData rate and the packet drop rate for each node. (a) Bottleneck link: N5-N8. (b) Bottleneck link: N1-N2 and N4-N5. (c) Bottleneck link: N1-N4, N4-N5, and N5-N8.

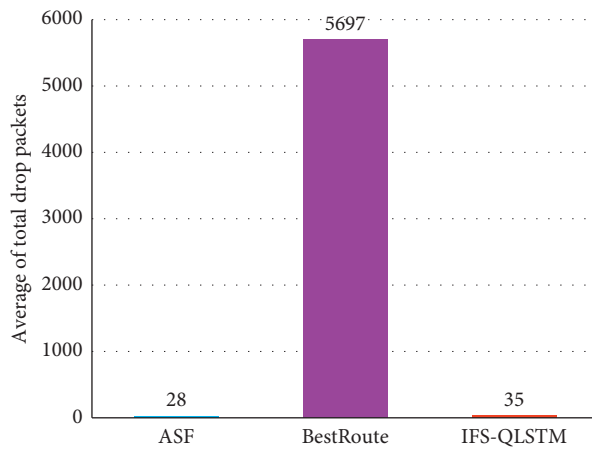
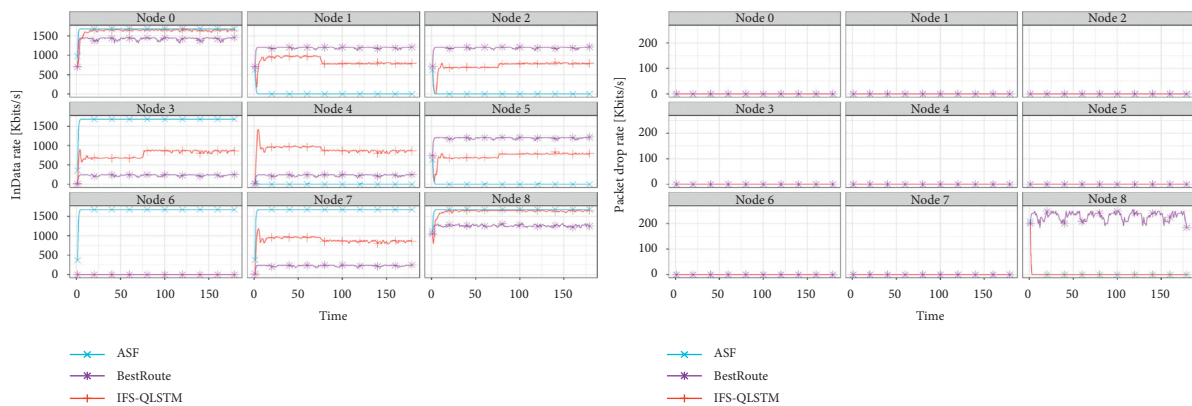


FIGURE 7: Comparison of the average received data rate between IFS-QLSTM and ASF and BestRoute.



(a)

FIGURE 8: Continued.

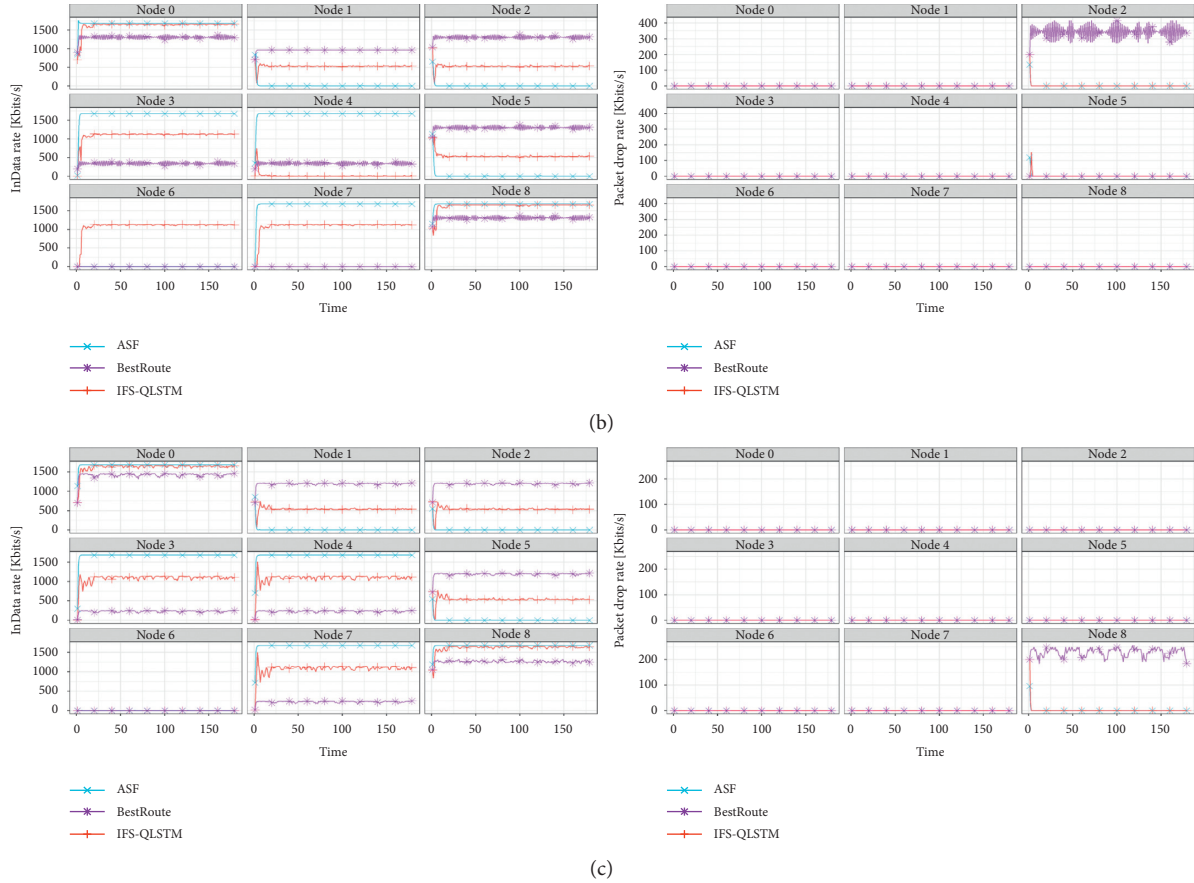


FIGURE 8: Comparison of the average packet drop between IFS-QLSTM and ASF and BestRoute.

amount of transmission according to the PIT entry rate for each node. Through this, the packet is properly divided into a bottleneck path and an alternate path and transmitted. Therefore, it shows an InData rate similar to ASF. In addition, looking at the packet drop rate in Figure 6, BestRoute cannot find an alternative path, resulting in high packet drops on the bottleneck link. On the other hand, in the ASF and IFS-QLSTM, a packet drop occurs briefly at the beginning, and a packet drop does not occur after finding an alternative path.

4.2.2. High-Level Congestion. We designed a 3x3 grid topology as shown in Figure 5. N5-N8 and N7-N8 in Figure 9(a), N1-N2, N4-N5, and N7-N8 in Figure 9(b), and N1-N2, N3-N6, N4-N5, and N4-N7 in Figure 9(c) have a bandwidth of 1 Mbps, while the rest of the link bandwidth was connected at 10 Mbps. The link delay is commonly set to 10 ms. Therefore, as shown in Figures 9(a)–9(c), the bottleneck links exist no matter which path from the consumer to producer is selected.

The graph in Figure 10 shows the average of the data packets received per second from the consumer in the three cases of Figures 9(a)–9(c). IFS-QLSTM showed 15.3% and 21.1% higher data rates than ASF and BestRoute. The graph in Figure 11 is the average of the total packet drops in Figures 9(a)–9(c). Since there are 35,750 packets transmitted, ASF, BestRoute, and the IFS-QLSTM show packet drop

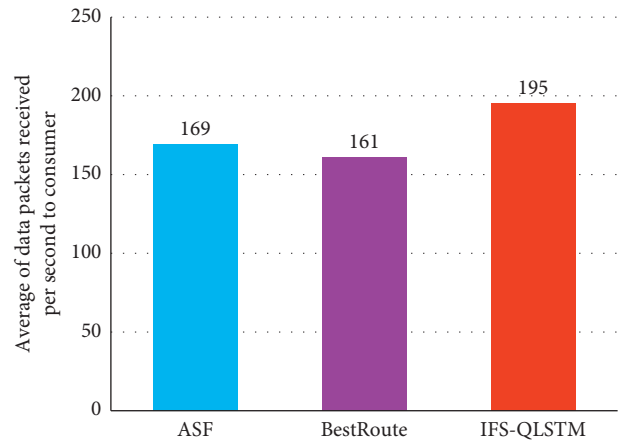


FIGURE 9: Comparison of the InData rate and the packet drop rate for each node. (a) Bottleneck link: N5-N8 and N7-N8. (b) Bottleneck link: N1-N2, N4-N5, and N7-N8. (c) Bottleneck link: N1-N2, N3-N6, N4-N5, and N4-N7.

rates of 14.7%, 18.8%, and 0.16%, respectively. In the case of this experiment, IFS-QLSTM shows overall higher performance than ASF and BestRoute.

In detail, by looking at the InData rate and the packet drop rate in Figures 9(a)–9(c), you can see how each method transmitted the packet and where it was dropped. In the case

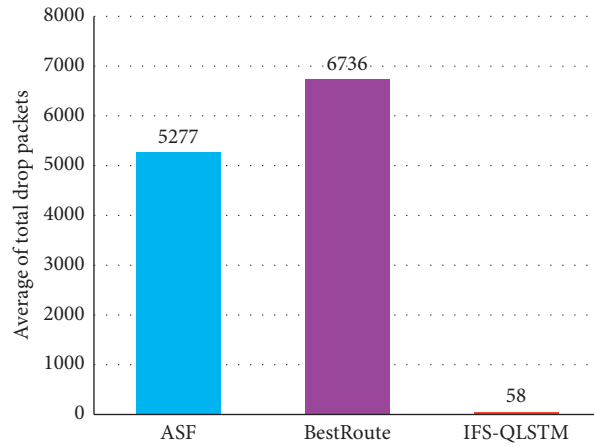
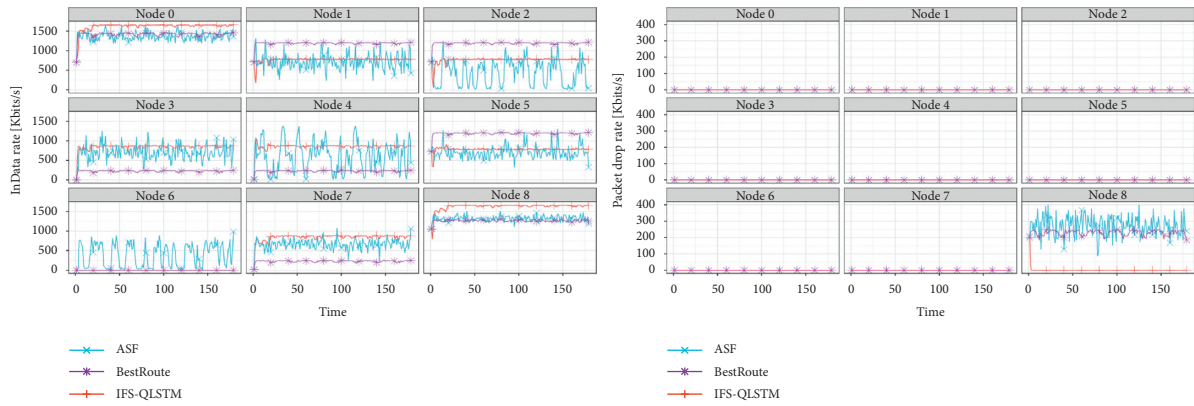
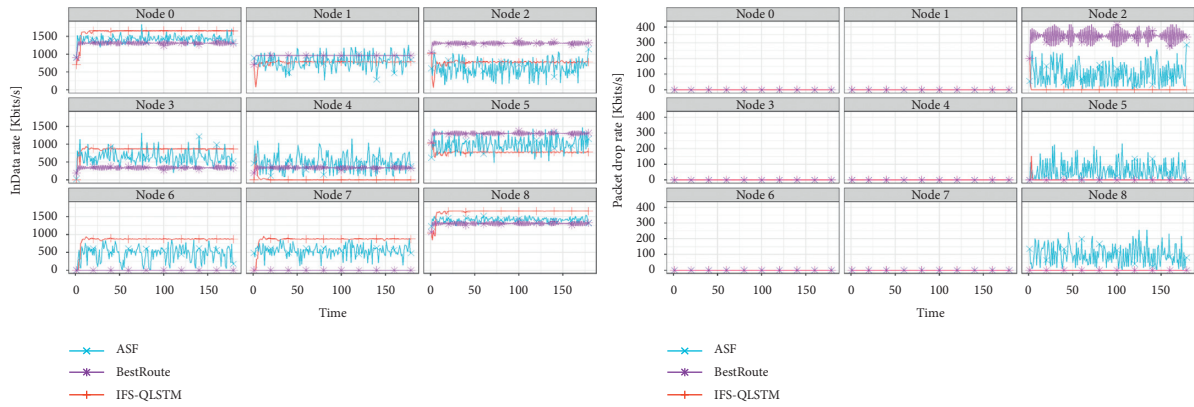


FIGURE 10: Comparison of the average received data rate between IFS-QLSTM and ASF and BestRoute.



(a)



(b)

FIGURE 11: Continued.

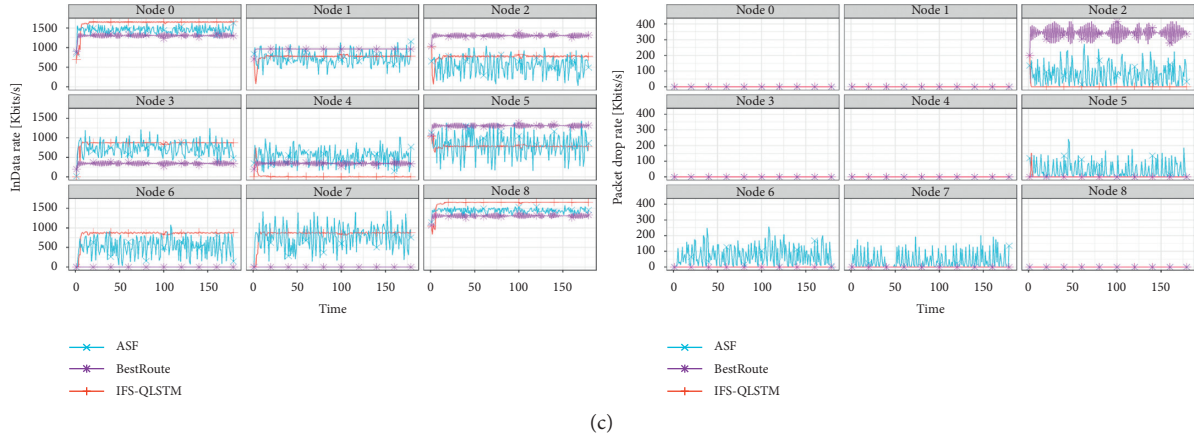


FIGURE 11: Comparison of the average packet drop between IFS-QLSTM and ASF and BestRoute.

of ASF, unlike previous cases, it shows poor performance. The reason is that if the adjacent nodes have the same SRTT, the path is not updated in time, and thus packets are transmitted over the bottleneck link. Therefore, it shows a low InData rate. Unlike the previous case, many packet drop rates occur in the bottleneck link because the alternative path cannot be found properly. In the case of BestRoute, as before, due to the slow FIB update, a low InData rate and a high packet drop rate are shown. In the case of IFS-QLSTM, as described above, since an alternative path is selected and transmitted according to the PIT entry rate of the neighboring node, the stable packet transmission is shown even in the bottleneck link. Therefore, by achieving a high InData rate and low packet drop rate, we prove that the performance is more effective than those of ASF and BestRoute.

5. Conclusions

In this paper, we propose IFS-QLSTM, an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking. The proposed method first trains the LSTM model using the PIT entry rate which can be used as a congestion detection indicator by knowing the amount of data to be returned in the future. After this step, Q-learning detects the congestion of the adjacent node through the PIT entry rate predicted by the trained LSTM model and forwards it to the appropriate path. As a result of the simulation, it was verified that IFS-QLSTM has a high data rate and low packet drop compared to BestRoute and ASF by selecting the bottleneck link and the alternative link well and transmitting the packet. Therefore, it is shown that the proposed method is efficient and reliable. This suggests that there is potential for it to be used as an effective congestion control algorithm for applications to which NDN will be applied in the future.

Future work will focus on evaluating our approach in various topologies and linking it with window-based congestion control algorithms. This approach will lead to improving the congestion control performance of IFS-QLSTM.

Data Availability

The data used to support the findings of this study have not been made available because this work has been supported by the Korean government and the data cannot be publicly open.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Promotion (no. 2015-0-00816) and by the Korea University of Technology and Education (KOREATECH), Education and Research Promotion Program (2021).

References

- [1] G. Xylomenos, C. Ververidis, V. Siris et al., "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, 2013.
- [2] L. Zhang, D. Estrin, J. Burke et al., "Named data networking (NDN) project," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1892, p. 1, 2010.
- [3] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014.
- [4] Named Data Networking Project, <https://named-data.net/>.
- [5] O. Akinwande, "Interest forwarding in named data networking using reinforcement learning," *Sensors*, vol. 18, no. 10, p. 3354, 2018.
- [6] D. Lan, X. Tan, J. Lv, Y. Jin, and J. Yang, "A deep reinforcement learning based congestion control mechanism for NDN," in *Proceedings of the 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019.
- [7] Z. Rezaeifar, J. Wang, H. Oh, S.-B. Lee, and J. Hur, "A reliable adaptive forwarding approach in named data networking," *Future Generation Computer Systems*, vol. 96, pp. 538–551, 2019.

- [8] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: design and evaluation of an interest control protocol for content-centric networking," in *Proceeding Of IEEE INFOCOM Workshop On Emerging Design Choices in Name Oriented Networking (INFOCOM NOMEN)*, 2012.
- [9] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Receiver-driven interest control protocol for content-centric networks," in *Proceeding of the ACM SIGCOMM Workshop on Information Centric Networking (ICN)*, Helsinki, Finland, August 2012.
- [10] L. Saino, C. Cocora, and G. Pavlou, "CCTCP: a scalable receiver-driven congestion control protocol for content centric networking," in *Proceeding of the IEEE ICC'13*, Sydney, Australia, September 2013.
- [11] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for CCN communications," in *Proceeding of the IEEE INFOCOM NOMEN Workshop*, Orlando, FL, USA, March 2012.
- [12] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," in *Proceeding of the ICN'13*, Hong Kong, China, August 2013.
- [13] A. Afanasyev, J. Shi, B. Zhang et al., *NFD Developer's Guide Technical Report NDN-0021*, Department of Computer Science, University of California, Los Angeles, CA, USA, 2014.
- [14] G. Carofiglio, M. Gallo, L. Muscariello et al., "Optimal multipath congestion control and request forwarding in information-centric networks: protocol design and experimentation," *Computer Networks*, vol. 110, 2016.
- [15] C. Yi, A. Afanasyev, I. Moiseenko et al., "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, 2013.
- [16] V. Lehman, A. Gawande, B. Zhang et al., "An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN," in *Proceedings of the IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, Passau, Germany, June 2016.
- [17] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, ndnSIM 2.0: A New Version of the NDN Simulator for NS-3 NDN, Technical Report NDN-0028, 2015.
- [18] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: an open-source simulator for ndn experimentation," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, 2017.

Research Article

Evaluation and Analysis of Traditional Physical Training by Using Mobile Edge Computing and Software-Defined Networking

Wenwen Pan ¹, Jianzhi Wang ² and Jingsheng Ji¹

¹Qiqihar University, Qiqihar 161006, China

²Mudanjiang Medical University, Mudanjiang 157011, China

Correspondence should be addressed to Jianzhi Wang; wangjianzhi@mdjmu.edu.cn

Received 11 January 2021; Revised 8 February 2021; Accepted 18 February 2021; Published 17 March 2021

Academic Editor: Jianhui Lv

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

The body health plays an important metric in people's everyday life, and it directly determines whether people have the ability to preferably contribute to the society. In fact, the physical training is a universal sport to enhance the body health. Therefore, the evaluation and analysis of physical training become particularly significant. With the rapid development and emerging of new techniques and networking paradigms, the traditional offline physical training evaluation and analysis cannot be performed well. Instead, this paper uses Mobile Edge Computing (MEC) and Software-Defined Networking (SDN) to implement the evaluation and analysis of physical training, shortened for MSPT, where MEC is the new computing technique and SDN is the new networking paradigm. The proposed MSPT includes two parts. At first, the physical training data from different mobile devices are migrated into the edge server for computing according to the current condition, in which the game theory is used to complete the task scheduling. Then, SDN is responsible for the global scheduling in the centralized control manner, in which the multi-granularity scheduling strategy is used to handle the traffic between the SDN controller and edge computing server. The experiments are driven by OMNet, including three aspects of evaluation, i.e., task offloading of MEC, traffic scheduling of SDN, and performance analysis of physical training, and the results show that the proposed MSPT has better performance than the corresponding baselines.

1. Introduction

The physical training has a huge following around the world and has been referred to as the national pastime since it can improve the body health greatly. According to a reliable survey, the number of persons which participate in the physical training program can reach 83.6% of the world's population [1], which further highlights that the physical training is an absolutely universal sport and plays an important role in people's everyday life. In spite of this, the improper physical training perhaps causes the accidental injury because the nonprofessional operations are usually adopted, especially for adolescents and elder population [2]. According to a rough report from the American Center for the Study of the Elderly, the proportion of which the old people are not obtained the proper guidance of physical training exceeds 28.9% [3]. Therefore, the correct and efficient conduction of physical training is very necessary. In

other words, the evaluation and analysis of physical training become particularly significant.

However, it is far from enough to make the effective evaluation and analysis of physical training if only the traditional offline physical training way is used. Instead, it needs some new techniques or/and networking paradigms as the auxiliary tools to help complete the effective evaluation and analysis of physical training. As everyone knows, Mobile Edge Computing (MEC) [4, 5] is a new technique, and its purpose is to effectively solve problems such as latency and network load. In addition, MEC can put some complex tasks at the edge computing servers for computing, in order to save response time and thus guarantee the real-time evaluation and analysis. Analogically, in this paper, there is a lot of traffic related to evaluation and analysis of physical training, and this traffic can be regarded as those complex tasks. In particular, the current physical training usually uses the mobile devices to collect the training data. Thus, MEC

can offload the collected data from these mobile devices into the edge computing servers for data computing.

In spite of this, the task offloading process of MEC can be covered, which has an important influence on the global evaluation and analysis of physical training. For this purpose, it needs to seek a networking paradigm to cover the whole network view. According to such demand, Software-Defined Networking (SDN) [6–8] is a very satisfactory candidate, which is a new future Internet networking paradigm with some special abilities. For example, SDN can grasp the global network view. According to such ability, the task offloading process of MEC can be taken in a glance. Besides, under SDN environment, all operations are performed in the centralized control manner; in other words, the evaluation and analysis of physical training are integrated.

Furthermore, as the introduction of SDN, it involves a necessary communication between the SDN controller and edge computing server, including traffic scheduling and message transmission (e.g., control signal). In order to improve the bandwidth utilization and save communication delay, the traditional object-level traffic scheduling [9, 10] between the SDN controller and edge computing server is discarded. Instead, this paper uses the multigranularity scheduling strategy to address different scenarios.

With the above consideration, this paper makes the evaluation and analysis of physical training by using MEC and SDN, called MSPT. The corresponding contributions are threefold. (1) In MEC, the game theory-based task offloading is performed. (2) In SDN, the multigranularity scheduling strategy between the SDN controller and edge computing server is devised. (3) The simulation experiments are implemented, including some main metrics' evaluation and analysis.

The rest paper is organized as follows. Section 2 reviews the related work, including task offloading in MEC and traffic scheduling in SDN. Section 3 presents the game theory-based task offloading. Section 4 devises the multigranularity scheduling strategy. Section 5 reports some major results. Section 6 concludes this paper.

2. Related Work

2.1. Task Offloading in MEC. There have lots of proposals on task offloading in MEC. In [11], there were three offloading options, i.e., nearest edge server, adjacent edge server, and remote cloud. It proposed a Reinforcement Learning (RL)-based algorithm to make the optimal offloading decision for minimizing system cost, including energy consumption and computing time delay. In [12], a distributed many-to-many matching model was constructed to capture the interaction between mobile tasks and edge nodes, with the consideration of their diverse resource requirements and availabilities. It designed both distributed and centralized stable matching-based algorithms to jointly offload the tasks to edge nodes and determine their payments. In [13], a multiuser offloading scenario with intensive deployment of edge servers was considered. It divided the offloading process into two stages, i.e., data transmission and

computation execution, in which the existence of Nash equilibrium was proven by using the noncooperative game method. In [14], the offloading decision problem was formulated as a 0-1 nonlinear integer programming problem under the constraints of channel interference threshold and the time deadline. Through the classification and priority determination for the mobile devices, a reverse auction-based offloading method was proposed to solve this optimization problem for energy efficiency improvement. In [15], the offloading problem was formulated as the joint optimization of computation task assignment and CPU frequency scaling, in order to minimize a tradeoff between task execution time and mobile energy consumption. It proposed a light-weight algorithm by using the Markov approximation technique to converge to a bounded near-optimal solution. In [16], a new task offloading scheme by considering the challenges of future edge, fog, and cloud computing paradigms was devised. To provide an effective solution toward an appropriate task offloading problem, it focused on two cooperative bargaining game solutions. The first method was used for time-sensitive offloading services, and the second method was applied to ensure computation-oriented offloading services. In [17], the computation offloading framework was proposed by considering the quality of service, server resources, and channel interference, where the offloading decision was made based on the beneficial degree of computation offloading measured by the total cost of the local mobile devices. In [18], an intelligent computation offloading in combination with artificial intelligence technology was proposed. It devised a task migration algorithm based on task prediction according to the data size of computation task from mobile users and the performance features of edge computing nodes. In [19], the deep RL was proposed to solve the offloading problem of multiple service nodes for the cluster and multiple dependencies for mobile tasks in large-scale heterogeneous MEC. The proposed strategy had good performance in terms of energy consumption, load balancing, latency, and average execution time by using iFogSim and Google cluster trace. In [20], an optimal auction method for delay and energy constrained task offloading was proposed by a pair of deep neural networks. It maximized the profit of the edge servers while satisfied the task processing delay and energy consumption constraints of the mobile devices. In [21], a distributed task unloading strategy to the low load base station group under MEC environment was proposed, including two parts. At first, the communication resource, computing resource, and task queue of the low load base station group were modeled to quantify the energy cost in the process of task unloading. Then, the potential game model was used to solve the problem of distributed task unloading. In [22], the collaborative task offloading and data caching models were proposed to reduce the overall latency of all mobile devices. It used the Lyapunov online algorithm to perform the joint task offloading and dynamic data caching strategies for computation tasks or data contents. In [23], the data quality-aware offloading sequential decision making problem by adopting the

principles of Optimal Stopping Theory (OST) was proposed to minimize the expected processing time. Besides, a variety of OST stochastic models and their applications to the offloading decision making problem were investigated.

2.2. Traffic Scheduling in SDN. There have also some proposals on traffic scheduling in SDN. For example, in [24], a nonsupervised deep learning-based routing strategy running in the SDN controller was proposed to address the impact on explosive growth in network traffic. In [25], a software-defined 5G architecture was first introduced for end-to-end reliability of mission-critical traffic. Then, a mathematical framework was constructed to model the process of critical session transfers. Finally, a hardware implementation was conducted to study the practical effects of supporting mission-critical data at the core network level. In [26], an energy flow scheduling and routing mechanism in SDN was proposed to optimize the energy consumption both in link and switches, which could minimize traffic energy in time dimension and improve utilization of switches. In [27], a traffic engineering scheme based on reinforcement learning was proposed, which constructed and solved a simple linear programming problem to reroute the flows selected by reinforcement learning to balance utilization of link in the network. In [28], a multimedia traffic control mechanism based on deep reinforcement learning was proposed to achieve the multimedia traffic control and avoid the additional mathematical computation. In [29], an energy-saving traffic scheduling algorithm was proposed to minimize the energy in hybrid software-defined wireless rechargeable sensor networks, which adequately used the features in SDN, such as direct control on SDN nodes and indirect control on normal nodes. In [30], a novel SDN-based architecture was devised to manage large-scale networks, which could guarantee quality of service and manage routing. The experiments showed that the proposed method could efficiently manage the hybrid architectures and reach good quality of service. In [31], the authors first formulated minimization time problem as an optimization problem and then proposed two polynomial-time algorithms to solve the optimization problem, which could test feasibility and obtain a solution, respectively. In [32], to minimize traffic load in software-defined wireless sensor networks, the authors first formulated the traffic load minimization problem as an optimization problem related to the optimization of relay sensor node and splitting flow transmission and then designed a Levenberg–Marquardt algorithm to solve the problem. In [33], a path determination and traffic scheduling strategy for SDN was proposed to ensure the bandwidth of quality of service to support businesses in network, which was able to obtain the optimal path with the port queues on switches.

Different from the abovementioned research studies, this paper uses the game theory to address the task offloading in MEC. In addition, this paper also designs the multi-granularity scheduling strategy in SDN.

3. Task Offloading between Mobile Device and Edge Computing Server

3.1. Mathematical Modeling. The task offloading in MEC needs to offload some complex tasks from the mobile devices into the edge computing server for computing through the wireless base station. Different from the traditional task offloading model, this paper uses SDN to manage all traffic. As depicted in Figure 1, the network model of MSPT consists of four roles, i.e., mobile device, wireless base station, edge computing server, and SDN controller. Among them, the path between mobile device and wireless base station supports the 5G communication, and that between wireless base station and edge computing server supports the optical fiber communication. Therefore, the proposed network model can achieve the quickly interactive response and guarantee the low delay.

The process of task offloading inevitably involves the physical training data transmission. Suppose that there are M wireless channels between mobile device and wireless base station and there are N mobile devices, and for the arbitrary mobile device U_n , it has a given offloading strategy, denoted by a_n . If $a_n = 0$, it means that these tasks will be computed at the local mobile device; otherwise, these tasks in the corresponding mobile device will be migrated into the edge computing server for computing. Let a^* denote the global offloading strategy with respect to all mobile devices, denoted by $a^* = \{a_1, a_2, \dots, a_N\}$, and the upstream link's transmission rate regarding the physical training data is defined as follows:

$$r_n(a^*) = K * \log_2 \left(1 + \frac{S_n}{N_i} \right)$$

$$S_n = q_n * g_n, \quad (1)$$

$$N_i = \sigma + \sum_{i=1}^N q_i g_i.$$

Among them, q_n is the transmission ability, g_n is the transmission gain, and σ is the white noise.

In addition to the network model and communication model, the computing model also plays an important role. For each mobile device, it has the single computing task, denoted by $I_n = (B_n, D_n)$, where B_n is the data size of I_n and D_n is the required number of CPU cycles to complete I_n . For all computing scenarios in MEC, two cases are considered, i.e., local computing and offloading computing.

At first, let t_n^l and e_n^l denote the consumed time and consumed energy, respectively, and they are defined as follows:

$$t_n^l = \frac{D_n}{s_n^l}, \quad (2)$$

$$e_n^l = \gamma_n^l * D_n,$$

where s_n^l is the CPU clock frequency of U_n and γ_n^l is the power dissipation of each CPU clock frequency. Let K_n^l

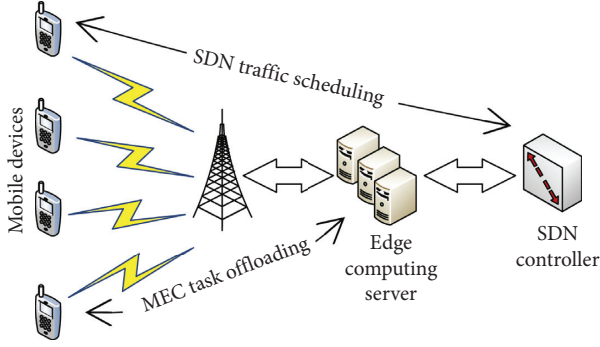


FIGURE 1: The network model of MSPT.

denote the total cost under the local computing case, and it is defined as follows:

$$K_n^l = \lambda_n^t * t_n^l + \lambda_n^e * e_n^l, \quad (3)$$

where λ_n^t and λ_n^e are the weights regarding time consumption and energy consumption, respectively.

Then, let $t_{n,\text{tran}}^c(a^*)$ denote the upstream link's transmission time, $e_{n,\text{tran}}^c(a^*)$ denote the upstream link's energy consumption, $t_{n,\text{exe}}^c$ denote the required computing time, and $e_{n,\text{exe}}^c$ denote the required computing energy, respectively, and they are defined as follows:

$$\begin{aligned} t_{n,\text{tran}}^c(a^*) &= \frac{B_n}{r_n(a^*)}, \\ e_{n,\text{tran}}^c(a^*) &= \frac{q_n * B_n}{r_n(a^*)}, \\ t_{n,\text{exe}}^c &= \frac{D_n}{s_n^c}, \\ e_{n,\text{exe}}^c &= \gamma_n^c * D_n. \end{aligned} \quad (4)$$

Among them, s_n^c is the CPU clock frequency of edge computing server and γ_n^c is the power dissipation of each CPU clock frequency. Here, this paper neglects two kinds of time/energy consumption, i.e., the path from edge computing server to mobile device, used to return the computing results, and the path from wireless base station to edge computing server. The first neglecting is because such path supports the optical fiber communication which has the ultra-low delay. The second neglecting is because the returned data size with respect to the physical training is very small, which takes nearly no delay. Let $K_n^c(a^*)$ denote the total cost under the offloading computing case, and it is defined as follows:

$$K_n^c(a^*) = \lambda_n^t (t_{n,\text{tran}}^c(a^*) + t_{n,\text{exe}}^c) + \lambda_n^e (e_{n,\text{tran}}^c(a^*) + e_{n,\text{exe}}^c). \quad (5)$$

3.2. Game Theory-Based Optimization. For a^* , if most mobile devices select the offloading computing, these transmissions with respect to the physical training data perhaps cause the serious interference among them and thus

generate some negative effects, such as increasing the system overhead including time consumption and energy consumption. Especially, when the total cost is larger than that under the local computing case, the offloading computing makes no sense. Therefore, in order to improve the overall revenue, the offloading computing has to be effective. Furthermore, for a^* , if the total cost under the offloading computing case is smaller than that under the local computing case, the corresponding edge computing server is effective. Mathematically,

$$K_n^c(a^*) < K_n^l. \quad (6)$$

For N mobile devices, the optimization of edge computing belongs to NP-hard problem [34, 35]. Thus, this paper plans to use the game theory [36] to optimize the task offloading regarding N mobile devices. Let a_{N-n} denote the decision strategy set regarding all mobile devices excluding U_n , and it is defined as follows:

$$a_{N-n} = (a_1, a_2, \dots, a_{n-1}, a_{n+1}, \dots, a_N). \quad (7)$$

Let Z_n denote the current total cost, and the purpose of game theory is to minimize $Z_n(a_n, a_{N-n})$. Mathematically,

$$Z_n(a_n, a_{N-n}) = \begin{cases} K_n^l, & a_n = 0, \\ K_n^c(a^*), & a_n > 0. \end{cases} \quad (8)$$

Given Z_n , the task offloading optimization based on game theory is expressed as $\Gamma_{\text{MEC}} = (N, \{a_n\}, \{Z_n\})$. Furthermore, let $a^{**} = (a_1^*, a_2^*, \dots, a_N^*)$ denote the global offloading strategy set when the Nash equilibrium is satisfied [13], and the corresponding mathematical express is shown as follows:

$$Z_n(a_n^*, a_{N-n}^*) \leq Z_n(a_n, a_{N-n}^*). \quad (9)$$

According to the above statements, the game theory-based task offloading strategy in MEC is shown as the following five steps:

Step 1: N mobile devices are initialized by considering that all tasks are completed at the corresponding mobile device.

Step 2: all wireless channels' transmission ability and transmission gain are computed.

Step 3: these wireless channels are arranged according to transmission ability and transmission gain in the descending order, and the wireless channel with the high power is assigned the high priority to make the offloading decision in advance.

Step 4: all offloading decision strategies and their corresponding total costs are computed. If equation (6) is satisfied, the current offloading decision strategy is unchanged. Otherwise, the current offloading decision strategy should be changed.

Step 5: the edge computing server sends the change request to the corresponding mobile device. If the corresponding mobile device receives the request, the offloading decision strategy is changed until all mobile

devices are notified. Otherwise, the current offloading decision strategy is regarded as the optimal solution.

As the proposed network model supports 5G communication and optical fiber communication, the task offloading process usually has no the phenomenon of packet loss.

4. Traffic Scheduling between Edge Computing Server and SDN Controller

Section 3 addresses the task offloading between mobile device and edge computing server. As the task offloading process of MEC has an important influence on the global evaluation and analysis of physical training, it needs SDN to cover the whole network view. In other words, by using SDN, the task offloading process of MEC can be taken in a glance. Therefore, this section pays attention to the traffic scheduling between the edge computing server and SDN controller. Furthermore, in order to improve the bandwidth

utilization and save communication delay, this paper devises the multigranularity traffic scheduling strategy.

This paper considers two granularity change conditions, i.e., traffic aggregation and traffic splitting, which depends on the remaining bandwidth and the current network state. To be specific, if the current network state is smooth and there is enough network bandwidth, the traffic aggregation is approved. Otherwise, the traffic splitting is approved.

At first, for the coarse-grained traffic scheduling, the SDN controller periodically checks whether the network bandwidth satisfies to transmit the physical training data. When the traffic splitting condition is satisfied, the optimization purpose is to maximize the bandwidth utilization while minimize the number of core links. Mathematically,

$$\text{minimize } \max\{UR_l\}, \quad l \in LC, \quad (10)$$

where LC is the number of core links between the SDN controller and edge computing server.

The corresponding constraints are shown as follows:

$$\begin{aligned} U_l - \sum_{k=1}^{A_l} bw_k + \sum_{k=1}^{A_l} \left(\sum_{i=1}^L X_{pki} * C_{lpki} \right) * bw_k = N_l, \quad N_l < T_l < thr, \\ \sum_{i=1}^{Pk} X_{pki} = 1, \\ \sum_{k=1}^{A_l} \sum_{i=1}^{Pk} (lpki * X_{pki} * C_{lpki} * D_{lkj}) < delay_k, \quad X_{pki}, lpki, C_{lpki} \in \{0, 1\}. \end{aligned} \quad (11)$$

Among them, U_l is the used bandwidth link l ; A_l is the aggregation traffic via link l ; N_l is the bandwidth load of link l ; X_{pki} is the result of aggregation; T_l is the total bandwidth of link l ; $lpki$ denotes the possibility of aggregation; C_{lpki} denotes whether the link l is included; and thr is a given parameter to determine whether the congestion of link l exits.

Then, for the fine-grained traffic scheduling, the SDN controller periodically checks the network bandwidth to schedule the physical training traffic from these congestion links to those light links as many as possible. Under this condition, the optimization purpose is defined as follows:

$$\text{minimize } \sum_{k=1}^{A_l} \sum_{i=1}^{Pk} (lpki * X_{pki}). \quad (12)$$

It is obvious that the above two optimization objects belong to the 0–1 Integer Linear Programming (ILP) [37], and they can be solved easily. Thus, this paper gives the corresponding solution discussion no longer.

5. Result Report

The involved simulation parameters are set as follows: $N = 10, 20, 30, 40, 50$, $M = 6$, $B_n = 26.5$ GB, $D_n = 1200$ M Cycle, $g_n = 100$ MW, and the network bandwidth of 5G

communication is 1 Gbps. The hardware environment is shown as follows: CPU, Intel (R) Core (TM) i5-7400 CPU 4.68 GHz; RAM, 64 GB; Hard Disk, 1024 GB; OS, Windows 10 (64 bit). This section evaluates three aspects, including task offloading of MEC, traffic scheduling of SDN, and performance analysis of physical training. For the first part, the average time consumption and the average energy consumption are used as two evaluation metrics, where reference [19] is used as the baseline, called BMEC. For the second part, the average scheduling time and the average bandwidth utilization are used as two evaluation metrics, where reference [27] is used as the baseline, called BSDN. For the last part, the average response time and the average correct guidance rate are used as two evaluation metrics, where reference [38] is used as the baseline, called BPT.

5.1. Task Offloading Evaluation. For different mobile devices, the corresponding average time consumption with respect to MSPT and BMEC is shown in Figure 2. It is obvious that the average time consumption of MSPT is smaller than that of BMEC. In addition, with the increasing of mobile devices, the average time consumption becomes larger and larger, because more physical training data need to be handled. In particular, the average time consumption of MSPT shows the linear growth while that of BMEC shows

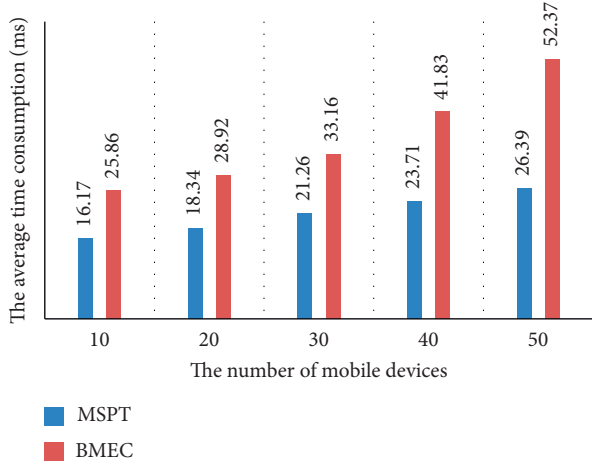


FIGURE 2: The average time consumption in case of task offloading evaluation.

the exponential growth, because BMEC uses the RL to solve the task offloading problem, which involves some iterations and generates more and more time consumption with the increasing of mobile devices.

For different mobile devices, the corresponding average energy consumption with respect to MSPT and BMEC is shown in Figure 3. At first, the average energy consumption of MSPT is smaller than that of BMEC, because MSPT uses the game theory to present a relatively optimal task offloading strategy and the corresponding energy consumption is limited. On the contrary, BMEC introduces the RL to repeatedly use the physical training data, which consumes too many energy inevitably. Then, similar to Figure 2, with the increasing of mobile devices, the corresponding average energy consumption increases. Besides, the increasing speed of MSPT is larger than that of BMEC.

5.2. Traffic Scheduling Evaluation. For MSPT and BSDN, the experimental results on the average scheduling time with respect to different mobile devices are shown in Figure 4. It is observed that MSPT has the significant advantage in terms of the average scheduling time, because it uses the multi-granularity scheduling strategy to handle the physical training data. In addition, the devised multigranularity scheduling strategy in MSPT belongs to a heuristic algorithm, while the traffic scheduling strategy in BSDN is the RL, which belongs to the intelligent algorithm. As a result, it needs too much time to handle the physical training data in terms of BSDN. Furthermore, with the increasing of mobile devices, the average scheduling time rises steadily, because the SDN controller needs to handle more physical training data.

For MSPT and BSDN, the experimental results on the average bandwidth utilization with respect to different mobile devices are shown in Figure 5. It can be found that MSPT has higher average bandwidth utilization than BSDN, because MSPT makes full use of the remaining network bandwidth according to the multigranularity scheduling, including two operations, i.e., traffic

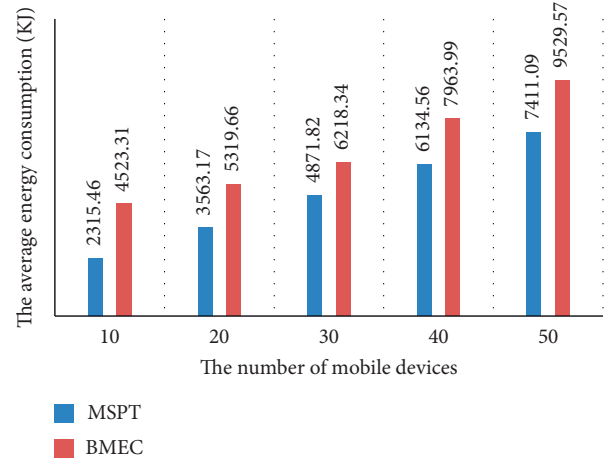


FIGURE 3: The average energy consumption in case of task offloading evaluation.

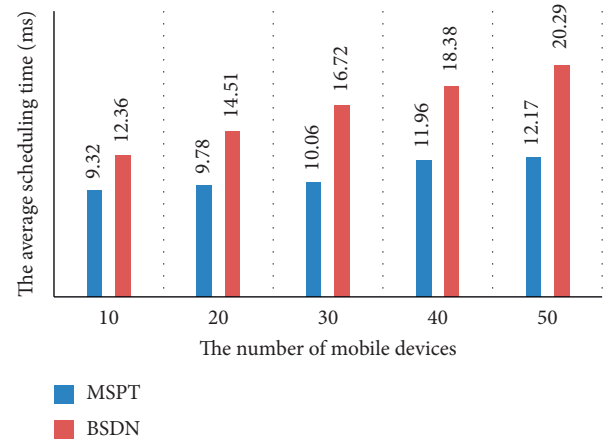


FIGURE 4: The average scheduling time in case of traffic scheduling evaluation.

aggregation and traffic splitting. If the current network bandwidth is sufficient, the traffic aggregation operation is adopted; otherwise, the traffic splitting operation is adopted. Unlike MSPT, BSDN only schedules the physical training data in the same granularity, which causes the congestion in terms of some links. Thus, the network bandwidth in BSDN cannot be leveraged timely.

5.3. Physical Training Evaluation. In this section, 1000 mobile devices and two edge computing servers are considered. For different simulation experiments, the corresponding average response time with respect to MSPT and BPT is shown in Table 1. It can see that the whole average response time of MSPT is smaller than that of BPT. It indicates that the game theory-based task offloading strategy and the multigranularity based traffic scheduling strategy have good optimization effect on the physical training. In particular, it only needs about 42 ms to obtain all physical training results in terms of these 1000 users, which is very satisfactory.

Furthermore, for different simulation experiments, the corresponding average correct guidance rate with respect to

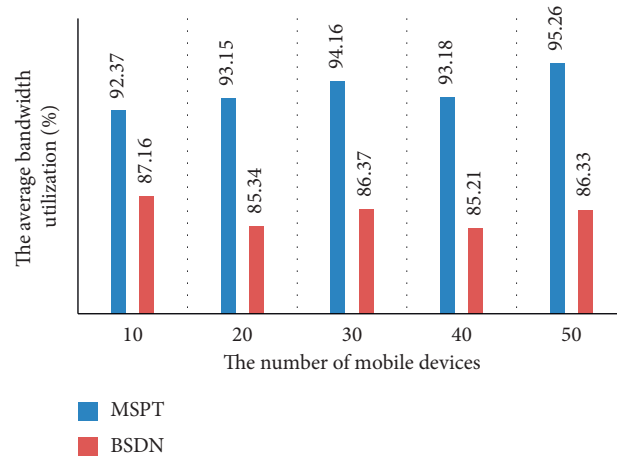


FIGURE 5: The average bandwidth utilization in case of traffic scheduling evaluation.

TABLE 1: The average response time in case of physical training evaluation \ms.

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| MSPT | 42.18 | 43.56 | 42.39 | 42.57 | 42.66 | 43.05 | 43.11 | 42.83 | 42.91 | 42.16 |
| BPT | 61.33 | 62.09 | 62.85 | 61.94 | 62.53 | 62.74 | 61.95 | 62.17 | 63.52 | 62.82 |

TABLE 2: The average correct guidance rate in case of physical training evaluation \%.

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| MSPT | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| BPT | 98.34 | 97.95 | 98.26 | 96.98 | 97.18 | 98.23 | 97.87 | 98.06 | 98.16 | 97.46 |

MSPT and BPT is shown in Table 2. It is obvious that the proposed MSPT has the absolutely higher average correct guidance rate than BPT. Especially, the correct guidance rate of MSPT always reaches 100%, with a strong stability. As a result, SDN- and MEC-based physical training evaluation and analysis are very valuable.

6. Conclusions

The correct and efficient conduction of physical training plays an important role. In this paper, SDN and MEC are used to do the evaluation and analysis of physical training. At first, the task offloading in MEC is addressed, including the mathematical modeling (i.e., network model, communication model, and computing model) and the game theory-based offloading optimization strategy. Then, the traffic scheduling in SDN is addressed, where the multigranularity scheduling strategy is used to handle the traffic between the SDN controller and edge computing server, including traffic aggregation and traffic splitting.

For the proposed MSPT, three parts of experiments are performed: (1) the task offloading evaluation with time consumption and energy consumption consideration; (2) the traffic scheduling evaluation with scheduling time and bandwidth utilization consideration; (3) the physical training evaluation with response time and correct guidance rate consideration. In particular, the correct guidance rate of MSPT regarding the physical training can always reach 100%.

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 Philosophy and Social Science Research Planning Project in Heilongjiang Province (19TYC157), Research Project of Basic Scientific Research Operating Expenses of Provincial Institutions of Higher Learning in Heilongjiang Province (135409235), Heilongjiang Higher Education Teaching Reform Project (SYGY20200801), and University Nuring Program for Young Scholars with Creative Talents in Heilongjiang Porovince (UNPYSCT-2020077).

References

- [1] N. E. Gilhus, "Physical training and exercise in myasthenia gravis," *Neuromuscular Disorders*, 2020, early view.
- [2] Q. Lang, "Research on the integration of computer technology and physical education and training integrating computer algorithms," in *Proceedings of the 2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, pp. 440–445, Atlanta, GA, USA, December 2020.

- [3] J. S. Coom, A. Williams, and J. Radford, "Training health professionals to provide physical activity counselling," *Progress in Cardiovascular Diseases*, 2020, early view.
- [4] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective," *Computer Networks*, vol. 182, pp. 1–24, 2020.
- [5] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions," *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.
- [6] L. Yang, B. Ng, W. K. G. Seah et al., "A survey on network forwarding in Software-Defined Networking," *Journal of Network and Computer Applications*, vol. 176, pp. 1–13, 2021.
- [7] Y. Zhang, L. Cui, W. Wang et al., "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, pp. 101–118, 2017.
- [8] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019.
- [9] K. Nisar, E. R. Jimson, M. H. A. Hijazi et al., "A survey on the architecture, application, and security of software defined networking: challenges and open issues," *Internet of Things*, vol. 12, pp. 1–27, 2020.
- [10] Z. AlSaeed, I. Ahmad, and I. Hussain, "Multicasting in software defined networks: a comprehensive survey," *Journal of Network and Computer Applications*, vol. 104, pp. 61–67, 2017.
- [11] T. Alfakih, M. M. Hassan, A. Gumaei, C. Savaglio, and G. Fortino, "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA," *IEEE Access*, vol. 8, pp. 54074–54084, 2020.
- [12] X. Wang, J. Wang, X. Zhang, X. Chen, and P. Zhou, "Joint task offloading and payment determination for mobile edge computing: a stable matching based approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12148–12161, 2020.
- [13] S. Chen, Y. Chen, X. Chen et al., "Distributed task offloading game in multiserver mobile edge computing networks," *Complexity*, vol. 2020, Article ID 7016307, 14 pages, 2020.
- [14] L. Li, X. Zhang, K. Liu et al., "An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing," *Mobile Information Systems*, vol. 2018, Article ID 7646705, 13 pages, 2018.
- [15] W. Zhou, W. Fang, Y. Li et al., "Markov approximation for task offloading and computation scaling in mobile edge computing," *Mobile Information Systems*, vol. 2019, Article ID 8172698, 12 pages, 2019.
- [16] S. Kim, "New application task offloading algorithms for edge, fog, and cloud computing paradigms," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8888074, 14 pages, 2020.
- [17] N. Shan, Y. Li, and X. Cui, "A multilevel optimization framework for computation offloading in mobile edge computing," *Mathematical Problems in Engineering*, vol. 2020, Article ID 4124791, 17 pages, 2020.
- [18] Y. Miao, G. Wu, M. Li et al., "Intelligent task prediction and computation offloading based on mobile-edge cloud computing," *Future Generation Computer Systems*, vol. 102, pp. 925–931, 2019.
- [19] F. Lu, C. Gu, F. Luo et al., "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 102, pp. 847–861, 2019.
- [20] F. Mashhadi, S. A. Salinas Monroy, A. Bozorgchenani et al., "Optimal auction for delay and energy constrained task offloading in mobile edge computing," *Computer Networks*, vol. 183, pp. 1–10, 2020.
- [21] Y. Li and C. Jiang, "Distributed task offloading strategy to low load base stations in mobile edge computing environment," *Computer Communications*, vol. 164, pp. 240–248, 2020.
- [22] N. Zhang, S. Guo, Y. Dong et al., "Joint task offloading and data caching in mobile edge computing networks," *Computer Networks*, vol. 182, pp. 1–10, 2020.
- [23] I. Alghamdi, C. Anagnostopoulos, and D. P. Pazaros, "Data quality-aware task offloading in mobile edge computing: an optimal stopping theory approach," *Future Generation Computer Systems*, vol. 117, pp. 462–479, 2021.
- [24] B. Mao, F. Tang, Z. M. Fadlullah et al., "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.
- [25] V. Petrov, M. A. Lema, M. Gapeyenko et al., "Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 485–501, 2018.
- [26] G. Xu, B. Dai, B. Huang, J. Yang, and S. Wen, "Bandwidth-aware energy efficient flow scheduling with SDN in data center networks," *Future Generation Computer Systems*, vol. 68, pp. 163–174, 2017.
- [27] J. Zhang, M. Ye, Z. Guo, C.-Y. Yen, and H. J. Chao, "CFR-RL: traffic engineering with reinforcement learning in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2249–2259, 2020.
- [28] X. Huang, T. Yuan, G. Qiao, and Y. Ren, "Deep reinforcement learning for multimedia traffic control in software defined networking," *IEEE Network*, vol. 32, no. 6, pp. 35–41, 2018.
- [29] Y. Wei, X. Ma, N. Yang et al., "Energy-saving traffic scheduling in hybrid software defined wireless rechargeable sensor networks," *Sensors*, vol. 17, pp. 2126–2132, 2017.
- [30] A. Bahnasse, F. E. Louhab, H. Ait Oulahyane, M. Talea, and A. Bakali, "Novel SDN architecture for smart MPLS traffic engineering-DiffServ aware management," *Future Generation Computer Systems*, vol. 87, pp. 115–126, 2018.
- [31] J. Zheng, G. Chen, S. Schmid, H. Dai, J. Wu, and Q. Ni, "Scheduling congestion- and loop-free network update in timed SDNs," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2542–2552, 2017.
- [32] G. Li, S. Guo, Y. Yang et al., "Traffic load minimization in software defined wireless sensor networks," *IEEE Internet of Things Journal*, vol. 14, pp. 1–9, 2018.
- [33] J. Xiao, S. Chen, and M. Sui, "The strategy of path determination and traffic scheduling in private campus networks based on SDN," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 430–439, 2019.
- [34] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, in press.
- [35] W. Shi, Q. Li, C. Wang et al., *LEAP: Learning-Based Smart Edge with Caching and Prefetching for Adaptive Video Streaming* IEEE/ACM IWQoS, Phoenix, AZ, USA, 2019.

- [36] X. Molinero and F. Riquelme, "Influence decision models: from cooperative game theory to social network analysis," *Computer Science Review*, vol. 39, pp. 1–6, 2021.
- [37] E. Klotz and A. M. Newman, "Practical guidelines for solving difficult mixed integer linear programs," *Surveys in Operations Research and Management Science*, vol. 18, pp. 18–32, 2016.
- [38] L. Yuan and M. Wei, "Performance optimization mechanism of adolescent physical training based on reinforcement learning and markov model," *Mobile Information Systems*, vol. 2020, Article ID 8868225, 10 pages, 2020.

Research Article

Education Data-Driven Online Course Optimization Mechanism for College Student

Ziqiao Wang and Ningning Yu 

Yanbian University, No. 977 Gongyuan Road, Yanji 133002, Jilin Province, China

Correspondence should be addressed to Ningning Yu; nancy510306@ybu.edu.cn

Received 10 February 2021; Accepted 4 March 2021; Published 13 March 2021

Academic Editor: Jianhui Lv

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

During the recent epidemic period of COVID-19, online courses have become an important learning form for college students. However, online learning cannot communicate face to face in class and position students' abilities accurately, and there are many problems and limitations such as one-way evaluation, for example, neglecting of process evaluation and simple evaluation form. Therefore, how to construct the evaluation system of online course teaching and make effective use of the evaluation mechanism becomes an urgent problem. Based on the big data mining of online course evaluation data, the online course evaluation optimization architecture based on process evaluation is proposed. The optimization of online course evaluation is analyzed from online course evaluation data and student comments using deep learning and collaborative filtering technology. This includes improving teacher teaching and improving student learning efficiency. Data experiment proves that the proposed algorithm can provide an optimal evaluation strategy, guarantee the students' learning quality, and improve the efficiency of online course.

1. Introduction

The outbreak of COVID-19 has led to the shift of courses from live to online. Online courses are available in live broadcast and on-demand mode. Compared to traditional classes, the feedback of online courses is provided in the form of comments and posts. Although online courses solve the problem of students' listening, the learning quality is not satisfactory, and many problems occur. For example, online courses are difficult to understand, and the teaching effect is difficult to guarantee. Specifically, most colleges are vulnerable to formal influences when developing online courses, which may change the teaching content at will. As a result, the teaching content is not properly correlated, and a complete knowledge system cannot be formed. In addition, the teaching method is focused on the teacher-oriented one-way inculcation. The student's main role in the classroom is not fully reflected and the learning efficiency is low. Teachers also have difficulty getting accurate students' feedback. Furthermore, limited by teaching form, there is a lack of effective classroom discussion among students.

According to the above observation, one of the root causes of online course problems is that teachers cannot effectively obtain student's feedback during course development, which results in information asymmetry. Fortunately, in just over a year (during COVID-19), online courses have generated a lot of education big data. Education big data is the sum of data generated during the teaching and receiving of knowledge. According to Gartner report [1], big data is a collection of data that cannot be captured, managed, and processed by conventional software tools within a certain period of time; it is a massive, high-growth, and diversified information asset that requires new processing models to have stronger decision-making, insight, and process optimization capabilities. IBM proposes that big data has "5V" characteristics: Volume (large amount), Velocity (high speed), Variety, Value (low-value density), and Veracity. Compared with traditional big data, education big data has features of two-way feedback, changeability, extensiveness, and specificity.

Education big data can reshape the three major characteristics of learning: feedback, individualization, and

probabilistic predictions [2]. We believe that education big data mining can collect students' feedback data on curriculum resources, implement personalized online courses resources to meet students' individual needs, and optimize the learning content, time, and learning mode of online courses resources through probability prediction to incorporate empirical data into online courses resources. With the support of education big data, information asymmetry in online courses is broken.

Data mining refers to the process of searching for hidden information in a large amount of data by using a specific algorithm. Essentially, data mining is an information search technology based on data management architecture and searching algorithm. Aimed at optimizing online courses, we propose a Feedback Mining-based Online Course (FMOC) optimization architecture in the context of education big data. FMOC is a teaching mode supported by education big data. As depicted in Figure 1, it includes teaching, differentiated feedback, guidance, and closed-loop feedback for course evaluation optimization, enabling a virtuous cycle of teaching activities. This provides a theoretical basis for studying the optimization of educational and teaching curriculum resources at other levels. In addition, based on FMOC, we propose an optimization scheme of online course teaching evaluation system and systematically analyze and evaluate the effect of the scheme. Then, based on the evaluation, a personalized exercise recommendation method combined with Deep Knowledge Tracing is proposed to provide suitable test questions for students. Finally, we synthesize all the feedback to form a personalized curriculum opinion model to promote the teacher to improve the online courses.

The rest of the paper is organized as follows. Section 2 provides an analysis of related work. In Section 3, the evaluation and optimization scheme of online course learning based on formative evaluation is proposed. Section 4 presents a personalized exercise recommendation method based on knowledge tracing. Section 5 describes the individualized opinion model for teacher improvement. The effectiveness of the scheme is evaluated by simulation experiment in Section 6. Finally, conclusions are drawn in Section 7.

2. Related Work

Traditional course evaluation focuses on students' final academic achievement and ignores the learning process. The formative evaluation shifts the focus to the students' learning process, for example, the actions, effects, and willingness in the process. It can help students correct their learning habits, adjust learning methods and objectives, and detect their learning effects and finally achieves the goal of completing the course and improving students' ability. Reference [3] proposes the theory of formative evaluation and final evaluation and their difference. The theory has greatly influenced the education circle, especially in language and program language teaching. Reference [4] proposes that teaching evaluation can be divided into three categories: diagnostic evaluation, formative evaluation, and final

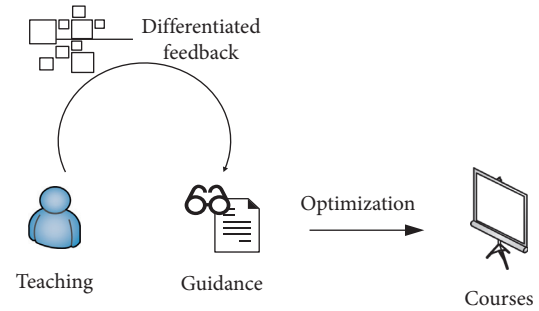


FIGURE 1: FMOC architecture.

evaluation. The three kinds of evaluations have different evaluation objectives and will produce different evaluation results. Diagnostic evaluation [5] is generally used for the stage when a course is just identified and emerging.S

The existing problem recommendation methods are mainly divided into the method based on collaborative filtering [6] and the method based on knowledge modeling [7]. The research works are as follows: [8] generates personalized recommendations and finds other users with similar interests by submitting comments on educational resources and visiting and searching for comments from others; [9] uses collaborative filtering and data mining technologies to analyze students' reading data and generate recommended scores to help students select appropriate courses. Reference [10] proposes the student personalized sorting algorithm EduRank, which uses collaborative filtering to find similar students and summarizes the rankings of similar students to construct the problem difficulty ranking of the target students. The algorithm can be used to assist teachers in customizing exercise sets and exams for students. Reference [11] proposes a recommendation module Protus of a programming tutoring system, which uses the Apriori algorithm to mine frequent sequences to analyze the habits and interests of similar students and makes personalized recommendations based on the ratings of these frequent sequences, which can automatically adapt to students' interests and knowledge level. Reference [12] proposes a learning material recommendation method based on fuzzy tree matching.

Corbett et al. propose Bayesian Knowledge Tracing (BKT) [13]. BKT models students' underlying knowledge state into a set of binary variables, each of which represents the understanding or lack of understanding of a single concept, using the implicit Markov model, and iteratively updates guessing rate, learning rate, error rate, and learning initial probability of each knowledge point to obtain students' knowledge point. Reference [14] introduces students' learning speed parameters into BKT, which increases the accuracy of BKT. It is also used in the intelligent tutoring system to guide students in learning. Deep Knowledge Tracing (DKT) [15] was proposed to train RNN to model students' knowledge status. DKT achieves better effects than BKT without manual labeling of the training dataset. In recent years, many scholars have applied and improved DKT: [16] proposes more features for learning and adding automatic encoder layer to reduce the input dimension and

improve the model performance. Reference [17] uses bidirectional LSTM to train the text representation information of the question and introduces the attention mechanism in the process of knowledge state modeling to improve the accuracy of knowledge tracing by combining the two models. Reference [16] reconstructs the loss function of the DKT model and adds three regular term parameters to improve the stability and accuracy of the mode. Reference [18] encodes students' programming submission records into the DKT, trains the model, and predicts students' performance in subsequent programming tests.

3. Online Course Evaluation Scheme Based on Process Evaluation

Process evaluation (PE) emphasizes timely information feedback to help teachers revise and improve their curriculum, thus improving teaching and learning effectiveness. Thus, the tasks involved in the process evaluation include the following:

- (1) Tracking the learning process: To track the learning process, the teacher is required to assign different learning tasks at each learning stage, clearly describe learning objectives, and design evaluation methods for these objectives. To achieve their learning goals, students will adjust their learning methods based on the evaluation results.
- (2) Testing the learning effect: The formative evaluation should give students a clear understanding of whether they have completed the current stage and what they are going to do for the next stage. This helps to enhance students' interest in learning, enhance students' confidence in learning, and improve their learning methods.
- (3) Identifying problems: Based on feedback and analysis of previous comments, both teachers and students can find out the causes of problems through formative evaluation and seek the best solution to the problem.
- (4) Correcting student learning strategies: Teachers can contact students in a timely manner and work with students to build and improve learning strategies and methods based on problems in the learning process. This process will greatly improve students' potential abilities.

Based on the above steps, the process of evaluating and optimizing the online course learning process is shown in Figure 2. The proposed PE can track the whole process of online learning and record and check all learning processes.

In order to better use data mining technology, weighting coefficients are considered, the classification of variable continuation (Figure 2) is refined, and the evaluation optimization method is modeled from the following aspects.

AHP [19] is a system analysis method proposed by T. L. Satty. The basic principles of AHP are as follows: a set of variables that contain specific factors related to a complex problem. It is modeled based on a composite structure

model, and factors at each level are compared and ranked by importance. Finally, the weight of each influence factor in the decision-making problem is determined.

Firstly, set a decision matrix, sort each comparison row, and generate a decision matrix $A = (a_{ij})_{n \times n}$, where n is the total number of associated pairs of comments. The evaluation weights are then confirmed by AHP. Solve the eigenvalue for the rank of the decision matrix A to obtain the maximum eigenvalue R_{\max} and the corresponding eigenvector W . The weight calculation results are sorted at different levels based on the sequence of feature values. Finally, the consistency check of the decision matrix needs to calculate the consistency index $\text{Ind} = (R_{\max} - n)/(n - 1)$. The random consistency ratio meets the following requirements: $\text{Ratio} = (\text{Ind}/I) < 0.1$. It is considered that the decision matrix meets the requirement of consistency. Otherwise, the decision matrix needs to be adjusted to meet the consistency requirement.

Assume that the evaluation condition set is as follows:

$$C = [c_1, c_2, \dots, c_5], \quad (1)$$

where C represents 5 score levels. The corresponding score is expressed in vectors. The corresponding evaluation standard may be expressed as

$$E = [e_1, e_2, \dots, e_5]. \quad (2)$$

Determine the fuzzy matrix of X_{ij} . Calculate the $r_{ij}^{(t)}$ value of X_{ij} . $r_{ij}^{(t)}$ belongs to the dimension value marked t . The member function $r_{ij}^{(t)}$ is as follows:

$$r_{ij}^{(t)} = \frac{\max r_{ij}^{(t)} - r_{ij}^{(t)}}{\max r_{ij}^{(t)} - \min r_{ij}^{(t)}}. \quad (3)$$

Define the fuzzy evaluation vector of X_{ij} as K_i :

$$K_i = \omega_i R_i(k_1, k_2, \dots, k_5). \quad (4)$$

The fuzzy compliance evaluation vector of X is expressed as Q and may be obtained by using the following formula:

$$Q = WK = (q_1, q_2, \dots, q_m), \quad (5)$$

where

$$q_t = \min 1, \sum_{j=1}^k b_j p_{jt}, \quad (6)$$

where $t = 1, 2, \dots, m$. Establish a fuzzy evaluation model and the evaluation value can be calculated using the following formula:

$$V = EQ^T. \quad (7)$$

4. Personalized Exercise Recommendation Method Based on Knowledge Tracing

Aiming at the disadvantages of using collaborative filtering method and knowledge modeling method to generate personalized exercise recommendation, a personalized

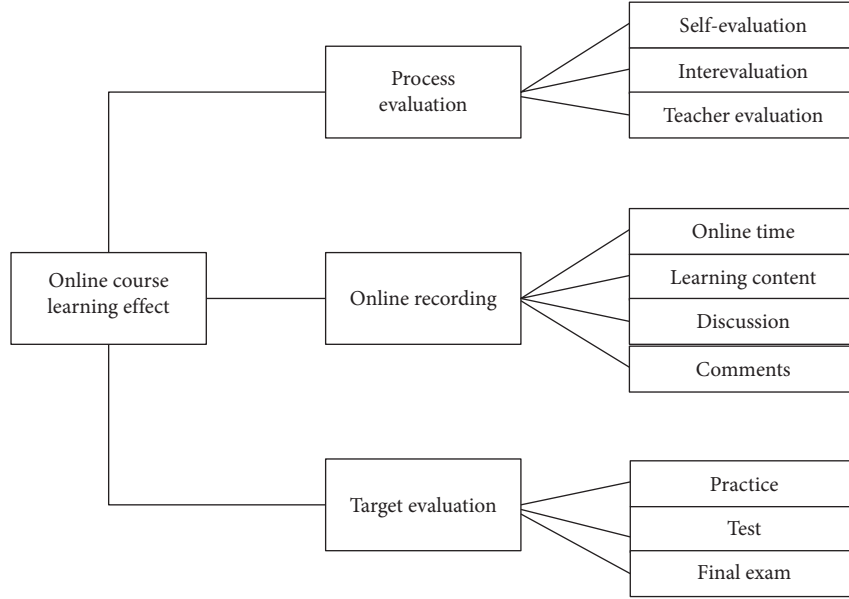


FIGURE 2: Online course evaluation scheme based on process evaluation.

exercise recommendation method combining Deep Knowledge Tracing (KT) model and collaborative filtering method is proposed. KT uses Deep Knowledge Tracing model to model students' learning status (i.e., learned knowledge) and then combines similar student information to perform collaborative filtering recommendation. The recommendation result considers both students' learning status and the generality of group students' learning status, thus improving the explanatory and accurate results.

4.1. Knowledge Tracing. Assume a time sequence $S = X_1, X_2, \dots, X_m$ of m students doing exercises where a contact sequence $X_i = x_1, x_2, \dots, x_t$ of each student is input into a Recurrent Neural Network (RNN). The input sequence is represented by monothermal encoding. For example, if the input data involves k exercises, each question is represented by 0 and 1, respectively, and the corresponding input length is $2k$. If the student answers the i -th question incorrectly, the value of the i -th position is 1 and those of the other positions are 0. Otherwise, the value of the i -th + 1 bit position is 1 and those of the other positions are 0. After data is encoded and input into the deep neural network model, the output sequence y_t of the model is established on the training of the exercise sequence of the students at the first t moment. The length of the sequence is n , and each bit position represents the correct answer probability of each corresponding question. By training the deep neural network model, the knowledge level vector of each student is calculated and the knowledge level vector matrix $U (m * n)$ is obtained.

4.2. Score Prediction. For the knowledge horizontal vector matrix U , the similarity between the vector U_i and U_j is determined by the cosine similarity. Assume that

$U_i = a_1, a_2, \dots, a_n$ and $U_j = b_1, b_2, \dots, b_n$, and the cosine similarity between U_i and U_j is calculated as follows:

$$\cos(\theta) = \frac{U_i * U_j}{\|U_i\| * \|U_j\|}. \quad (8)$$

A student whose cosine similarity is greater than a certain threshold (e.g., 0.9) is selected as a similar user by ranking the cosine similarity. For target student i , a final exercise score vector of the target student i is predicted according to the following formula:

$$f_i = \rho * U_i + (1 - \rho) * (N_i)^{\text{avg}}, \quad (9)$$

where U_i is the student's knowledge level vector, and $(N_i)^{\text{avg}}$ is the average of similar students' knowledge level vectors. The ratio of individual students' knowledge level to the common knowledge level among students is adjusted by the parameter ρ . The value range of ρ is $[0, 1]$. As the value of β increases, the impact on the student's personal knowledge level increases. Particularly, when $\rho = 0$, f_i is the average of the student's knowledge level excluding the student's personal status. When $\rho = 1$, f_i is the level vector of personal knowledge output by the DKT model [20].

4.3. Recommendation. In this step, a recommendation result is generated for students. When creating an exercise recommendation, determine an appropriate difficulty range $[\alpha_1, \alpha_2]$ ($\alpha_2 > \alpha_1$) as the difficulty range of the problem, and then score the problem vector f_i , and recommend the students with the correct probability of α_1 to α_2 . For example, setting α_1 to 0.6 and α_2 to 0.8 will recommend questions with a correct answer probability of 0.6 to 0.8 to the student based on the predicted problem score vector. The proposed KT method takes into account the individual knowledge level of students and combines the similar group

of students to the practice recommendation. The recommended practice can reflect the real knowledge level of students and recommend personalized exercises with certain difficulty to students. The Algorithm 1 is described as follows.

5. Personalized Opinion Model

Students' comments on the course can reflect users' preference for the online course to a certain extent. Therefore, this section uses the Text Classification (TC) model to score the comments of students and courses and uses the classic recommendation algorithm to recommend personalized course advice to teachers.

5.1. Scoring Model. The result of user comments on the course has an important impact on personalized recommendation. Course comments are short and have few valid information. The traditional text classification model does not have good effects [21]. In this paper, we use deep learning algorithm to score comments and use Word2-Vector [22] to represent short text information as low-dimensional vectors. Based on this, we use neural network model to train comments. Compared with traditional methods, this method can improve the accuracy and recall rate of scoring models effectively. In this paper, the rating is $-1/1$. The value -1 indicates that you do not like the course (0 indicates that you do not like the course), and the value 1 indicates that someone likes the course. The definition of the problem is as follows: $R = r_1, r_2, \dots, r_n$ indicates the course review dataset. The feature set of each course review sample is represented by X , and the corresponding category label is represented by Y : positive comment and negative comment. P indicates the probability of positive and negative comments: $P \in [0, 1]$. F indicates the characteristic matrix of $n * m$, n indicates the total number of samples, m indicates the total number of features $|X|$, and y_i indicates the prediction result of the i -th sample.

$$y_i = \begin{cases} 1, & P_i > \Phi, \\ 0, & \text{other,} \end{cases} \quad (10)$$

where Φ indicates the threshold of the classification model. Usually, the value was 0.45 during the experiment.

On the basis of the problem definition, we design a specific neural network model to score the course review. Figure 3 shows the deep learning model of course review scoring.

The neural network consists of five modules. The input layer, convolution layer, pooling layer, fully connected layer, and output layer are in sequence. The input layer is used to cut the comment text and uses one-hot to represent the text. Convolution layer converts the one-hot feature of the input layer into a low-dimensional vector representation by using an embedding method and extracts text features by using different convolutions [23]. The pooling layer normalizes data at the convolution layer and converts convolution kernels of different dimensions into the same dimension. Fully connected layer learns linear relationships in the

output of the pooling layer and learns more text features. The output layer is mainly used to predict samples. First, the ReLU function is used to perform nonlinear transformation on the linear relationship of the fully connected layer, and then the Softmax function is used to perform classification and scoring. The network structure can effectively learn the potential semantic space of short text and thus learn the student's scoring prediction of the course.

5.2. Advice Recommendation Model. Based on the course score, the preference matrix of users is constructed by collecting the relationships among students, courses, and comments. Based on the matrix, the personalized learning model of online courses is trained by using the recommendation algorithm. The collaborative filtering recommendation algorithm is used for model training and prediction. A specific algorithm process is shown in Figure 4.

In the collaborative filtering recommendation algorithm, a preference matrix of $m * n$ is used to indicate the preference of a user for a course. A higher score indicates that the user likes the course. The value 0 indicates that the course is not selected. In Figure 4, the row of the preference matrix indicates support, the column indicates a course, and U_{ij} indicates a preference for course j using i . The collaborative filtering algorithm consists of two processes: prediction and recommendation. The prediction process is to predict the possible scores of users who have not selected courses. The recommendation process is based on the results of the prediction phase and recommends top N questions that students may have advice about.

6. Performance Evaluation

6.1. Settings. We leverage Keras [24] as basic neural model and use Python 3.7 library as coding tools. The FMOC algorithm is run on a workstation equipped with two GeForce GTX 1080Ti GPUs.

In order to verify the implementation effect of the proposed FMOC model in the online course learning process evaluation, this paper extracts and analyzes the background data of several major online course websites. In order to enhance the reliability of the verification results, the results of the formative evaluation optimization matching course were compared. The test data selected in this paper are extracted from website script data. Incomplete records are removed, and data information is classified by keyword index.

6.2. Results. On the whole, from the aspect of course matching optimization, this paper performs a comparison with the expert manual recommendation classification and verifies the reliability of the matching theory from another dimension. This test compares the automatic matching results of the proposed solution, the current course matching status on the website, and the classification results of experts' manual suggestions, as shown in Figure 5.

According to the analysis in Figure 5, the matching degree of the proposed solution and expert suggestions is

```

Input: S
Output: Recommended list
Begin
01: s = onehot(S); /* onehot encoding */
02: U (m * n) = trainDKT(s);
03: for Ui ∈ U;
04: for Ij ∈ U;
05: Simi,j = cos(Ui, Uj);
06: Ni = rank(Simi);
07: fi = ρ * Ui + (1 - ρ) * (Ni)αvg;
08: Ri = recom(fi, K, β1, β2);
09: Return R;
End

```

ALGORITHM 1: KT-based recommendation.

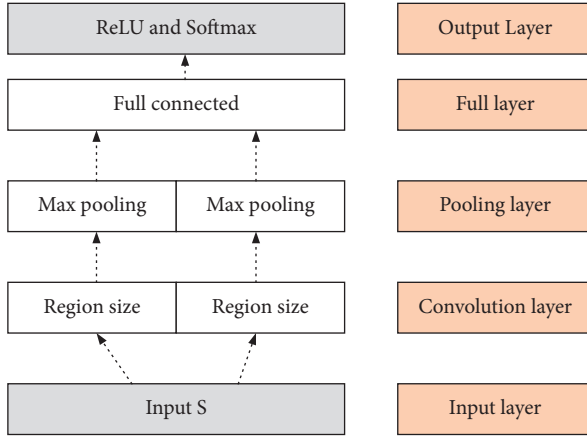


FIGURE 3: Scoring model.

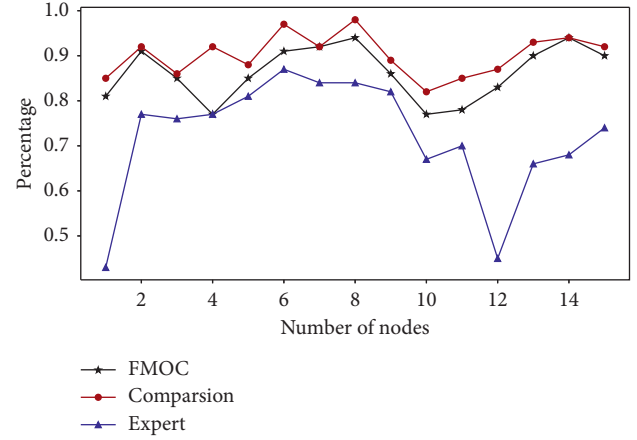


FIGURE 5: Analysis of course matching optimization.

better than the result of no matching optimization on the

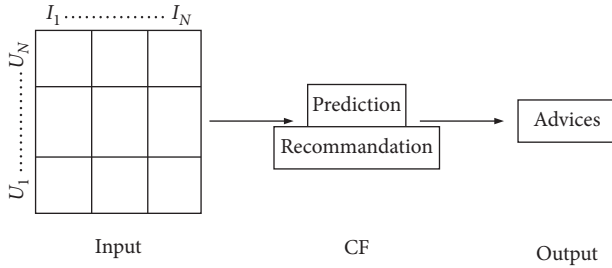


FIGURE 4: Advice recommendation.

website. It can also be seen that the judgment results of some courses in the proposed scheme are lower than those of experts, mainly because the lack of complete information guidance in the matching considerations leads to some deviations.

To evaluate the effect of personalized exercise recommendation, two sets of datasets are used. The first set is the ASSISTments [25] public dataset. ASSISTments is an open online education platform that can simultaneously teach and

evaluate students' learning; the dataset contains more than 500,000 records of submitted math questions. The second dataset is the history records of student submissions crawled from the online judge of a university. Table 1 lists the information about the two datasets.

In order to evaluate the effectiveness of the DKT-CF [20] method in generating personalized exercise recommendations for students, this paper firstly uses the indexes of precision, recall, and f1 to evaluate the recommendation effect; the formula for calculating the evaluation indicators is as follows:

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

$$\text{recall} = \frac{TP}{TP + FN}, \quad (11)$$

$$f1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}.$$

In the DKT model's training phase, 200 LSTM [26] nodes are used at the hidden layer, keep_prob is set to 0.5, learning_rate is set to 0.01, max_epoch is set to 200, and set batch_size is set to 32. The experimental results are shown in Tables 2 and 3. The following information can be obtained

TABLE 1: Dataset.

| Dataset | Number of students | Exercises | Number of crimest |
|-------------|--------------------|-----------|-------------------|
| ASSISTments | 4200 | 202 hours | 645156 |
| OJ | 2698 | 2540 | 455287 |

TABLE 2: Results of ASSISTments.

| | DKT | DKT+ | UCF | KCF | DKT-CF |
|-----------|-------|-------|-------|-------|--------|
| Precision | 0.967 | 0.958 | 0.901 | 0.945 | 0.988 |
| Recall | 0.354 | 0.308 | 0.258 | 0.235 | 0.297 |
| f1 | 0.446 | 0.424 | 0.421 | 0.374 | 0.356 |

TABLE 3: Results of OJ.

| | DKT | DKT+ | UCF | KCF | DKT-CF |
|-----------|-------|-------|-------|-------|--------|
| Precision | 0.851 | 0.868 | 0.789 | 0.864 | 0.824 |
| Recall | 0.333 | 0.301 | 0.276 | 0.304 | 0.311 |
| f1 | 0.456 | 0.478 | 0.400 | 0.457 | 0.468 |

TABLE 4: Results of BaseDataset.

| Method | SVM | | FMOC-TC | |
|----------|-----------|--------|-----------|--------|
| Dataset1 | Precision | Recall | Precision | Recall |
| Dataset1 | 0.61 | 0.58 | 0.81 | 0.63 |

from the results: (1) In the experiments on the two datasets, the proposed DKT-CF algorithm is better than other algorithms. The accuracy, recall rate, and f1 value are all the highest. This indicates that the DKT-CF method can produce more accurate recommendation results compared to other methods and is more effective than other methods. (2) In the experiment, the accuracy of the UserCF algorithm using the traditional collaborative filtering algorithm is lower. After the students' knowledge status is integrated, better results (KS-CF method) were obtained. The accuracy increased by 7.5% and 6.7%, respectively, and the method of introducing knowledge state information to improve the accuracy of traditional collaborative filtering is effective. (3) DKT-CF is a combination of DKT and KS-CF, which achieves the best effect; the method of considering both students' personality and students' study group in exercise recommendation can give the most effective personalized exercise recommendation.

Finally, to evaluate the effect of the personalized opinion model, an effect obtained by using the traditional word feature + SVM and deep learning improved algorithms in the training dataset is shown in Table 4.

As shown in the preceding table, the classification effect of the traditional word feature + SVM algorithm is slightly worse than that of the deep learning method proposed in this paper. We enrich the semantic information of short text by using Word2Vec word vector expression and then use the deep network structure to learn the semantic information. This function greatly improves the accuracy and recall rate of

the scoring model, effectively identifies improvement suggestions, and provides basis for teachers to optimize courses.

7. Conclusions

Based on the analysis of three kinds of evaluation characteristics, this paper researches the formation evaluation into problem modeling. Through collecting the information of each link in the teaching process, comprehensive analysis, and evaluating the evaluation of online course teaching, we provide an optimization scheme for students' online learning. Aiming at the fact that traditional collaborative filtering recommendation method does not consider students' knowledge mastering level and the recommendation method based on knowledge modeling ignores the disadvantages of group learning among students, this paper proposes a personalized exercise recommendation method based on Deep Knowledge Tracing DKT-CF. Finally, the experiment proves the validity and rationality of the proposed DKT-CF method, which can improve the learning experience and efficiency of users.

In future research, we can improve the Deep Knowledge Tracing model and add more personalized features for learning. Based on the recommendation method, you can use other recommendation algorithms to generate recommendations. Based on recommendation scenarios, recommendation policies in different scenarios are also worth discussing.

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 there are no conflicts of interest.

Acknowledgments

This work was supported by the Approved Project of Jilin Undergraduate Higher Education and Teaching Reform 2020 (General Project).

References

- [1] M. A. Beyer and D. Laney, *The Importance of Big Data: A Definition*, Gartner Publications, Stamford, CT, USA, 2012.
- [2] B. Williamson, *Big Data in Education: The Digital Future of Learning, Policy and Practice*, Sage, Thousand Oaks, CL, USA, 2017.
- [3] T. Scriven, *The Methodology of Evaluation*, American Educational Research, Washington DC, USA, 1967.
- [4] B. S. Bloom, *Evaluation to Improve Learning*, McGraw-Hill, New York, NY, USA, 1977.
- [5] D. H. Gutmann, A. Aylsworth, J. C. Carey et al., "The diagnostic evaluation and multidisciplinary management of neurofibromatosis 1 and neurofibromatosis 2," *JAMA: The Journal of the American Medical Association*, vol. 278, no. 1, pp. 51-57, 1997.

- [6] X. He, L. Liao, H. Zhang et al., "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, Perth Australia, April 2017.
- [7] A. J. Canas, G. Hill, R. Carff et al., "CmapTools: a knowledge modeling and sharing environment," 2004.
- [8] A. Walker, M. Recker, K. Lawless et al., "Collaborative information filtering: a review and an educational application," *International Journal of Artificial Intelligence in Education*, vol. 14, no. 1, pp. 3–28, 2004.
- [9] H. Mei-hua, "A personalized English learning recommender system for ESL students," *Expert Systems with Applications*, vol. 34, no. 1, pp. 683–688, 2008.
- [10] A. Segal, Z. Katzir, B. Shapira et al., "Edurank: a collaborative filtering approach to personalization in e-learning," in *Proceedings of the 7th International Conference on Educational Data Mining*, pp. 68–75, London, UK., July 2014.
- [11] A. K. Milicevic, B. Vesin, M. Ivanovic et al., "E-learning personalization based on hybrid recommendation strategy and learning style identification," *Computers and Education*, vol. 56, no. 3, pp. 885–899, 2011.
- [12] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2412–2426, 2015.
- [13] A. T. Corbett and J. R. Anderson, "Knowledge tracing: modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.
- [14] M. V. Yudelson, K. R. Koedinger, and J. Gordon, "Individualized Bayesian knowledge tracing models," *International Conference on Artificial Intelligence in Education*, vol. 42, pp. 172–181, 2013.
- [15] P. Chris, B. Jonathan, H. Jonathan et al., "Deep knowledge tracing," in *Proceedings of the NIPS'15 28th International Conference on Neural Information Processing Systems*, pp. 505–513, Vancouver, Canada, April 2015.
- [16] C. K. Yeung and D. Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proceedings of the ACM Conference on Learning at Scale*, pp. 41–50, London, UK, June 2018.
- [17] L. Zhang, L. Xiong, S. Zhao et al., "Incorporating rich features into deep knowledge tracing," in *Proceedings of the ACM Conference on Learning at Scale*, pp. 169–172, Cambridge, MA., USA, April 2017.
- [18] L. Wang, S. Angela, L. Liu Larry et al., "Deep knowledge tracing on programming exercises," in *Proceedings of the ACM Conference on Learning at Scale*, pp. 201–204, Cambridge, MA, USA, April 2017.
- [19] Y.-M. Wang, Y. Luo, and Z. Hua, "On the extent analysis method for fuzzy AHP and its applications," *European Journal of Operational Research*, vol. 186, no. 2, pp. 735–747, 2008.
- [20] L. Zhang, X. Xiong, S. Zhao et al., "Incorporating rich features into deep knowledge tracing," in *Proceedings of the fourth ACM conference on learning*, pp. 169–172, Cambridge, MA, USA, April 2017.
- [21] R. Y. Toledo and Y. C. Mota, "An e-learning collaborative filtering approach to suggest problems to solve in programming online judges," *International Journal of Distance Education Technologies*, vol. 12, no. 2, pp. 51–65, 2014.
- [22] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [23] P. Dwivedi and K. Bharadwaj, "Effective trust-aware e-learning recommender system based on learning styles and knowledge levels," *Journal of Educational Technology and Society*, vol. 16, no. 4, pp. 201–216, 2013.
- [24] Keras, <https://keras.io/>, accessed on Feb. 2021.
- [25] Assisments, <https://new.assisments.org/>, accessed on Feb. 2021.
- [26] K. Greff, R. K. Srivastava, J. Koutnk et al., "L S T M: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

Research Article

Pattern Recognition and Neural Network-Driven Roller Track Analysis via 5G Network

Yuliang Guo 

Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Yuliang Guo; guoyl1975@dlmu.edu.cn

Received 19 November 2020; Revised 1 December 2020; Accepted 16 December 2020; Published 30 December 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Yuliang Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Roller skating is an important and international physical exercise, which has beautiful body movements to be watched. However, the falling of roller athletes also happens frequently. Upon the roller athletes' fall, it means that the whole competition is over and even the roller athletes are perhaps injured. In order to stave off the tragedy, the roller track can be analyzed and be notified the roller athlete to terminate the competition. With such consideration, this paper analyzes the roller track by using two advanced technologies, i.e., pattern recognition and neural network, in which each roller athlete is equipped with an automatic movement identifier (AMI). Meanwhile, AMI is connected with the remote video monitor referee via the transmission of 5G network. In terms of AMI, its function is realized by pattern recognition, including data collection module, data processing module, and data storage module. Among them, the data storage module considers the data classification based on roller track. In addition, the neural network is used to train the roller tracks stored at AMI and give the further analysis results for the remote video monitor referee. Based on NS3, the devised AMI is simulated and the experimental results reveal that the prediction accuracy can reach 100% and the analyzed results can be used for the falling prevention timely.

1. Introduction

With respect to the beautiful sports, the roller skating is a nonnegligible and international physical exercise thanks to its graceful body movements. Since the roller athletes make the highly ornamental postures during the process of competition, it perhaps causes the dangerous situations, where the falling of roller athletes is regarded as the most representative tragedy [1, 2]. Upon the roller athletes' fails, this indicates that the whole roller competition is forced to be over; at the same time, the roller athletes may be seriously affected. Therefore, it is a very important research topic to make the falling detection for the roller athletes. Regarding this, a comment and popular method is to analyze the roller track, and based on this, the falling risk can be decreased.

The neural network [3–5] is usually used to analyze the behavior track including but not limited to the roller track, this is because it has some obvious advantages. For example, it has the strong computation ability, especially for the large-scale data features. Besides, it also has the strong stability to

obtain the convergent analysis results. To sum up, the neural network can be used to train the prestored data related to the previous roller tracks and give the real-time training results to be used for the falling prevention. Well, there are a number of neural network models, but the back-propagation (BP) neural network [6, 7] has the simplest network structure. Thus, this paper uses BP neural network to do the data training.

In spite of this, it also exists three issues: at first, where the BP neural network module is embedded; then, what the prestored data container is; finally, how to transmit the analysis result between roller athletes and the remote video monitor referee. These first two issues require this paper to explore an additional component, i.e., automatic movement identifier, (AMI) to store such roller tracks data. In particular, AMI is equipped into the body of roller athlete. In other words, the BP neural network module is embedded into AMI and used to train the roller tracks stored at AMI. However, the further problem is how to obtain the roller tracks data. Regarding this, the pattern recognition [8–10] is

a highly-regarded candidate. That is to say, the function of AMI can be realized by pattern recognition by three inherent modules, i.e., data collection, data processing, and data storage. Furthermore, in terms of the last issue, it has the ultra-high requirement on the delay and stability of data transmission. Upon the transmission delay being out of the certain level of acceptance, the falling behavior cannot be avoided even though the system has detected the analysis results on the falling of roller athletes. Similarly, if the data transmission is unstable, the remote video monitor referee and roller athletes cannot make the following decision well. With these concerns, the 5G-enabled network can be deployed at the competition gymnasiums and stadiums so as to enhance the transmission stability and cut down the transmission delay.

By reviewing the above statements, it is observed that pattern recognition and BP neural network are used to analyze the roller track, and the 5G network is used to guarantee the data transmission [11–13]. To sum up, the contributions of this paper are shown as follows. At first, the whole roller track analysis structure is introduced. Then, AMI is devised and its function is realized by pattern recognition. Thirdly, the improved BP neural network is used to do the data training. Finally, the experimental results reveal that the prediction accuracy can reach 100% and the analyzed results can be used for the falling prevention timely.

The remaining paper is organized as follows. Section 2 introduces the whole network architecture. Section 3 uses pattern recognition to realize AMI. Section 4 uses BP neural network to train the data. Section 5 reports the experimental results. The overview of this paper is concluded in Section 6.

2. Network Architecture

The whole network architecture includes two character roles, i.e., roller athlete and remote video monitor referee, where the former is the main research object and the latter is the assistant research optimization, as shown in Figure 1. In particular, the whole network architecture introduces a new component, i.e., AMI to store the roller tracks data, and AMI is equipped into the body of roller athlete.

Furthermore, AMI consists of two modules, i.e., pattern recognition and BP neural network. Meanwhile, the pattern recognition module is used to realize AMI based on data collection, data processing, and data storage, where the data storage part considers the data classification based on roller track. The BP neural network module is used to train the roller tracks stored at AMI and give the further analysis results.

Moreover, based on the analyzed results, the whole network architecture involves two kinds of message transmissions. On one hand, AMI delivers the analyzed results to the roller athlete directly and the competition is forced to be terminated. On the other hand, AMI delivers the analyzed results to the remote video monitor referee for conducting the posture adjustment. If the analyzed results belong to the emergency situation, the first kind of message is performed; otherwise, the second kind of message is performed. Especially for the second condition, the data transmission

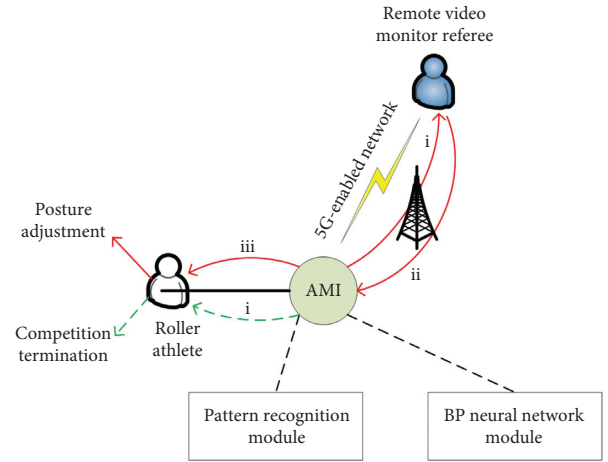


FIGURE 1: The whole network architecture.

between AMI and remote video monitor referee depends on the 5G-enabled network to guarantee the fast and stable performance.

3. Pattern Recognition

The pattern recognition is used to realize the function of AMI, which includes data collection, data processing, and data storage. The system framework of pattern recognition is shown in Figure 2, and the whole process is completed based on field-programmable gate array (FPGA) [14]. To be specific, the inherent static random access memory (SRAM) [15] is used to extract the roller tracks. Then, the digital signals are converted into the analog signals by a D/A transverter. Finally, the analog signals are amplified based on retractor and input into FPGA for receiving, processing, and storing.

3.1. Feature Vector Determination. During the process of data classification for roller tracks, it involves the attribution problem in terms of data samples. For such problem, the basic principle is to build the maximum membership degree function. Furthermore, let W_x denote the x -th data object's feature vector, and for N data applications in terms of roller tracks, W_x is expressed as $\{w_1, w_2, \dots, w_N\}$ and $W_x \in f(g)$. For any $g_0 \in G$, if it exists, $i \leq N$, and g_0 is regarded as one membership degree value of W_x . On this basis, all membership degree functions can be determined based on the feature values of roller tracks, which are used to make the data classification of roller tracks.

Regarding the determination of feature vector, the law of large numbers based on the Chebyshev [16] is used to obtain the coagulation equation with respect to the random variable:

$$\left[\frac{h - G[h]}{G[h]} < \delta \right] \geq 1 - \frac{\text{var}[h]}{\delta G^2[h]}, \quad (1)$$

where h is a random variable; $\text{var}[h]$ is the variance; $G[h]$ is the average value; and δ is the error value. When the variance is relatively small, the corresponding error value is smaller

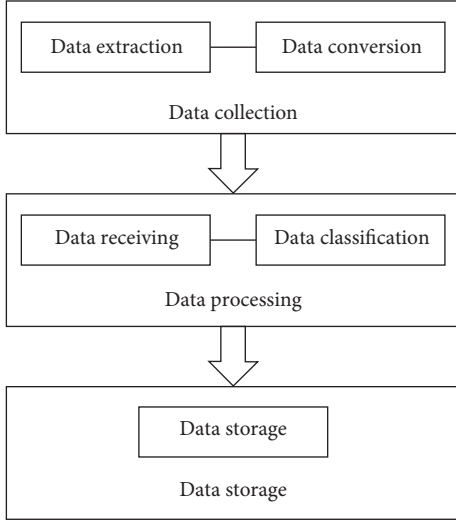


FIGURE 2: The whole system framework of pattern recognition.

than δ . In other words, when and only when the coagulation value is around the average value, the relatively small error value can be guaranteed. In order to further make the data classification for roller tracks, this paper introduces a resolution for objective set:

$$P = \frac{|G[h_0] - G[h_a]|}{\sqrt{\text{var}[h_0] + \text{var}[h_a]}}, \quad (2)$$

where h_0 is the object and h_a is the backdrop.

3.2. Membership Degree Function Computation. Let V_i denote any data application's feature vector, i.e., $V_i = \{v_1, v_2, \dots, v_j\}$, where v_j is the subcomponent of V_i and it concentrates around the average value. For a data application a , v_j is used to compute the related membership degree value e_a . In particular, the Cauchy distribution [17] is used as the reference of the membership degree function, as shown in Figure 3.

Based on the Cauchy distribution reference, e_a is computed as follows:

$$e_a = 1 - \frac{|v_j - \text{ave}|}{\sum_N v_j - \text{ave}}, \quad (3)$$

where ave is the average value. Regarding further obtaining of the method of data storage in terms of AMI, this paper uses the maximum membership degree to address this. Let D_a denote the maximum membership degree function regarding a , and it is defined as follows:

$$D_a = \Delta e_a, \quad a \in [1, j]. \quad (4)$$

According to D_a , this paper can complete the data storage in AMI by considering the data classification based on roller tracks, and the related pseudocode is described as follows. Among them, line 1 means the clocks collection; line 2 means the initialization of serial communication clocks; lines 4-5 and 7-8 mean the address processing; line 6 means

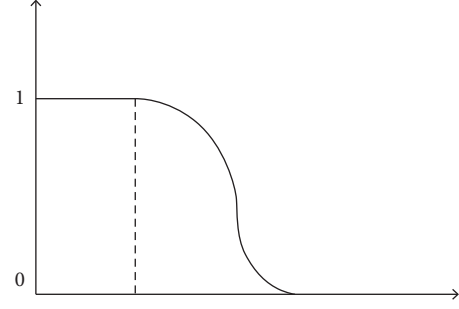


FIGURE 3: The Cauchy distribution reference.

to output the handled data; line 9 means the completion of data storage.

- (1) module flash
- (2) input w_clk
- (3) input r_clk
- (4) input w_runtrack
- (5) input r_runtrack
- (6) output [8:0]datain
- (7) input [9:0]wrad
- (8) input [9:0]rad
- (9) input [8:0]set_rtrack

4. BP Neural Network

BP neural network is a feedforward network with the multiple layers, and its topology is shown in Figure 4, including n input layers, p hidden layers, and q output layers.

For the BP neural network, the weight adjustment is considerably important, which is as follows:

$$\Delta w_{j,k} = -\eta \frac{\partial E}{\partial w_{j,k}} = \eta \delta_k y_j, \quad (5)$$

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} = \eta \delta_j x_i,$$

where $w_{j,k}$ is the network weight between the output layer and hidden layer; $w_{i,j}$ is the network weight between the input layer and hidden layer; $\Delta w_{j,k}$ and $\Delta w_{i,j}$ are two increments with respect to the weight; x_i is the input data from the i -th node; y_j is the output data from the j -th node; and $\eta \in (0, 1)$ is the learning rate of BP neural network. In particular, the negative sign means the gradient descent action. Furthermore, E is the error function of BP neural network, and it is defined as follows:

$$E = \frac{1}{2} \sum_{o=1}^q (d_o(n) - y_o(n))^2, \quad (6)$$

where $d_o(n)$ is the expected output result and $y_o(n)$ is the actual output result.

However, the traditional BP neural network has two obvious limitations. On one hand, the training speed is very slow; on the other hand, it is very difficult to converge to the

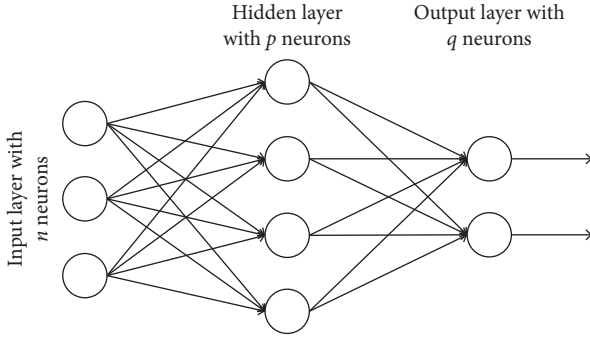


FIGURE 4: BP neural network topology.

global optimum. Therefore, in order to enhance the recognition efficiency and the accurate rate of roller track analysis, this paper improves the traditional BP neural network from two aspects, i.e., momentum attachment and learning rate optimization.

By using the momentum attachment method, the improved weight adjustment is as follows:

$$\Delta w(I+1) = \beta(w(I+1) - w(I)) + \alpha(1 - \beta) \frac{\partial E(I)}{\partial w(I)}, \quad (7)$$

where β is the momentum factor; α is the self-learning rate; I is the number of training; and $(\partial E(I)/\partial w(I))$ is the gradient of $w(I)$. Equation (7) indicates that, when the modeling error decreases, the different momentum factors are used to adjust the whole network parameters so as to reach the global optimum in terms of the network performance. To be specific, when the surface of error is the flat state, the momentum factor is increased to accelerate the convergence process; on the contrary, when the surface of error is the steep state, the momentum factor is decreased to avoid the network instability. According to the abovementioned statements, the momentum factor is defined as follows:

$$\beta = e^{-R - \|\partial E(I)/\partial w(I)\|}, \quad (8)$$

where R is a random number, satisfying $R \in (0, 1)$. In particular, when the surface of error is the flat state and the corresponding curve stays the bottom, $(\partial E(I)/\partial w(I))$ gradually decreases until the following equation is satisfied:

$$\Delta w(I+1) = \beta(w(I+1) - w(I)) = \Delta w(I), \quad (9)$$

which avoids the situation of $\Delta w(I) = 0$.

In fact, the unchanging learning rate has the important influence on the BP neural network performance. In order to guarantee that the BP neural network always has the maximal training effect, the dynamic learning rate is defined as follows:

$$\eta(I) = \begin{cases} \alpha(I-1), & E(I) < E(I-1), \\ \alpha(I-1)e^{-\gamma}, & \text{otherwise,} \end{cases} \quad (10)$$

where γ is a parameter, satisfying $\gamma \in [0.0001, 0.001]$.

According to the abovementioned statements, the BP neural network is used to train the roller tracks stored at AMI, and the related pseudocode is described as follows:

- (1) Input the roller tracks data
- (2) Build BP neural network according to Figure 4
- (3) Add momentum factor β and self-learning rate α into BP neural network
- (4) Compute the error function according to equation (6)
- (5) Modify the weight adjustment
- (6) Check whether the number of training satisfies the given I
- (7) Output the analyzed results on roller tracks
- (8) Send the results to the roller athlete or the monitor referee according to the real emergency

5. Performance Evaluation

The proposed pattern recognition and neural network-driven roller track analysis method, called PRNN, is implemented by NS3 [18], which is a discrete event emulator. Since the open dataset on the roller tracks cannot be obtained, this paper simulates 1500 roller competitions and considers them as the data trace. In addition, this paper is the first paper to propose roller track analysis based on the emerging computer technologies, and there is no suitable comparison baseline. As a result, this paper considers the method without the improved BP neural network, called PRWI, as the baseline. Meanwhile, there are four 5G wireless base stations which are used to support the efficient data communication. The detailed deployment is shown in Figure 5.

5.1. Parameter Determination: α and β . For the improved BP neural network, α and β are two important parameters. In this paper, suppose that $\alpha, \beta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and different settings generate different network performances. In this section, the standard deviation is used to measure the network performance, and the smaller standard deviation means the better network performance. The experimental results on standard deviation with respect to different α and β settings are shown in Table 1 and Figure 6.

As can be seen from Table 1 and Figure 6, with the increasing of α , the change of standard deviation includes two phases. At first, the standard deviation gradually decreases. Then, when α reaches 0.7, the standard deviation will have the minimal value. After that, the standard deviation gradually increases. As a result, α is set as 0.7. Different from α , the standard deviation from β shows four change phases, and the corresponding inflection points are 0.4115 ($\beta = 0.3$), 0.5106 ($\beta = 0.6$), and 0.4953 ($\beta = 0.7$) respectively. It is obvious that when $\beta = 0.3$, the corresponding standard deviation has the minimal value. As a result, β is set as 0.3.

5.2. Prediction Ratio. Based on $\alpha = 0.7$ and $\beta = 0.3$, this paper performs the roller track analysis based on 1500 roller competitions, where the number of simulations is set as 15. The experimental results on prediction ratio with respect to

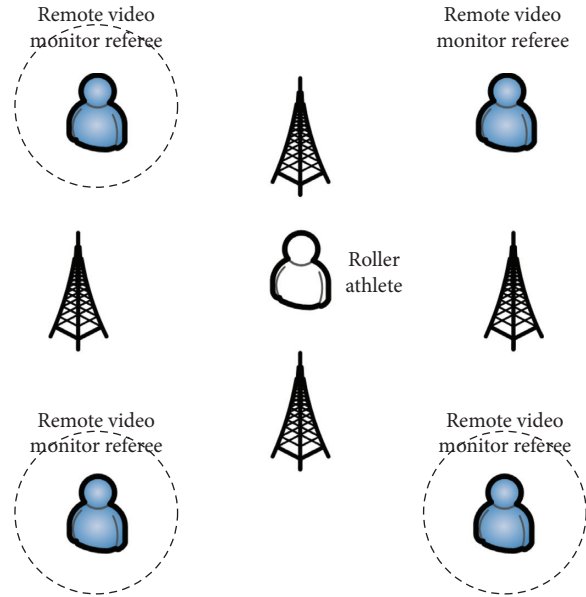


FIGURE 5: The detailed deployment on 5G-enabled network.

TABLE 1: The experimental results on standard deviation with respect to different α and β settings.

| α | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|--------------------|--------|--------|---------------|--------|--------|--------|---------------|--------|--------|
| Standard deviation | 0.6323 | 0.5964 | 0.5319 | 0.5216 | 0.4906 | 0.4534 | 0.3613 | 0.4059 | 0.4638 |
| β | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Standard deviation | 0.5361 | 0.4937 | 0.4115 | 0.4661 | 0.5032 | 0.5106 | 0.4953 | 0.5357 | 0.5661 |

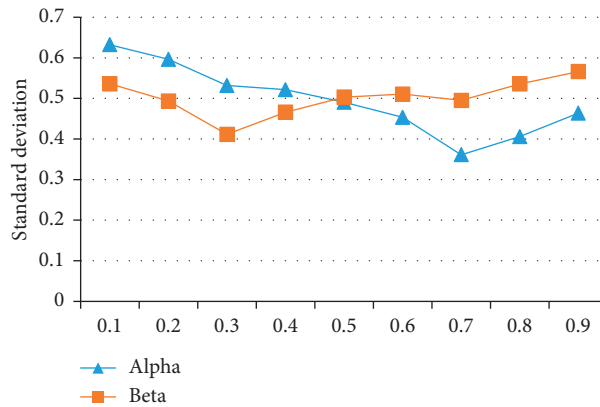


FIGURE 6: Different standard deviations with different α and β settings.

different simulations are shown in Figure 7. It is observed that PRNN always has better prediction ratio than PRWI, and the prediction ratio of PRNN always keeps 100%, because this paper enhances the traditional BP neural network from two aspects, i.e., momentum attachment and learning rate optimization. For PRWI, its prediction ratio is low and the corresponding stability is also bad, varying from 86.49% (the ninth simulation) to 95.06% (the second simulation).

5.3. *Communication Time.* Although PRNN has reached 100% prediction ratio, this does not mean that the analyzed results can be used for the falling prevention timely. Therefore, this section tests the communication time to

illustrate the availability of PRNN. For 1500 roller competitions, there are 60 times events requiring AMI to transmit the adjustment message or the termination message. Furthermore, there are 54 times events for the first condition while there are 6 times events for the second condition. The communication time between AMI to the roller athlete can be neglected due to the fact that AMI is equipped into the body of roller athlete. Given this, this paper only tests the communication time between AMI and the remote video monitor referee. The average communication times regarding 54 times events under different simulations are shown in Figures 8–11, where Figure 8 shows the experiment results based on 1 Gbps network bandwidth, Figure 9 shows

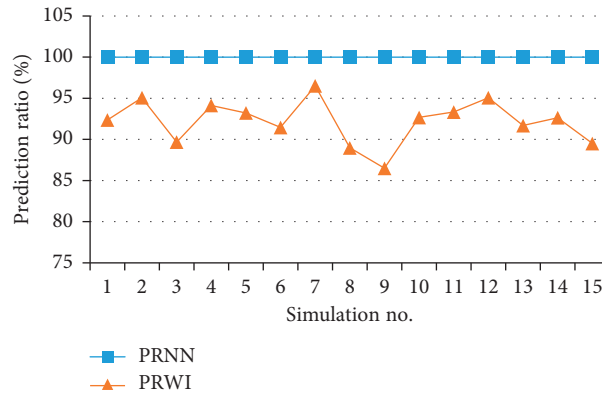


FIGURE 7: The experimental results on prediction ratio with respect to different simulations.

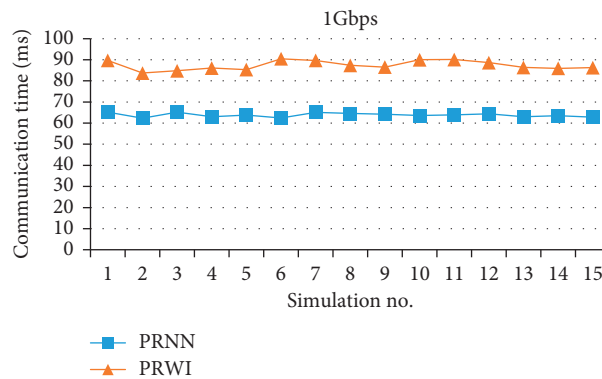


FIGURE 8: The experimental results on communication time with respect to different simulations, where network bandwidth is 1 Gbps.

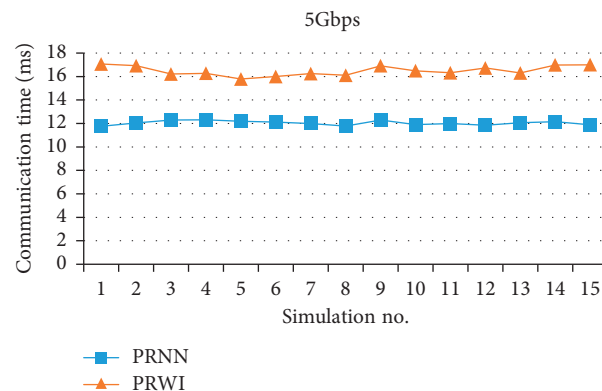


FIGURE 9: The experimental results on communication time with respect to different simulations, where network bandwidth is 5 Gbps.

those based on 5 Gbps network bandwidth, Figure 10 shows those based on 10 Gbps network bandwidth, and Figure 11 shows those based on 15 Gbps network bandwidth.

As can be seen from Figures 8–11, the communication time of PRNN is always smaller than that of PRWI, this is because the improved BP neural network accelerates the convergence speed. Furthermore, under the scenario of 5G-enabled network, with the increasing of network bandwidth,

the communication time decreases drastically. It suggests that 5G network has great performance improvement in terms of the communication time. In addition, it is also observed that, with the increasing of network bandwidth, the communication time difference between PRNN and PRWI becomes smaller and smaller. This suggests that 5G network can guarantee the availability of PRNN; that is to say, the analyzed results can be used for the falling prevention timely.

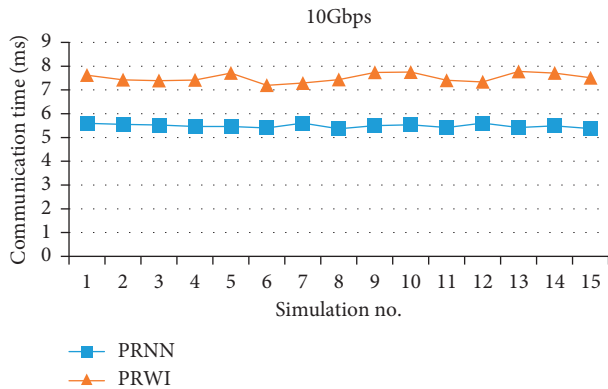


FIGURE 10: The experimental results on communication time with respect to different simulations, where network bandwidth is 10 Gbps.

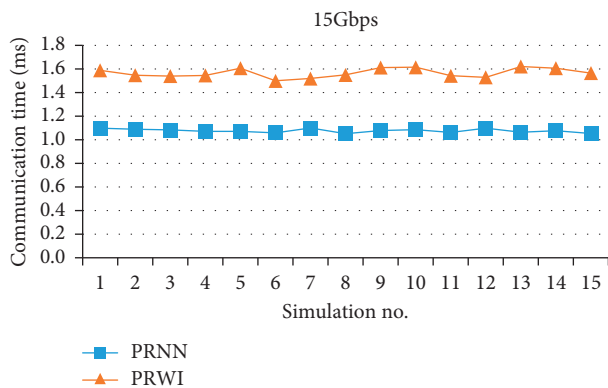


FIGURE 11: The experimental results on communication time with respect to different simulations, where network bandwidth is 15 Gbps.

6. Conclusions

This paper studies the roller track analysis based on pattern recognition and BP neural network. In particular, the 5G-enabled network is used to guarantee the stability and efficiency of transmission. Meanwhile, regarding pattern recognition, it is used to realize the function of AMI, which includes data collection, data processing, and data storage, and the whole process is completed based on FPGA. Then, regarding BP neural network, this paper improves it by considering momentum attachment and dynamic learning rate. The simulation is driven by NS3, where prediction ratio and communication time are regarded as two performance evaluation metrics, and the experimental results demonstrate that the proposed roller track analysis method can be used for the falling prevention of roller athlete timely.

This paper is the first paper to propose roller track analysis based on the emerging computer technologies, which has the important reference value in the field of sports. However, this paper also has some limitations. At first, the system platform of AMI is not implemented. Then, the simulation only depends on NS3 and lacks of the verification based on the test-bed. At last, the posture variety of roller

athlete is not considered. In future, the abovementioned limitations will be overcome.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author has no conflicts of interest.

References

- [1] Q. Bu, S. Lan, C. Gu et al., "A judgment framework for roller skating using faster R-CNN," in *Proceedings of the 2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, pp. 53–57, Xi'an, China, October 2017.
- [2] G. Olmi, "Failure of the chassis of roller skates for agonistic figure skating," *Case Studies in Engineering Failure Analysis*, vol. 3, pp. 62–67, 2015.
- [3] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, In press.
- [4] P. Suresh Kumar, H. S. Behera, A. Kumari et al., "Advancement from neural networks to deep learning in software effort estimation: perspective of two decades," *Computer Science Review*, vol. 38, pp. 1–32, 2020.
- [5] S. Kaviani and I. Sohn, "Influence of random topology in artificial neural networks: a survey," *ICT Express*, vol. 6, no. 2, pp. 145–150, 2020.
- [6] Z. Xu, R. Zhang, and W. Jing, "When does online BP training converge?" *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1529–1539, 2009.
- [7] W. Wu, G. Feng, Z. Li, and Y. Xu, "Deterministic convergence of an online gradient method for BP neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 533–540, 2005.
- [8] M. Paolanti and E. Frontoni, "Multidisciplinary pattern recognition applications: a review," *Computer Science Review*, vol. 37, pp. 1–23, 2020.
- [9] X.-Y. Zhang, C.-L. Liu, and C. Y. Suen, "Towards robust pattern recognition: a review," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 894–922, 2020.
- [10] V. Jacobson, J. J. Li, K. Tapia et al., "Visualizing neural networks for pattern recognition," in *Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence-PRAI 2018*, pp. 18–22, Union, NJ, USA, August 2018.
- [11] F. Moggio, M. Boldi, S. Canale et al., "5G EVE a european platform for 5G application deployment," in *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pp. 124–125, London, UK, September 2020.
- [12] A. Jain, E. Lopez-Aguilera, and I. Demirkol, "Are mobility management solutions ready for 5G and beyond?" *Computer Communications*, vol. 161, pp. 50–75, 2020.
- [13] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 905–929, 2020.
- [14] A. G. Blaiech, K. Ben Khalifa, C. Valderrama, M. A. C. Fernandes, and M. H. Bedoui, "A survey and

- taxonomy of FPGA-based deep learning accelerators,” *Journal of Systems Architecture*, vol. 98, pp. 331–345, 2019.
- [15] P. Cascaval and D. Cascaval, “March test algorithm for unlinked static reduced three-cell coupling faults in random-access memories,” *Microelectronics Journal*, vol. 93, pp. 1–4, 2019.
- [16] B. K. Muite, “A numerical comparison of Chebyshev methods for solving fourth order semilinear initial boundary value problems,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 2, pp. 317–342, 2010.
- [17] X. Wang, J. Lv, M. Huang et al., “Energy-efficient ICN routing mechanism with QoS support,” *Computer Networks*, vol. 131, pp. 38–51, 2018.
- [18] NS3, <https://www.nsnam.org>.

Research Article

In-Network Caching and Edge Computing-Based Live Broadcasting Optimization for Football Competitions

Zhigang Li 

Jilin Institute of Physical Education, Changchun 130022, China

Correspondence should be addressed to Zhigang Li; wxcf2000@sina.com

Received 11 November 2020; Accepted 25 November 2020; Published 12 December 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Zhigang Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The business of football competitions is called the number one sport in the world, thanks to more than one billion people's attention. With the development of big convergence media, the live broadcasting of football competitions gradually becomes industrialization and commercialization, which has a direct relationship with economic growth. For the live broadcasting of football competitions, the users focus more on quality of experience, i.e., definition and instantaneity. In terms of such two metrics, the current live broadcasting schemes are difficult to cover them well. Therefore, this paper exploits the emerging in-network caching and edge computing technologies to optimize the live broadcasting of football competitions, shorten for IELB. At first, the live broadcasting optimization framework based on in-network caching and edge computing is presented. Then, the auction-based method is used to address the task scheduling problem in the edge computing. In addition, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission and guarantee users to obtain the contents of football competitions as quickly as possible. The proposed IELB has been verified based on the collected real football competitions dataset by evaluating response time, and the experimental results demonstrate that IELB is feasible and efficient.

1. Introduction

With the progress of the times, the industrialization and commercialization development of football has been rapidly promoted. In all outdoor sports, the football obtains the highest productive value, owns the biggest influence, and has the most widespread concern. According to the statistical data, the total output value of football per year can account for 43.5% of that of all sports, reaching 400 billion dollars and exceeding some developed countries and regions' GDP. Thus, the football is worthy of being called the number one sport in the world [1, 2]. Furthermore, according to the statistical data from FIFA, there are 1.6 million teams with more than 0.2 billion athletes worldwide playing the various football competitions by the end of 2019 [3]. Especially, with the rapid development of mobile Internet and big convergence media, the live broadcasting of football competitions [4] gradually becomes industrialization and commercialization, which has attracted many Internet companies to

develop the related businesses. In fact, for the live broadcasting of football competitions, the users pay more attention to quality of experience (QoE) [5], i.e., high definition and nice instantaneity, which needs that the whole process of live broadcasting guarantees the fluency. However, for the two evaluation metrics (i.e., high definition and nice instantaneity), the current live broadcasting schemes based on content delivery networks (CDN) [6–8] paradigm are very difficult to cover them well. In other words, it is necessary to explore the new networking paradigm to support the live broadcasting of football competitions with high definition and nice instantaneity.

Information centric networking (ICN) [9–11] is a new emerging and popular networking paradigm, which inherently owns one considerably important feature, i.e., in-network caching, that is to say, the ICN router also has the caching ability [12, 13] to store the contents that are related with the live broadcasting of football competitions. However, different from both the CDN server and traditional

router's buffer, the cache size of the ICN router is between them, and the ICN router can also be deployed at the edge nodes to satisfy the users' requests as many as possible. In other words, this paper uses the in-network caching feature of ICN to help the live broadcasting optimization of football competitions so as to obtain high definition and nice instantaneity. In fact, the usage of ICN has two obvious advantages. On the one hand, there is no need for fanfare to deploy the expensive CDN server. On the other hand, the ICN router can be regarded as a nice and big cache pool to accommodate more intermediate data stream. The two advantages can motivate the application of ICN into live broadcasting of football competitions.

As abovementioned, the ICN router can be deployed at the edge nodes which usually refer to the mobile devices, such as smartphones. Given this, the situation of mobile edge computing (MEC) [14–16] has to be considered and addressed. In fact, MEC has the widespread application value, especially for the optimization of live broadcasting. To be specific, the mobile devices have the limited storage resources and computation resources, and thus, it is very difficult for them to completely handle all tasks which are related to the live broadcasting. Therefore, it is one possible solution to offload some complex tasks at the edge server for computing, and the remaining simple tasks are performed at the local device. Under this condition, it is very important to address the task scheduling problem in the edge computing. At present, there have been some task scheduling methods [17–23], including artificial intelligence (AI) based ones [24, 25], but they usually cannot obtain the fast response speed and the low energy consumption. As a conclusion, it is necessary to explore the new method to address the task scheduling problem generated from the live broadcasting optimization.

Furthermore, the live broadcasting of football competition cannot do without the transmission of data stream. In other words, the video transmission is an indispensable operation to connect the contents provider and the users [26]. Regarding the video transmission, the video compression algorithm is very important, which can accelerate the video transmission and guarantee users to obtain the contents of football competitions as quickly as possible. However, the current video compression algorithms exit some limitations [27–31], and especially, the frame loss rate and the transmission time cannot reach the satisfactory level. Therefore, the video compression algorithm is also studied in this paper.

With the above consideration, this paper optimizes the live broadcasting of football competitions by using three aspects of technical points, i.e., the in-network caching feature of ICN, the task scheduling of edge computing, and the video compression of video transmission, called IELB. To sum up, the major contributions of this paper are concluded as follows. (i) The live broadcasting optimization framework based on in-network caching and edge computing is presented. (ii) The auction-based method is used to address the task scheduling problem in the edge computing. (iii) A video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission.

The rest of this paper is structured as follows. Section 2 reviews the related work. Section 3 presents the comprehensive system framework. Section 4 introduces the auction-based task scheduling. Section 5 proposes the video compression algorithm. Section 6 shows the significant experiment results. Section 7 concludes this paper.

2. Literature Review

2.1. Task Scheduling. There have been some research studies on task offloading proposals in the edge computing. In particular, some review papers have presented the comprehensive summary, such as [32–35]. Furthermore, in [17], the authors formulated successful computation probability, successful communication probability, and successful edge computing probability for offloading tasks to the MEC server. In addition, they also analyzed by simulation how the formulated probabilities vary for different sizes of task, task's target latency, and task arrival rate at the MEC server helping users to make offloading decision. In [18], the authors critically analyzed the resource-intensive nature of the latest existing computational offloading techniques for MEC and highlighted technical issues in the establishment of distributed application processing platforms at runtime where a prototype application was evaluated with different computation intensities in a real MEC environment. In [19], the authors presented a collaborative approach based on MEC and cloud computing that offloaded services to automobiles in vehicular networks. Meanwhile, a cloud-MEC collaborative computation offloading problem was formulated through jointly optimizing computation offloading decision and computation resource allocation. Besides, they also proposed a collaborative computation offloading and resource allocation optimization scheme and designed a distributed computation offloading and resource allocation algorithm to achieve the optimal solution. In [20], the authors proposed a price-based distributed method to manage the offloaded computation tasks from users. Therein, a Stackelberg game was formulated to model the interaction between the edge cloud and users so as to maximize the revenue subject to its finite computation capacity. For given prices, each user locally made offloading decision to minimize its own cost which was defined as latency plus payment. Depending on the edge cloud's knowledge of the network information, they developed the uniform and differentiated pricing algorithms, which could both be implemented in the distributed manner. In [21], the authors proposed a multiuser noncooperative computation offloading game to adjust the offloading probability of each vehicle in vehicular MEC networks and designed the payoff function considering the distance between the vehicle and MEC access point, application and communication model, and multivehicle competition for MEC resources. Also, they constructed a distributed best response algorithm based on the computation offloading game model to maximize the utility of each vehicle and demonstrated that the strategy could converge to a unique and stable equilibrium under certain conditions. In [22], the authors used the partial computation offloading

problem for multiuser in mobile edge computing environment with the multiwireless channel. The computation overhead model was built based on game theory. Then, the partial computation offloading algorithm with low time complexity was given to achieve the Nash equilibrium. In [23], the authors captured a user-centric view to tackle the offloading scheduling problem via jointly allocating communication and computation resources with consideration of the QoE of users where they formulated the design as a mix-integer nonlinear programming problem and solved it in an efficient way by the branch-and-bound method.

2.2. Video Compression. A number of video compression methods have been proposed, including some comprehensive survey papers, such as [36–38]. Furthermore, in [27], the authors maintained that a significant reduction in file size without sacrificing the visual quality could be achieved by using several efficient compression techniques. To this end, they proposed the video compression method by using global affine frame reconstruction, which involved affine parameter estimation for motion estimation and affine warping for motion compensation where the motion parameters were estimated and stored as compressed data. In [28], the authors proposed an adaptive transfer function based on perceptual quantizer for video compression, which used a fixed mapping curve from luminance to luma, i.e., the proposed transfer function adaptively mapped luminance to luma according to the contents. In [29], the authors introduced a hybrid spatially and temporally constrained content-adaptive tone mapping operator to convert the input high dynamic range video into a tone mapped video sequence, which was then encoded using the high efficiency video coding standard. In particular, the proposed tone mapped video simultaneously exploited intraframe spatial redundancies and preserved interframe temporal coherence of the tone mapped video sequence. In [30], the authors proposed a novel rate control scheme in H.264 to control the rate of compression ratio where the level of compression was decided with the help of the rate controller scheme. They measured the quality of the transmitted video and available bandwidth with the proposed technique and also built a quality multimedia content and transferred over the transmitter. In [31], the authors provided a lightweight video compression scheme through interframe and intraframe compression. In interframe compression, redundant frames were removed by a proposed interpolation search-based method and a lightweight edge detection technique. Then, intraframe compression was performed by a proposed adaptive column dropping technique modifying an existing technique. Besides, they also devised two reconstruction filters targeting to improve reconstruction quality.

3. System Framework

By using the in-network caching ability of ICN and edge computing, the proposed system framework of live broadcasting optimization of football competitions is shown in Figure 1. We can see that there are two kinds of different

servers, i.e., contents provider server used to store the football competitions related videos and edge computing server used to compute the complex tasks. The whole live broadcasting refers to the video transmission from the contents provider to the mobile device via some indeterminate ICN routers used to store the hot data stream. In particular, during the process of video transmission, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission. The live broadcasting contents arrive at the mobile device. Given the limited storage resources and computation resources, the complex tasks are scheduled to the edge computing server for computing, while the remaining simple tasks are performed at the local mobile device. Regarding this, the auction-based method is used to address the task scheduling problem in the edge computing.

In particular, the ICN router stays at the network level instead of the application level (e.g., CDN), which is assumed with the enough cache size to store the hot data stream. In addition, this paper also assumes that all mobile devices have the same configuration. In terms of two servers, the contents provider server has the very abundant space to store these videos of football competitions, just like a top-level distribution server. Differently, the edge computing server is only a high performance computing and storage server, and its computing ability and storage ability is not a circumstance to the contents provider server. The whole workflow of live broadcasting optimization of football competitions is shown in Figure 2. According to the above statements, task scheduling and video compression are two major research points in this paper, which will be addressed in the following sections.

4. Task Scheduling

The whole task scheduling in the edge computing includes two parts, i.e., offloading decision used to determine whether the tasks need to be offloaded and the scheduling method used to make resources computation and allocation.

4.1. Offloading Decision. Regarding whether the task is offloaded, its decision depends on whether the task's running time and energy consumption can be decreased in case of performing task offloading. Given this, two conditions, i.e., local performing and offloading performing, are considered and analyzed.

At first, for the arbitrary task τ_i , when it is performed at the local mobile device, the required time is defined as follows.

$$t_l = \frac{C_i}{v_l}, \quad (1)$$

where c_i is the required computation resources which refer to the number of CPU cycles, and v_l is the execution rate of CPU. Furthermore, the required energy consumption is defined as follows.

$$e_l = t_l p_l, \quad (2)$$

where p_l is the power of the mobile device.

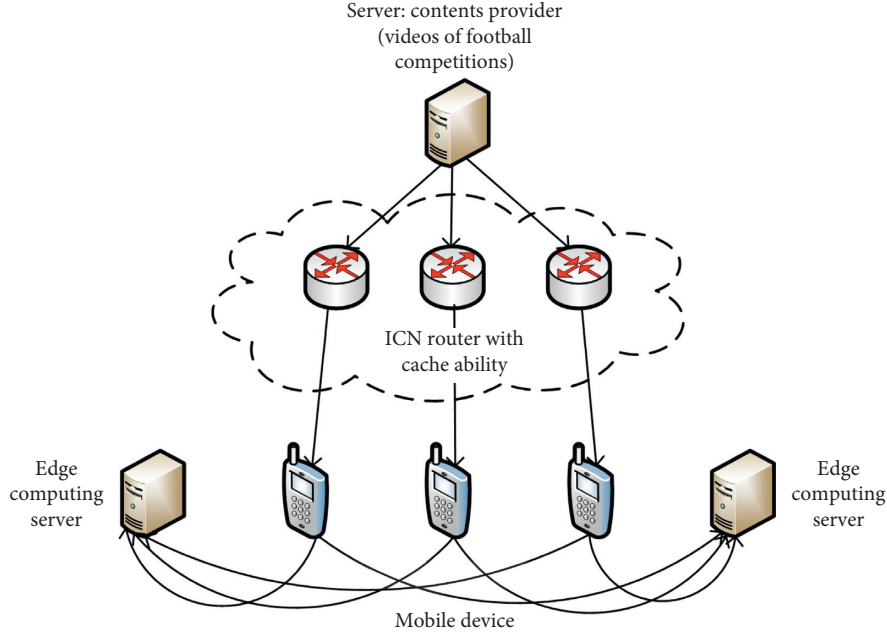


FIGURE 1: The proposed system framework of live broadcasting optimization.

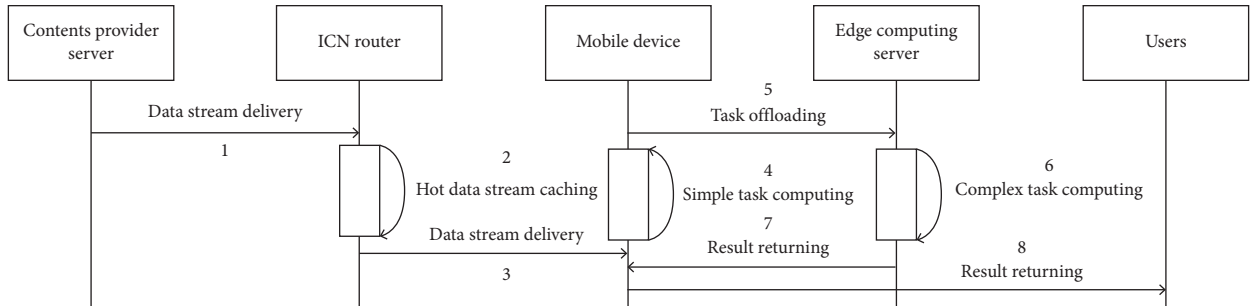


FIGURE 2: The workflow of IELB.

Then, when task_{*i*} is performed at the edge computing server, the required time t_{off} is composed of three parts, i.e., transmission time of task_{*i*} from the mobile device to the edge computing server, computing time at the edge computing server, and returning time of computing result from the edge computing server to the mobile device, denoted by t_{up} , t_{down} , and t_e , respectively. Mathematically,

$$t_{\text{off}} = t_{\text{up}} + t_e + t_{\text{down}}. \quad (3)$$

Consider that the size of the output result is far smaller than that of input data for most applications, and the returning time has no the significant influence on the total required time. On this basis, the above equation (3) is modified as follows.

$$t_{\text{off}} = t_{\text{up}} + t_e,$$

$$t_{\text{up}} = \frac{D_i}{W \log_2(1 + (p_l \text{Los}/N))}, \quad (4)$$

$$t_e = \frac{C_i}{v_e}.$$

Among them, D_i is the size of task_{*i*}, N is the Gaussian noise power of channel, W is the bandwidth of channel, Los is the transmission gain, and v_e is the edge computing server CPU's execution rate. To sum up, the total required time under such condition is expressed as follows.

$$t_{\text{off}} = \frac{D_i}{W \log_2(1 + (p_l \text{Los}/N))} + \frac{C_i}{v_e}. \quad (5)$$

Let e_{off} denote the total required energy consumption under such condition, as follows.

$$e_{\text{off}} = \frac{p_l D_i}{W \log_2(1 + (p_l \text{Los}/N))}. \quad (6)$$

Moreover, when the consumed time and energy under the task offloading condition are smaller than those under the local performing condition, the task should be performed in the offloading way. For this purpose, the following two constraint conditions should be satisfied.

$$\begin{aligned} t_{\text{off}} &< t_l, \\ e_{\text{off}} &< e_l. \end{aligned} \quad (7)$$

For these two constraint conditions, the further derivation results are shown as follows.

$$v_e > \frac{v_l C_i}{C_i - ((D_i v_l)/(W \log_2(1 + (p_l \text{Los}/N))))}, \quad (8)$$

$$W > \frac{v_l p_l D_i}{p_l C_i \log_2(1 + (p_l \text{Los}/N))}, \quad (9)$$

which indicates that, if the computing ability of edge computing server satisfies in equation (8) and the current network bandwidth environment satisfies in equation (9), the task will be performed in the offloading way.

4.2. Scheduling Method. This paper uses the auction method [39] to address the scheduling strategy, including two roles, i.e., buyers and sellers. In terms of n buyers, they refer to these tasks, i.e., the set of buyers is denoted by $\text{TASK} = \{\text{task}_1, \text{task}_2, \dots, \text{task}_n\}$. Each task has two attributes, i.e., $\text{task}_i(t)$ and $\text{task}_i(p)$ which denote the required time to complete task_i and the bid of task_i , respectively. In terms of m sellers, they refer to the unoccupied virtual machines (VMs), i.e., the set of sellers is denoted by $\text{VM} = \{\text{vm}_1, \text{vm}_2, \dots, \text{vm}_m\}$. Each VM is equipped with one attribute, i.e., $\text{vm}_i(t)$, which denotes the rental time of vm_i . Furthermore, let $\text{price}_{\text{ini}}$ denote the initial bid price, and it is defined as follows.

$$\text{price}_{\text{ini}} = \frac{1}{t_l - t_{\text{off}}}. \quad (10)$$

Although equation (10) considers the emergency degree of task, there exists the unfairness. Especially, the task in the offloading way will gradually lose the price advantage. Therefore, this paper proposes the compensation strategy in terms of the condition of bidding failure. Mathematically, equation (10) is modified as follows.

$$\text{price} = \frac{1}{t_l - t_{\text{off}} - sN_{\text{fai}}}, \quad (11)$$

where s is the length of time slice, and N_{fai} is the number of bidding failures.

The derivation operation in terms of s is performed, and the following equation is obtained.

$$\frac{\partial \text{price}}{\partial s} = \frac{s}{(t_l - t_{\text{off}} - sN_{\text{fai}})^2}, \quad (12)$$

which indicates that the derivation value increases with the increase of N_{fai} . In other words, the large derivation value means the fast increasing speed in terms of price. Therefore, another meaning of equation (12) is that, more price compensations are performed based on more bidding failures.

Besides task and VM, there is a set of intermediate variables (denoted by CA) used to store some tasks satisfying auction condition. If $\text{task}_i(t) < \text{vm}_i(t)$, it means that task_i satisfies the auction condition, and it is added into CA. Otherwise, task_i is marked as the bidding failure. Regarding the whole auction process, reference [39] has presented the detailed steps, and this paper does give the corresponding steps no longer. In particular, the time complexity is $O(m \log m)$ rather than $O(nm)$.

5. Video Compression

The whole video compression process depends on the motion compensation which is used to realize the motion evaluation among video frames and pixel information compensation. In particular, the part is completed based on adaptive convolution kernel [40]. Inspired by Simon et al. [40], this paper learns the motion offset in terms of the continuous video frames so as to realize the motion evaluation among video frames. Regarding such process, it is defined as follows.

$$I'_{c2} = f(I_{c1}, I_{c2}) = I_{c1} + \Delta_{i,i+1}. \quad (13)$$

Among them, I_{ci} denotes the frame i , $\Delta_{i,i+1}$ denotes the offset from frames i to $i+1$, I'_{ci} denotes the motion compensation on I_{ci} , and f is the mapping function when the adaptive convolution kernel is used.

When the adaptive convolution kernel is used, it needs two convolution kernels to do the motion prediction, denoted by $K_h(x, y)$ and $K_v(x, y)$, which are considered as the horizontal vector and the vertical vector of $K(x, y)$, respectively. Mathematically,

$$K(x, y) = K_h(x, y) * K_v(x, y), \quad (14)$$

where $*$ denotes the convolution operation. In particular, the rectified linear units function is added after each convolution operation so as to enhance the express ability of the network. On this basis, the general motion compensation on I_{ci} is defined as follows.

$$I'_{ci}(x, y) = K_h(x, y) * K_v(x, y) * I_{ci}(x, y). \quad (15)$$

In summary, by using the network self-learning operation, the motion compensation is analyzed and performed based on $K_h(x, y)$ and $K_v(x, y)$ with the nonlinear mapping relationship. In terms of $K(x, y)$ with $K \times K$ convolution kernels, the current number of parameters is $2K$ instead of K^2 , which indicates that two one-dimensional convolution kernels have better computation efficiency than one two-dimensional convolution kernel.

TABLE 1: The average transmission times (ms) for ATS, IoTJ, and CoMNeT.

| Data size | 10 GB | 20 GB | 30 GB | 40 GB | 50 GB | 60 GB |
|-----------|--------|--------|--------|--------|--------|--------|
| ATS | 26.351 | 29.462 | 35.662 | 43.506 | 54.102 | 67.389 |
| IoTJ | 35.118 | 37.525 | 42.727 | 52.354 | 64.178 | 75.065 |
| CoMNeT | 48.672 | 49.192 | 55.881 | 64.394 | 77.296 | 86.406 |

TABLE 2: The independent transmission times (ms) for ATS, IoTJ, and CoMNeT where the data size is 50 GB.

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| ATS | 54.069 | 53.945 | 54.261 | 53.886 | 54.612 | 54.119 | 53.687 | 54.234 |
| IoTJ | 61.598 | 65.363 | 68.552 | 64.364 | 62.506 | 62.395 | 63.094 | 65.556 |
| CoMNeT | 78.932 | 79.642 | 77.265 | 78.345 | 77.064 | 76.355 | 75.106 | 75.663 |

6. Experiment Results

This paper makes three parts of experiments. At first, the proposed auction-based task scheduling strategy called ATS is verified. Then, the proposed adaptive convolution kernel-based video compression scheme called ACK is verified. Finally, the proposed whole IELB is verified. Among them, the first part considers transmission time and energy consumption as the evaluation metrics where references [21, 22] are used as the baselines, shorten for IoTJ and CoMNeT, respectively. The second part considers frame loss rate and transmission time as the evaluation metrics where references [30, 31] are used as the baselines, shorten for CoMCoM and JVCIR, respectively. The third part considers response time and QoE of users as the evaluation metrics by collecting the 1000 football competitions and testing 500 users.

6.1. Experiments on Task Scheduling. The average transmission times under different data sizes (from 10 GB to 60 GB) for ATS, IoTJ, and CoMNeT are shown in Table 1. We can see that the proposed ATS task scheduling strategy in the edge computing has the smallest average transmission time, followed by IoTJ and CoMNeT. In addition, with the increase of data size, the corresponding average transmission time increases. It suggests that the auction-based task scheduling strategy is more efficient than IoTJ and CoMNeT. Furthermore, when the data size is 50 GB, the independent transmission times under different experiments (from 1 to 8) for ATS, IoTJ, and CoMNeT are shown in Table 2. It is obvious that ATS has the smallest transmission time for each experiment. In particular, for 8 experiment results in terms of the independent transmission time, the proposed ATS has the best stability, followed by CoMNeT and IoTJ. It suggests that the proposed task scheduling is the optimal.

Moreover, the average energy consumption under different data sizes (from 10 GB to 60 GB) for ATS, IoTJ, and CoMNeT are shown in Table 3. We can see that the proposed ATS consumes the smallest energy. In summary, ATS has the smallest transmission time and the smallest energy consumption, which indicates that the proposed task scheduling strategy in the edge computing is considerably satisfactory.

TABLE 3: The average energy consumption (kJ) for ATS, IoTJ, and CoMNeT.

| Data size | 10 GB | 20 GB | 30 GB | 40 GB | 50 GB | 60 GB |
|-----------|-------|-------|-------|-------|-------|-------|
| ATS | 50263 | 50652 | 51297 | 52096 | 53121 | 54465 |
| IoTJ | 68355 | 68626 | 69172 | 70104 | 71233 | 72531 |
| CoMNeT | 83623 | 84069 | 84707 | 85391 | 85426 | 86829 |

6.2. Experiments on Video Compression. The average frame loss rates under different resolution ratios (i.e., 320P, 480P, 720P, and 1080P) for ACK, CoMCoM, and JVCIR are shown in Table 4. We can see that the average frame loss rate of ACK is the lowest. Although when the resolution ratio is 320P, their average frame loss rates have no significant difference, and the change of ACK is the linear while CoMCoM and JVCIR show the index movement. It suggests that the proposed video compression scheme can guarantee the integrated video transmission. Furthermore, the average transmission time under different resolution ratios for ACK, CoMCoM, and JVCIR are shown in Table 5. We can see that the average transmission time of ACK from the contents provider server to the mobile device is the smallest. Similarly, when the resolution ratio is 320P, their average transmission times have no significant difference. In particular, we also see that the average transmission time generated from video transmission is larger than that generated from the task offloading.

6.3. Experiments on Live Broadcasting Optimization. When the data size is 60 GB and the resolution ratio is 720P, the independent response times under different experiments (from 1 to 10) for IELB are shown in Table 6. We can see that the response time is around 120 ms with the ms-level, which can be acceptable by users.

Furthermore, this paper evaluates QoE of users where the QoE is divided into five grades, i.e., very satisfactory, satisfactory, borderline, dissatisfaction, and very dissatisfaction. From two perspectives (i.e., users and football competition), the evaluation results on QoE of users are shown in Table 7. We can see that the satisfaction rate can reach 100%, which indicates that the proposed live broadcasting optimization of football competition is feasible and efficient.

TABLE 4: The average frame loss rates (%) for ACK, CoMCoM, and JVCIR.

| Resolution ratio | 320P | 480P | 720P | 1080P |
|------------------|-------|-------|-------|-------|
| ACK | 0.015 | 0.039 | 0.051 | 0.073 |
| CoMCoM | 0.017 | 0.062 | 0.103 | 0.218 |
| JVCIR | 0.018 | 0.071 | 0.135 | 0.267 |

TABLE 5: The average transmission times (ms) for ACK, CoMCoM, and JVCIR.

| Resolution ratio | 320P | 480P | 720P | 1080P |
|------------------|--------|--------|--------|--------|
| ACK | 36.517 | 40.268 | 47.516 | 53.992 |
| CoMCoM | 47.068 | 53.267 | 64.333 | 82.908 |
| JVCIR | 61.285 | 68.593 | 79.112 | 97.267 |

TABLE 6: The average independent response times (ms) for IELB.

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| IELB | 120.236 | 119.351 | 120.692 | 119.864 | 121.006 | 120.443 | 121.507 | 119.868 | 120.612 | 121.323 |

TABLE 7: The statistical results on QoE of users.

| Grade | Very satisfactory | Satisfactory | Borderline | Dissatisfaction | Very dissatisfaction | Satisfaction rate (%) |
|---------|-------------------|--------------|------------|-----------------|----------------------|-----------------------|
| #Users | 496 | 4 | 0 | 0 | 0 | 100 |
| #Videos | 998 | 2 | 0 | 0 | 0 | 100 |

7. Conclusions

This paper optimizes the live broadcasting of football competitions by using three aspects of technical points, i.e., the in-network caching feature of ICN, the task scheduling of edge computing, and the video compression of video transmission. At first, the live broadcasting optimization framework based on in-network caching and edge computing is presented. Second, the auction-based method is used to address the task scheduling problem in the edge computing. Third, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission. For these proposed strategies, this paper makes three kinds of experiments: (i) the proposed auction-based task scheduling strategy is verified by testing transmission time and energy consumption; (ii) the proposed adaptive convolution kernel-based video compression scheme is verified by testing frame loss rate and transmission time; and (iii) the proposed whole IELB is verified by testing response time and QoE of users. The experimental results demonstrate that the proposed IELB is feasible and efficient.

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.

References

- [1] T. Dima, "The business model of European football club competitions," *Procedia Economics and Finance*, vol. 23, pp. 1245–1252, 2015.
- [2] S. J. Koopman and R. Lit, "Forecasting football match results in national league competitions using score-driven time series models," *International Journal of Forecasting*, vol. 35, no. 2, pp. 797–809, 2019.
- [3] J. M. Buldu, D. R. Antequera, and J. Aguirre, "The resumption of sports competitions after COVID-19 lockdown: the case of the Spanish football league, chaos," *Solitons & Fractals*, vol. 138, pp. 1–7, 2020.
- [4] F. Zhou, L. Chen, and Q. Su, "Understanding the impact of social distance on users' broadcasting intention on live streaming platforms: a lens of the challenge-hindrance stress perspective," *Telematics and Informatics*, vol. 41, pp. 46–54, 2019.
- [5] W.-H. Hsu and C.-H. Lo, "QoS/QoE mapping and adjustment model in the cloud-based multimedia infrastructure," *IEEE Systems Journal*, vol. 8, no. 1, pp. 247–255, 2014.
- [6] S. Cui, M. R. Asghar, and G. Russello, "Multi-CDN: towards privacy in content delivery networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 984–999, 2020.
- [7] A. Passarella, "A survey on content-centric technologies for the current internet: CDN and P2P solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2011.
- [8] N. Anjum, D. Karamshuk, M. Shikh-Bahaei, and N. Sastry, "Survey on peer-assisted content delivery networks," *Computer Networks*, vol. 116, pp. 79–95, 2017.
- [9] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information

- centric network: research challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 52, pp. 1–10, 2015.
- [10] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, “Congestion control in named data networking—a survey,” *Computer Communications*, vol. 86, pp. 1–11, 2016.
- [11] J. Lv, X. Wang, and M. Huang, “ACO-inspired ICN routing mechanism with mobility support,” *Applied Soft Computing*, vol. 58, pp. 427–440, 2017.
- [12] N. Lal, S. Kumar, G. Kadian, and V. K. Chaurasiya, “Caching methodologies in content centric networking (CCN): a survey,” *Computer Science Review*, vol. 31, pp. 39–50, 2019.
- [13] M. Zhang, H. Luo, and H. Zhang, “A survey of caching mechanisms in information-centric networking,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.
- [14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [15] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [16] H. Elazhary, “Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions,” *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.
- [17] B. Bahadur Bista, J. Wang, and T. Takata, “Probabilistic computation offloading for mobile edge computing in dynamic network environment,” *Internet of Things*, vol. 11, pp. 1–11, 2020.
- [18] M. Shiraz, M. Sookhak, A. Gani, and S. A. A. Shah, “A study on the critical analysis of computational offloading frameworks for mobile cloud computing,” *Journal of Network and Computer Applications*, vol. 47, pp. 47–60, 2015.
- [19] J. Zhao, Q. Li, Y. Gong, and K. Zhang, “Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [20] M. Liu and Y. Liu, “Price-based distributed offloading for mobile-edge computing with computation capacity constraints,” *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 420–423, 2018.
- [21] Y. Wang, P. Lang, D. Tian et al., “A game-based computation offloading method in vehicular multiaccess edge computing networks,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4987–4996, 2020.
- [22] S. Zhou and J. Waqas, “The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment,” *Computer Networks*, vol. 178, pp. 1–12, 2020.
- [23] J. Luo, X. Deng, H. Zhang, and H. Qi, “QoE-driven computation offloading for edge computing,” *Journal of Systems Architecture*, vol. 97, pp. 34–39, 2019.
- [24] L. Ma, S. Cheng, and Y. Shi, “Enhancing learning efficiency of brain storm optimization via orthogonal learning design,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, p. 1, 2020.
- [25] S. Cheng, L. Ma, H. Lu, X. Lei, and Y. Shi, “Evolutionary computation for solving search-based data analytics problems,” *Artificial Intelligence Review*, 2020.
- [26] Y.-M. Hsiao, J.-F. Lee, J.-S. Chen, and Y.-S. Chu, “H.264 video transmissions over wireless networks: challenges and solutions,” *Computer Communications*, vol. 34, no. 14, pp. 1661–1672, 2011.
- [27] D. R. J. Dolly, G. J. Bala, and J. D. Peter, “Performance enhanced spatial video compression using global affine frame reconstruction,” *Journal of Computational Science*, vol. 18, pp. 1–11, 2017.
- [28] S. Yu and C. Jung, “Adaptive perceptual quantizer for high dynamic range video compression,” *Journal of Visual Communication and Image Representation*, vol. 58, pp. 25–36, 2019.
- [29] C. Ozcinar, P. Lauga, G. Valenzise, and F. Dufaux, “Spatio-temporal constrained tone mapping operator for HDR video compression,” *Journal of Visual Communication and Image Representation*, vol. 55, pp. 166–178, 2018.
- [30] K. Siva Kumar, S. Sasi Kumar, and N. Mohan Kumar, “Efficient video compression and improving quality of video in communication for computer encoding applications,” *Computer Communications*, vol. 153, pp. 152–158, 2020.
- [31] T. Pal and S. Das Bit, “Low overhead spatiotemporal video compression over smartphone based delay tolerant network,” *Journal of Visual Communication and Image Representation*, vol. 70, pp. 1–20, 2020.
- [32] H. Lin, S. Zeadally, Z. Chen et al., “A survey on computation offloading modeling for edge computing,” *Journal of Network and Computer Applications*, vol. 169, pp. 1–25, 2020.
- [33] Q.-H. Nguyen and F. Dressler, “A smartphone perspective on computation offloading—a survey,” *Computer Communications*, vol. 159, pp. 133–154, 2020.
- [34] S. Ali, M. Ghobaei-Arani, and S. Ali, “A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective,” *Computer Networks*, vol. 182, pp. 1–24, 2020.
- [35] A. Bhattacharya and P. De, “A survey of adaptation techniques in computation offloading,” *Journal of Network and Computer Applications*, vol. 78, pp. 97–115, 2017.
- [36] A. J. Tabatabai, R. S. Jasinschi, and T. Naveen, “Motion estimation methods for video compression—a review,” *Journal of the Franklin Institute*, vol. 335, no. 8, pp. 1411–1441, 1998.
- [37] J.-S. Lee and T. Ebrahimi, “Perceptual video compression: a survey,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 684–697, 2012.
- [38] H. A. Choudhury, N. Sinha, and M. Saikia, “Nature inspired algorithms (NIA) for efficient video compression—a brief study,” *Engineering Science and Technology, an International Journal*, vol. 23, no. 3, pp. 507–526, 2020.
- [39] I. Abraham, S. Athey, M. Babaioff, and M. D. Grubb, “Cookies: peaches, lemons, and cookies: designing auction markets with dispersed information,” *Games and Economic Behavior*, vol. 124, pp. 454–477, 2020.
- [40] N. Simon, L. Mai, and F. Liu, “Video frame interpolation via adaptive convolution,” 2017, <https://arxiv.org/abs/1703.07514>.

Research Article

DRL-Based Edge Computing Model to Offload the FIFA World Cup Traffic

Hongyi Li¹ and Xinrui Che² 

¹Dalian Maritime University, Dalian 116026, China

²Liaoning Police College, Dalian 116036, China

Correspondence should be addressed to Xinrui Che; chexinrui@lnpc.cn

Received 14 September 2020; Revised 22 October 2020; Accepted 1 November 2020; Published 19 November 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Hongyi Li and Xinrui Che. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the volume of global video traffic has been increasing rapidly and it is considerably significant to offload the traffic during the process of video transmission and improve the experience of users. In this paper, we propose a novel traffic offloading strategy to provide a feasible and efficient reference for the following 2022 FIFA World Cup held in Qatar. At first, we present the system framework based on the Mobile Edge Computing (MEC) paradigm, which supports transferring the FIFA World Cup traffic to the mobile edge servers. Then, the Deep Reinforcement Learning (DRL) is used to provide the traffic scheduling method and minimize the scheduling time of application programs. Meanwhile, the task scheduling operation is regarded as the process of Markov decision, and the proximal policy optimization method is used to train the Deep Neural Network in the DRL. For the proposed traffic offloading strategy, we do the simulation based on two real datasets, and the experimental results show that it has smaller scheduling time, higher bandwidth utilization, and better experience of user than two baselines.

1. Introduction

The mobile Internet and Internet of Things (IoT) have been developed in recent years [1] especially during the process of building the smart cities [2], which has been generating the billions of Internet traffic due to the sharp increasing number of mobile devices (such as smart phone and wearable monitors). For example, the Cisco Annual Internet Report (CAIR) [3] shows that the total number of global mobile devices will grow from 5.1 billion (66 percent of population) in 2018 to 5.7 billion (71 percent of population) by 2023. In addition, the global IP traffic is expected to reach 5.3 ZB by 2023 [4] and the video traffic will account for 67.5% of the global IP traffic due to the introduction of new techniques and applications such as 4K/8K [5] and AR/VR [6] which are the necessary products in the smart cities. As we know, there are usually three kinds of video transmission, i.e., video on demand, carousel, and live streaming [7]. At the era of mobile Internet, more and more users pay attention to the applications of live streaming. Particularly, at

the special time frame, the internationalized sport events which belong to the field of live streaming attract a large number of audiences without doubt. For example, the FIFA World Cup is the famous live sport event. Given this, this paper plans to investigate the FIFA World Cup and gives a network solution from the perspective of traffic offloading in order to provide support for the following 2022 FIFA World Cup in Qatar.

The traditional video delivery strategies usually depend on the technique of Content Delivery Networks (CDN) [8, 9]. In other words, the Internet content providers deliver the abundant contents to the edge users based on CDN in the push mode, in which the intermediate and the edge servers store the hop contents in advance. However, the strategies based on CDN are not suitable to the traffic offloading for the FIFA World Cup and the main reasons are analyzed as follows. The volume of the FIFA World Cup traffic is the unspeakably large and the corresponding features show the periodicity, abruptness, and explosivity. For the traffic, CDN fails to deliver them to the Metropolitan Area Network

(MAN) or the Access Network (AN) which is very close to the mobile end-users, due to the fact that CDN servers are very expensive and it is impossible for the Internet content providers to deploy many CDN servers at the MAN or AN. As an alternative solution on the traffic offloading of the FIFA World Cup, the edge computing [10] can support the closest contents caching at the edge servers to satisfy the requirements of users. Nevertheless, at the era of mobile Internet, the ability of edge computing cannot be handled with the billions of mobile devices. At the right time, the Mobile Edge Computing (MEC) [11–13] has been regarded as the relatively appropriate alternative solution.

Different from the centralized cloud computing mode, the computation resources and storage resources in MEC are deployed at the edge network (such as mobile base station, wireless hotspot, and edge router) in the distributed way. On this basis, the computation tasks on the FIFA World Cup traffic can be offloaded to the mobile edge servers for running, which greatly reduces the communication overhead and the network delay of application programs. At the same time, the pressure faced by Internet content providers as well as core networks can be relieved effectively. In fact, the network performance improvement in MEC strictly depends on the tasks offloading decision [14]. Furthermore, the decision problem usually involves some necessary factors, such as network bandwidth, timing sequence of application program, and dependence between tasks. As a result, the mobile application programs can usually be built as a Directed Acyclic Graph (DAG) model [15] to realize the fine-grained traffic scheduling and enable the parallel processing for the multiple tasks. However, the task scheduling based on DAG belongs to the NP-hard problem [16], which indicates that those heuristic and approximate traffic offloading strategies cannot satisfy the requirements of users (especially for the real-time requirements) during the process of watching the FIFA World Cup. Therefore, it is extremely urgent to find a stable and powerful method to solve such problem.

To the best of our knowledge, the Deep Reinforcement Learning (DRL) [17, 18] has attracted much attention from the global researchers in the field of Artificial Intelligence (AI), which integrates the advantages of Reinforcement Learning (RL) [19] and Deep Neural Networks (DNN) [20] and enables obtaining the optimal decision by automatically learning the network environment based on the multiple interactions. To be specific, the DRL has the following benefits. (1) DRL can learn the optimal decision strategy in the model-free way, and the whole process has no need for the environment modelling, reflecting the great flexibility. (2) DRL has the strong presentation ability and generalization ability of supporting the huge state space in terms of the DAG-based traffic scheduling problem. (3) DRL is a global optimization strategy and it has the large probability of obtaining the optimal solution. Although there have been some proposals on using DRL to address the traffic offloading problem in MEC, they cannot relieve the network pressure well and cannot be used for the traffic offloading in the FIFA World Cup directly. With the above considerations, this paper plans to propose a novel DRL-based MEC

traffic offloading (NDMT) strategy for the 2022 FIFA World Cup, and the major contributions are summarized as follows:

- (i) We compute the priorities of tasks and transfer DAG into a sequence of tasks according to the computed priorities, in which the scheduling process with respect to the tasks sequence is regarded as the Markov Decision Process (MDP)
- (ii) A DNN model based on the Sequence to Sequence (S2S) form is designed and used to fit the scheduling strategy of MDP, where the DAG is converted into the sequence of tasks to be input into the DNN
- (iii) The proximal policy optimization (PPO) method is used to train the DNN in the DRL, for obtaining the high stability and reliability

The rest paper is organized as follows. Section 2 reviews and compares the related work. Section 3 introduces the system framework of NDMT. Section 4 gives the problem description on traffic offloading in MEC. Section 5 presents the construction method of MDP. The traffic scheduling strategy based on DRL is proposed in Section 6. Section 7 reports the experimental results. Section 8 concludes this paper and gives the future research direction.

2. Related Work

In this section, we review the researches on the task (traffic) offloading of MEC in last three years (2018–2020) from two aspects, i.e., the heuristic methods and the DRL-based methods.

2.1. Heuristic Methods. There have been a lot of heuristic traffic offloading strategies in MEC. For example, the authors in [21] studied the scenario where multiple mobiles uploaded tasks to a MEC server in a single cell by allocating the limited server resources and wireless channels between mobiles devices. In particular, the authors formulated the optimization problem for the saved energy on the mobile devices with the tasks being dividable and utilized the selection maximum saved energy first algorithm to realize the solving process. In [22], the authors investigated an energy-efficient joint computation offloading, load balancing, and transmission power control problem and further proposed a heuristic algorithm to obtain the good traffic offloading while guaranteeing load balancing among the multiple servers. In [23], a distributed computation offloading and resource allocation optimization scheme in the heterogeneous networks with MEC was proposed, in which an optimization problem was formulated to provide the optimal computation offloading strategy. In [24], the traffic offloading strategy based on Software-Defined Networking (SDN) in the ultra-dense network was devised to minimize the delay while saving the battery life of mobile devices. It transformed this optimization problem into task placement subproblem and resource allocation subproblem, which could reduce 20% of the task duration with 30% energy saving. In [25], a MEC-enabled multicell wireless network

was considered where each base station is equipped with a MEC server that assisted mobile users in executing computation-intensive tasks via task offloading. It formulated the involved problem as a mixed integer nonlinear program, including the task offloading decision, uplink transmission power of mobile users, and resource allocation computation at the MEC servers. Furthermore, the authors in [26] jointly decided on the computing resource allocation for the hosted applications and designed a novel thoughtful decomposition based on the technique of the logic-based benders decomposition, with the heterogeneity in the requirements of the offloaded tasks (different computing requirements, latency, and so on) and limited MEC capabilities consideration. In [27], the authors studied the trade-off between task execution time and energy consumption at end-users under varying wireless channel conditions for soft real-time applications and involved tasks. It proposed a genetic algorithm with constrained mutation for optimal job partitioning and introduced an edge-proposing deferred acceptance algorithm to solve the preference based matching game. In [28], a task offloading algorithm that utilized the cache function of edge server was proposed. When the task with the same cached type of contents was uploaded to the edge server, the preset evaluation parameters took some factors into account to calculate the optimal processing position for the task. In [29], each base station was integrated with a MEC server in terms of the executing intensive computation task, and an iterative algorithm was proposed to solve the optimization problem in a single mobile user MEC system. In [30], an agent was introduced into the offloading of computation tasks, and a novel framework of agent-enabled task offloading in the unmanned aerial vehicle aided MEC was proposed to help the users obtain the good Quality of Experience (QoE). In [31], the authors addressed the problem of coordinating the offloading decisions of wireless devices that periodically generated computationally intensive tasks due to the various delay sensitive applications. In addition, they also developed a game theory based model and proposed a polynomial complexity algorithm for computing an equilibrium. In [32], the authors considered a system where most mobile devices migrated the duplicate computation tasks to the edge servers and shared the requested contents for computation tasks. Therein, an efficient Lyapunov online algorithm that could perform joint task offloading and dynamic data caching strategies for computation tasks or contents was proposed to reduce the overall latency of all mobile devices. Although these traffic offloading strategies had good performance, they could not decrease the network pressure well and could not be used for the traffic offloading in the FIFA World Cup directly due to the special traffic features, i.e., the periodicity, abruptness, and explosivity.

2.2. DRL-Based Methods. There have also been some DRL-based traffic offloading strategies in MEC. For example, in [33], a multiuser MEC system was considered, where the multiple users could perform computation offloading via wireless channels to a MEC server. Particularly, the RL-

based optimization framework was introduced to tackle the resource allocation in wireless MEC. In [34], a deep-Q network based task offloading and resource allocation algorithm for the MEC was proposed, where each mobile terminal had the multiple tasks offloaded to the edge server. It also designed a joint task offloading decision and bandwidth allocation optimization to minimize the overall offloading cost in terms of energy cost, computation cost, and delay cost. In [35], the DRL was first proposed to solve the offloading problem of multiple service nodes for the cluster and multiple dependencies for mobile tasks in the large-scale heterogeneous MEC. In particular, it used the long short term memory network layer and the candidate network set to improve the deep-Q network algorithm in combination with the actual environment of the MEC. In [36], the authors considered MEC for a representative mobile user in a sliced Radio Access Network (RAN), where the multiple base stations were available to be selected for computation offloading. Meanwhile, a double DQN-based strategic computation offloading algorithm to learn the optimal policy without knowing a priori knowledge of network dynamics was proposed to break the curse of high dimensionality in state space. In [37], an intelligent offloading system for vehicular edge computing by leveraging DRL was constructed, in which both communication and computation states were modelled by the finite Markov chains. In [38], the authors investigated the problem of delay sensitive task scheduling and resource management on the server side in multiuser MEC scenario, where a new online algorithm based on DRL was devised to reduce average slowdown and average timeout period of tasks in the queue. In [39], the computing aware scheduling strategy in MEC was proposed, in which a support vector machine based multiclass classifier was adopted. Although these DRL-based strategies also showed good effect on the traffic scheduling, they always had some limitations to be improved, such as scheduling time, bandwidth utilization, and QoE of user. This motivates the study of this paper. Besides, this paper also gives a special application scenario, i.e., the 2022 FIFA World Cup, which can provide a significant reference.

3. System Framework

This section introduces the system framework of NDMT, including MEC-based traffic offloading architecture and DRL-based workflow for MEC traffic offloading. In addition, the abbreviations frequently used in this paper are listed in Table 1.

3.1. MEC-Based Traffic Offloading Architecture. In this paper, we present a MEC-based traffic offloading architecture, as shown in Figure 1, where the MEC servers are deployed at the edge network (e.g., AN) to provide the handy and low-latency computation services for the mobile users. In particular, MEC allocates the specialized hardware and software resources for each user and separates such resources by using the virtualization technology, so that the quality of service and the privacy of user can be guaranteed effectively.

TABLE 1: Abbreviations in alphabetical order.

| Abbreviation | Full name |
|--------------|--|
| AI | Artificial Intelligence |
| AN | Access Network |
| ByLu | Baseline by Lu et al. |
| ByCh | Baseline by Chen et al. |
| CDN | Content Delivery Networks |
| CPU | Central Processing Unit |
| DAG | Directed Acyclic Graph |
| DNN | Deep Neural Networks |
| DRL | Deep Reinforcement Learning |
| DTU | Data Transmission Unit |
| MAN | Metropolitan Area Network |
| MDP | Markov Decision Process |
| MEC | Mobile Edge Computing |
| NDMT | Novel DRL-based MEC traffic offloading |
| PPO | Proximal policy optimization |
| QoE | Quality of Experience |
| RL | Reinforcement Learning |
| RNN | Recurrent Neural Network |
| S2S | Sequence to Sequence |

For the side of user with the mobile device, the computation tasks generated from the mobile applications (e.g., the FIFA World Cup Traffic) can be performed at the local mobile device's Central Processing Unit (CPU) directly or can be sent to the MEC server via the Data Transmission Unit (DTU) and be performed by the corresponding service instance (i.e., the remote traffic offloading). Meanwhile, the traffic offloading module is used to make the scheduling decision for all tasks in the mobile device, including two functions, i.e., the execution way and the scheduling order.

An application program usually includes the multiple computation tasks that have the dependence among them, which can help realize the fine-grained traffic scheduling and enable the parallel processing for the multiple tasks. In this paper, we model the mobile application program related to the FIFA World Cup Traffic as a DAG, denoted by $G = (T, L)$, where T is the set of tasks and L is the set of links constructed by the tasks. Here, the arbitrary task is denoted by t_i ; the arbitrary link is denoted by $l(t_i, t_j)$. Particularly, for $l(t_i, t_j)$, t_i is the precursor task of t_j while t_j is the successor task of t_i ; that is to say, the performing of t_j relies on t_i . In DAG, the task without any precursor task is called the entry task, whilst the task without any successor task is called exit task. In addition, it allows an application program to have the multiple entry tasks and the multiple exit tasks in case of the parallel processing. For t_i , it has three attributes, i.e., the input volume of traffic, the number of CPU cycles, and the output volume of traffic, denoted by Iv_i , Cy_i , and Ov_i respectively, and the corresponding values can be obtained by the program analyzer, just like in [40], which reflect the required transmission cost and computation cost.

As depicted in Figure 1, the performing of task has two ways, i.e., offloading performing and local performing. If t_i is scheduled to the remote edge server for performing, the whole process consists of three phases, i.e., task sending, edge performing, and result returning. At the first phase, the volume of traffic Iv_i is transmitted to the remote edge server.

Let Rul denote the transmission rate of uplink, and the required transmission time Tul_i of t_i is defined as follows:

$$Tul_i = \frac{Iv_i}{Rul}. \quad (1)$$

Then, at the edge performing phase, the Cy_i CPU cycles are performed at the corresponding server instance in the MEC server. Let Fv denote the virtual clock frequency of server instance for t_i , and the required performing time Tep_i of t_i is defined as follows:

$$Tep_i = \frac{Cy_i}{Fv}. \quad (2)$$

Similar to the first phase, the returning time at the result returning phase Tdl_i is defined as follows:

$$Tdl_i = \frac{Ov_i}{Rdl}, \quad (3)$$

where Rdl is the transmission rate of downlink.

Furthermore, let Tof_i denote the total time cost in case of performing t_i via the traffic offloading way, and it concludes three parts of time costs, i.e.,

$$Tof_i = Tul_i + Tep_i + Tdl_i. \quad (4)$$

If t_i is performed at the local mobile device, it is unnecessary to upload and download the data to which t_i corresponds; this is to say, the total time cost only depends on the local computation overhead with respect to the consumption of CPU resources. Let Tlo_i denote the total time cost in case of performing t_i at the local mobile device, and we have

$$Tlo_i = \frac{Cy_i}{Fl}, \quad (5)$$

where Fl is the local CPU's clock frequency.

3.2. DRL-Based Workflow for MEC Traffic Offloading. In this section, we describe the DRL-based workflow for MEC traffic offloading, as shown in Figure 2. The whole workflow consists of four main modules, i.e., MEC problem description, MDP construction, DNN-based strategy fitting, and PPO-based reinforcement training. Among them, the first module is to present the involved problem on traffic offloading, including the local scheduling and the offloading scheduling. The second module is to transfer the scheduling process with the tasks sequence as the MDP, where the priority of task is computed and regarded as the transferred attribute. In the third module, the S2S-based DNN model is used to fit the scheduling strategy. In the last module, the PPO-based DRL method is adopted to train the DNN because the PPO has the good stability and reliability.

Furthermore, during the whole process of workflow, all processing units (including CPU, DTU, virtual CPU, and virtual DTU) only perform and send one task, which indicates that the multiple tasks preemption phenomenon is not allowed. In addition, the DAG-based task scheduling in MEC satisfies the following two features. (1) Given the bandwidth limitation of edge network, the transmission rate

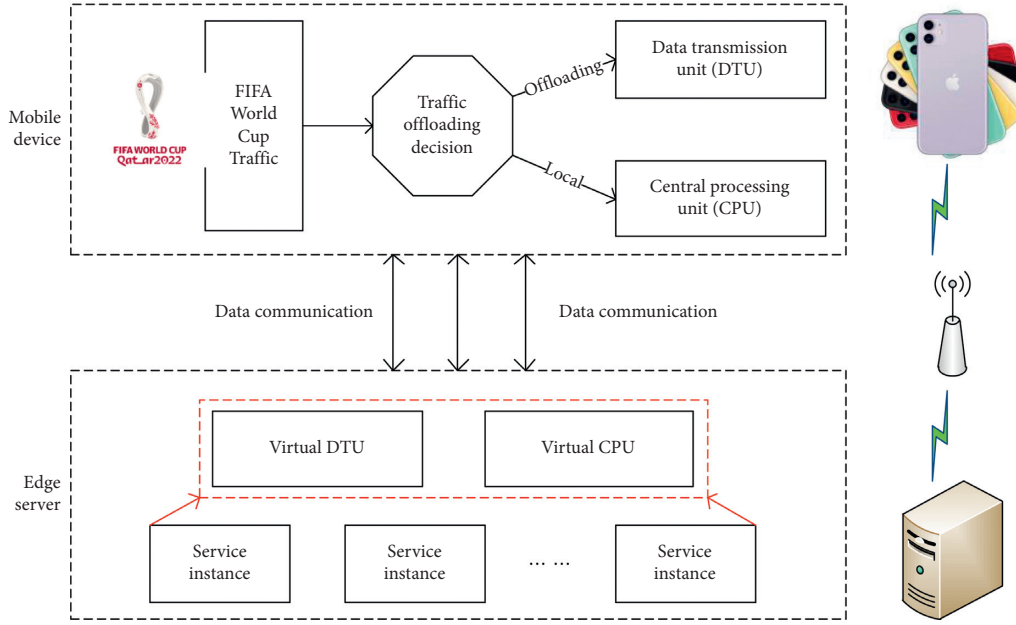


FIGURE 1: MEC-based traffic offloading architecture.

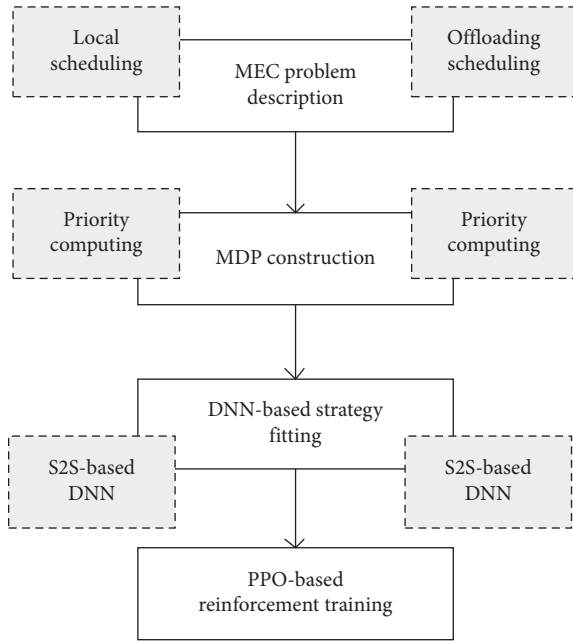


FIGURE 2: DRL-based workflow for MEC traffic offloading.

between mobile device and edge server (uplink or downlink) keeps the fixed value. (2) The whole scheduling can be finished until the computation result is returned to the mobile device.

4. Problem Description

At first, we define four timestamps with respect to the completed time, i.e., task sending, edge performing, result returning, and local performing, denoted by Ts_{os_i} , Ts_{op_i} , Ts_{or_i} , and Ts_{lp_i} , respectively. If t_i is performed at the local device, we have $Ts_{os_i} = Ts_{op_i} = Ts_{or_i} = 0$; otherwise,

$Ts_{lp_i} = 0$. In particular, before t_i is scheduled, it is required that all precursor tasks of t_i have to be performed in advance.

Consider the condition where t_i is performed at the local mobile device, let RTs_{lp_i} be the ready timestamp regarding scheduling t_i , and we have

$$RTs_{lp_i} = \max_{t_j \in \text{pre}_i} \{Ts_{lp_j}, Ts_{or_j}\}, \quad (6)$$

which indicates that RTs_{lp_i} is the earliest timestamp in terms of such condition where all precursor tasks are completed, pre_i is the set of precursor tasks with respect to t_i , and t_j is one precursor task of t_i . In particular, it perhaps needs the queueing for each task before being performed at CPU; thus, the starting timestamp may be not equal to the ready timestamp. Let STs_{lp_i} denote the starting timestamp regarding scheduling t_i , and we have $STs_{lp_i} \geq RTs_{lp_i}$, satisfying

$$Ts_{lp_i} = STs_{lp_i} + Tlo_i. \quad (7)$$

Consider the condition where t_i is scheduled to the remote edge server for performing, let RTs_{os_i} denote the ready timestamp regarding sending t_i , and we have

$$RTs_{os_i} = \max_{t_j \in \text{pre}_i} \{Ts_{os_j}, Ts_{lp_j}\}, \quad (8)$$

where all precursor tasks of t_i are performed at the local mobile device or the remote server. Similarly, let STs_{os_i} denote the starting timestamp when t_i is sent, and we have $STs_{os_i} \geq RTs_{os_i}$ and $Ts_{os_i} = STs_{os_i} + Tul_i$.

Then, let RTs_{op_i} denote the ready timestamp regarding performing t_i via the service instance, and we have

$$RTs_{op_i} = \max \left\{ Ts_{os_i}, \max_{t_j \in \text{pre}_i} Ts_{op_j} \right\}. \quad (9)$$

Among them, RTs_{op_i} depends on the fact that the input traffic of t_i is completed; i.e., the transmission of Iv_i is

finished. Similarly, the starting performing timestamp of t_i at the virtual CPU relies on the corresponding queuing situation, and we have $ST_{sop_i} \geq RT_{sop_i}$ and $T_{sop_i} = ST_{sop_i} + T_{ep_i}$, where ST_{sop_i} is the starting timestamp when t_i is performed via the service instance.

Finally, when the performing of v_i is finished, v_i gets into the ready status of result returning. Let RT_{sor_i} denote the ready timestamp regarding returning the computed result of t_i , and we have $RT_{sor_i} = T_{sop_i}$. Furthermore, let ST_{sor_i} denote the starting timestamp when the computed result of t_i is returned, and we have $ST_{sor_i} \geq RT_{sor_i}$ and $T_{sor_i} = ST_{sor_i} + T_{dl_i}$.

Let $TT_{Total}(G)$ denote the required time to perform an application program (i.e., all involved tasks have completed the results returning), and we have

$$\begin{aligned} TT_{Total}(G) &= \max_{t_i \in \text{exit}(G)} \{\max\{T_{sor_i}, T_{slp_i}\}\} \\ &= \max_{t_i \in \text{exit}(G)} \{\max\{T_{of_i}, T_{lo_i}\}\}, \end{aligned} \quad (10)$$

where $\text{exit}(G)$ is the combinatorial set of exit tasks in G . This paper considers the delay sensitive FIFA World Cup traffic; therefore, the main purpose is to maximize the QoE while minimizing $TT_{Total}(G)$ based on different scheduling strategies. Particularly, during the process of scheduling, each task's execution way and scheduling order should be determined. In addition, we emphasize that the scheduling order of task at the local mobile device and that at the service instance keep consistent.

5. MDP Construction

5.1. Priority Computation. For all traffic scheduling strategies, computing the priority for each task is the indispensable operation. Furthermore, based on the computed priority, the scheduling order can be determined [41]. For the arbitrary t_i in the DAG, its computation cost with respect to the time can be obtained by (4) and expressed by T_{of_i} . Based on T_{of_i} , the priority of t_i is defined as follows:

$$Pr_i = \max_{t_j \in \text{suc}_i} Pr_j + T_{of_i}, \quad (11)$$

where Pr_i is the priority of t_i and suc_i is the set of successor tasks with respect to t_i . It is obvious that the (11) shows the recursive form. If t_i is the exit task, we have

$$Pr_i = T_{of_i}, \quad t_i \in \text{exit}(G). \quad (12)$$

Then, the DAG is depth-firstly traversed with starting from the exit task, and all tasks' priorities can be obtained by (11) and (12). These tasks are arranged according to the corresponding priorities in the descending order, and the scheduling sequence of tasks can be defined as follows:

$$Q = (t'_1, t'_2, \dots, t'_n), \quad (13)$$

where n is the number of tasks (or nodes in DAG). In particular, Q is the special topological sorting result on G , and the original dependence among tasks can be guaranteed according to the sequential scheduling in Q .

5.2. Construction Method. The sequential scheduling decision process for all sorted tasks in Q can be modelled as one MDP, denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{D}_0, \mathcal{R}, \lambda)$, where \mathcal{S} , \mathcal{A} , \mathcal{P} , \mathcal{D}_0 , \mathcal{R} , and λ are the state space, action space, state-transition matrix, the probability distribution of initial state, reward function, and discount factor, respectively.

Let k denote the scheduled number of tasks in Q , and the current state space can be expressed as follows:

$$\begin{aligned} \mathcal{S} &= \{s | st = \mathcal{r}(G, k, \mathcal{A}^k)\}, \\ \mathcal{A}^k &= (a_1, a_2, \dots, a_k), \end{aligned} \quad (14)$$

where \mathcal{A}^k is used to describe the scheduling condition (i.e., state space) on the first k tasks in Q and a_k is used to record the execution way of task: $a_k = 1$ means that k -th task in Q is performed in the offloading way; otherwise, it is performed at the local mobile device.

Furthermore, let $Q_{1 \rightarrow i}$ denote the scheduled sequence with respect to the first i tasks in Q , and we can construct a scheduled subgraph of G , denoted by $G_{1 \rightarrow i} = (T', L')$; here $T' \subseteq T$, $L' \subseteq L$, $G_{1 \rightarrow n} = G$ and $G_{1:0} = \Phi$. Under such condition, in order to minimize the scheduling time, we give a reward function for the current task, defined as follows:

$$r_i = \mathcal{R}(s_i, a_i) = TT_{Total}(G_{1 \rightarrow i}) - TT_{Total}(G_{1 \rightarrow i+1}), \quad (15)$$

which refers to the time difference between a_i performed before and that after in terms of s_i .

Moreover, the traffic offloading decision module in Figure 1 can be defined as a conditional probability function, denoted by $\theta(a_i | ts_i)$. From the initial state s_0 , upon the traffic offloading decision module completes an action, the system enters a new state and further gets the corresponding reward based on (15) until the last task in Q is completed. According to the above statements, the whole task scheduling process based on MDP is described as follows:

$$\text{MDP} := (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{n-1}), \quad (16)$$

where s_{n-1} is the termination state to mean that all tasks have been completed. Then, the accumulated reward with the discount factor consideration is defined as follows:

$$\begin{aligned} R &= \sum_{i=0}^{n-1} \lambda * r_i = \lambda * (TT_{Total}(G_{1 \rightarrow i}) - TT_{Total}(G_{1 \rightarrow i+1})) \\ &= -\lambda * TT_{Total}(G), \end{aligned} \quad (17)$$

which indicates that the maximization of the accumulated reward with the discount factor consideration is consistent with the minimization of the total scheduling time.

6. DRL-Based Traffic Scheduling Strategy

In fact, the DAG has the feature of diversity and the involved state space is infinitely great; thus, it is impossible to obtain the corresponding state-transition matrix in advance. Given this, this paper uses the DRL to find the optimal scheduling strategy for the traffic scheduling decision module.

6.1. DNN-Based Strategy Fitting. We employ DNN to fit $\theta_x(a_i|ts_i)$, where x is the set of parameters related to DNN. As we know, the input of DNN is s_i which is related to G but G cannot be input to the DNN directly due to the restriction of data features; therefore, the DAG is converted into the sequence of tasks (just like Q) to be input into the DNN. For such sequence, it consists of the following three vectors: (1) time vector including Tu_i , Te_i , Td_i , and Tl_i , (2) precursor vector including all precursor indexes, and (3) successor vector including all successor indexes. Among them, the size of precursor/successor vector is set as a fixed value, denoted by sz . If the number of precursor/successor tasks is smaller or equal to sz , the corresponding locations are filled by -1 ; otherwise, the extra parts are ignored directly.

The output of DNN is the probability distribution of executable actions based on the current task state. In fact, for the current task, its scheduling strategy decision action (a_i) has the direct influence on the next task's state (s_{i+1}); therefore, this paper leverages the S2S-based DNN structure model, including encoder and decoder, as shown in Figure 3. In particular, both encoder and decoder are realized by the Recurrent Neural Network (RNN). Meanwhile, the encoder receives such input sequence in turn and finally outputs the hidden layer(s) as the features of DAG. The decoder initializes its own hidden layer(s) by using the output result from the encoder. In addition, the decoder sequentially inputs the scheduling actions (\mathcal{A}^i) and then outputs the corresponding $\theta_x(a_i|ts_i)$. The above process can be still performed until s_{n-1} is completed.

6.2. PPO-Based Reinforcement Training. In order to obtain the optimal traffic scheduling decision, the training objective of DRL can be defined as follows:

$$\max_x \text{Objective}(x) = \max_x \left(\theta_x^*(\mathcal{A}^{n-1}|tG) * \sum_{i=0}^{n-1} r_i \right), \quad (18)$$

$$\theta_x^*(\mathcal{A}^{n-1}|tG) = \prod_{i=0}^{n-1} \theta_x(a_i|s_i),$$

which indicates that the whole traffic scheduling decision depends on all scheduling decisions of tasks. In this paper, we use the PPO [42] to train such traffic scheduling decisions, which can accelerate the efficient convergence while guaranteeing the scalability and reliability.

Since the scheduled DAG has the feature of diversity and the involved state space is infinitely great, it is impossible to search all DAGs. With such consideration, we can continuously collect the scheduled DGAs after strategy deployment to construct the training set related to the DAG. Then, based on the training set, we also can train the traffic scheduling strategies. In addition, in order to obtain the better effect on the convergence, we scale back the obtained reward by each scheduling decision during the process of training; that is, r_i always keeps in $[0, 1]$.

7. Performance Evaluation

7.1. Setup. The proposed NDMT is implemented by the C++ programming, and the involved simulation parameters are

set in Table 2. Among them, the network scale is dynamically changing from $n = 20$ to $n = 60$ according to the pattern of Figure 4 due to the fact that it accords with the network deployment feature to offload the FIFA World Cup traffic, where the step length is 10. Particularly, for each scale, there are 2500 DAGs being used for training the DRL and there are 300 DAGs being used for testing the DRL; in other words, there are $5 * 2500 = 12500$ and $5 * 300 = 1500$ DAGs being used to train and test the DRL, respectively. The TensorFlow [43] is used to realize the DRL, where both encoder and decoder have 256 hidden neurons. In addition, the method of layer normalization [44] is used to improve the training efficiency.

Furthermore, the researches from [35, 36], respectively, are used as two baselines, because they are the latest research representatives on the DRL-based traffic offloading in MEC. Therein, [35] is proposed by Lu et al. while [36] is proposed by Chen et al. and in this paper they are abbreviated to ByLu and ByCh, respectively. In terms of experiments, the DRL convergence for traffic offloading is analyzed firstly. Then, three metrics, i.e., scheduling time, bandwidth utilization, and QoE are used to measure the proposed NDMT's performance.

7.2. Convergence Analysis. This section verifies the convergence of DRL based on 12500 DAGs, in which the average accumulated reward is considered as the evaluation metric. For each training, we record the corresponding training result. When the training within one period is finished, we input 300 testing DAGs into the S2S-based DNN and the related traffic offloading strategy is obtained. Then, the obtained traffic offloading strategy is simulated and the average accumulated reward can be computed. The relationship between the average accumulated reward and the training period is shown in Figure 5. We can observe that the whole training process includes three stages, i.e., rapid increasing stage, stable increasing stage, and stationary stage, and the stationary stage is reached with the need of 190 training periods. It indicates that the proposed NDMT can converge to the optimal state space and further obtain the optimal traffic scheduling decision.

7.3. Scheduling Time. The scheduling time is defined as the time difference between the time point when the first task is sent and that when the computation result of the last task is obtained by the local mobile device. The average scheduling times of NDMT, ByLu, and ByCh are shown in Figure 6.

We can observe that NDMT always has the smallest average scheduling time, followed by ByLu and ByCh, and there are two main reasons. On the one hand, NDMT considers the process of scheduling as the MDP, which can improve the processing speed for each task in the S2S-DNN structure model. On the other hand, NDMT adopts the layer normalization method to increase the training efficiency, which can further save the training time and strive for the scheduling time as small as possible. For two baselines, ByCh does not consider a priori knowledge of network dynamics to learn the optimal policy and the state space is the high

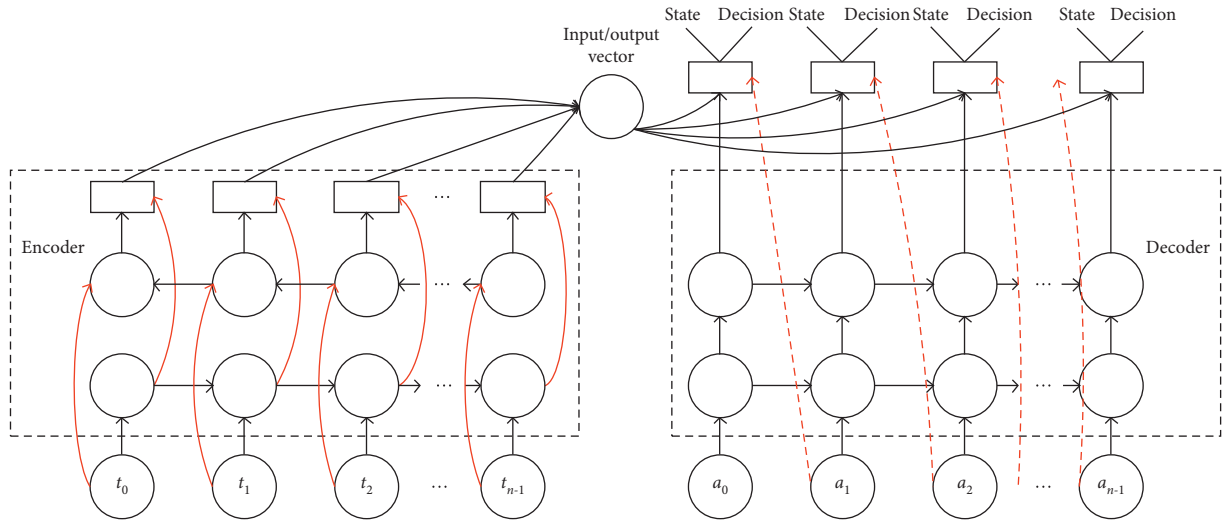


FIGURE 3: S2S-based DNN model.

TABLE 2: Simulation settings.

| Parameter | Setting |
|-----------------------------------|-------------------------------|
| F _l | 2×10^9 cycles/s |
| F _v | 1.5×10^{10} cycles/s |
| T _{ul_i} | 10 Mbps |
| T _{dl_i} | 10 Mbps |
| λ | 0.85 |
| sz | 14 |
| n | 20, 30, 40, 50, 60 |
| The number of simulations | 100 |
| The learning rate in DRL | 2×10^{-4} |
| The training number of DAGs for n | 2500 |
| The testing number of DAGs for n | 300 |

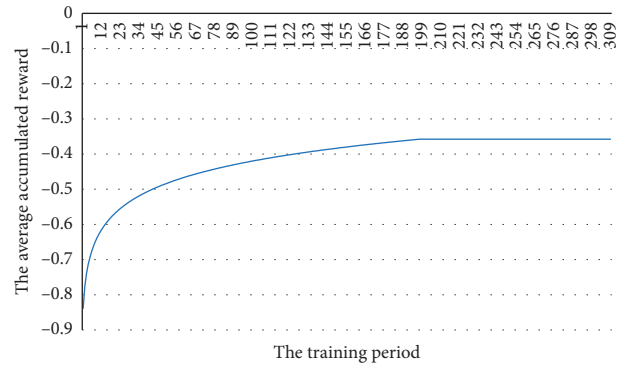


FIGURE 5: The convergence analysis result.

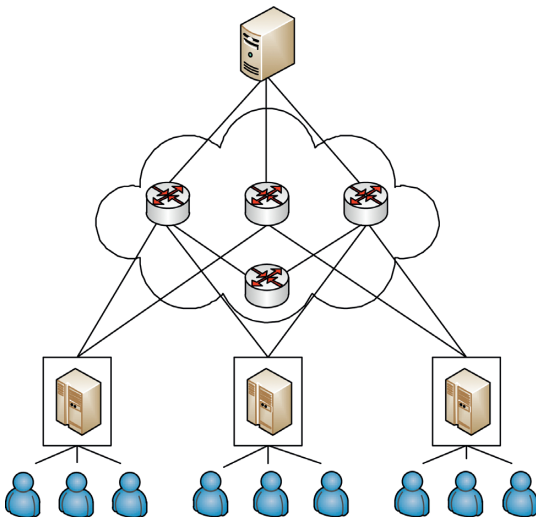


FIGURE 4: The network topology pattern used for simulation.

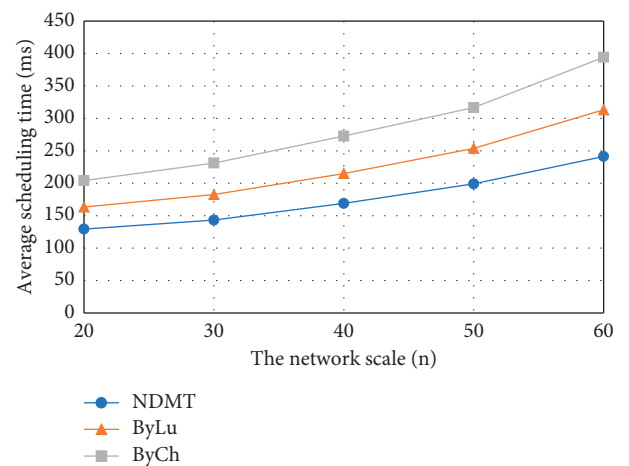


FIGURE 6: The average scheduling times among NDMT, ByLu, and ByCh.

dimensional; thus, it has larger average scheduling time than ByLu. In addition, we can also observe that the average scheduling time becomes larger and larger with the increasing of network scale, which results from two aspects.

On the one hand, it needs much more time to train the DAGs; on the other hand, it needs much time to compute more tasks.

7.4. Bandwidth Utilization. The bandwidth utilization is defined as the ratio of the used bandwidth and the total network bandwidth. The average bandwidth utilizations of NDMT, ByLu, and ByCh are shown in Figure 7.

We can observe that NDMT always has the highest average bandwidth utilization, followed by ByLu and ByCh, which is regarded as an important benefit of NDMT. Regarding this, there are no the concrete reasons. Furthermore, we can observe that the average bandwidth utilization basically remains unchanged, i.e., having the strong stability for different network scales; this is because NDMT always can converge to the optimal solution (see Figure 5). It suggests that the proposed NDMT has the considerable reference value for the following 2022 FIFA World Cup. However, the average bandwidth utilizations of ByLu and ByCh become lower and lower with the increasing of network scale because the corresponding convergences are nondeterminate.

7.5. QoE of User. We use the watching fluency to measure the QoE of user, and the watching fluency is defined as the number of network lags per 10 mins. The average numbers of network lags of NDMT, ByLu and ByCh are shown in Figure 8.

We can observe that NDMT always has the smallest average number of network lags, followed by ByCh and ByLu, which further indicates that the user has the best watching experience in NDMT, because NDMT has the highest bandwidth utilization and the smallest response time. For ByLu and ByCh, the latter deploys the large number of base stations to offload traffic and thus the required response time is relatively smaller than the former; as a result, ByCh has better QoE of user than ByLu. In addition, we can also observe that NDMT has the most stable QoE of user but the two baselines do not have, and similar reasons can be found in the above section. Moreover, the experimental results suggest that the users can enjoy the best experience when watching the following 2022 FIFA World Cup by the personal mobile devices under the environment of MEC.

8. Conclusions

In this paper, we investigate the MEC traffic offloading strategy based on DRL. At first, we introduce the proposed system framework, including MEC-based traffic offloading architecture and DRL-based workflow for MEC traffic offloading. Then, we give the problem description, i.e., minimizing the total performing time for one application program. For the concrete scheduling strategy, it includes three parts. At the first part, we compute the priorities of tasks and transfer DAG into a sequence of tasks according to the computed priorities, and the scheduling process with respect to the tasks sequence is regarded as the MDP. At the second part, one S2S-based DNN model is used to fit the scheduling strategy. At the third part, the PPO method is employed to train the DNN. We do the simulation experiments based on the TensorFlow including the convergence

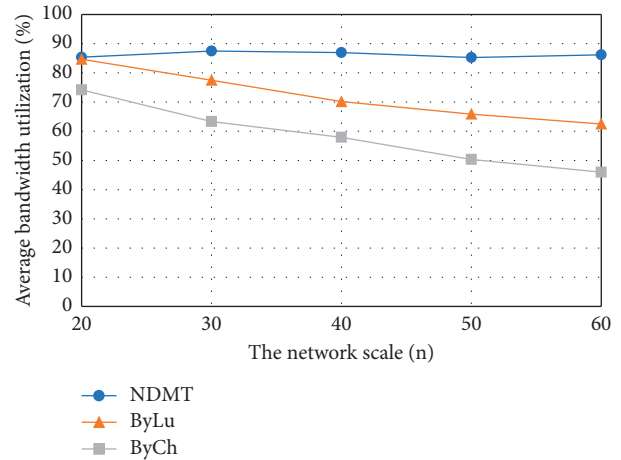


FIGURE 7: The average bandwidth utilizations among NDMT, ByLu, and ByCh.

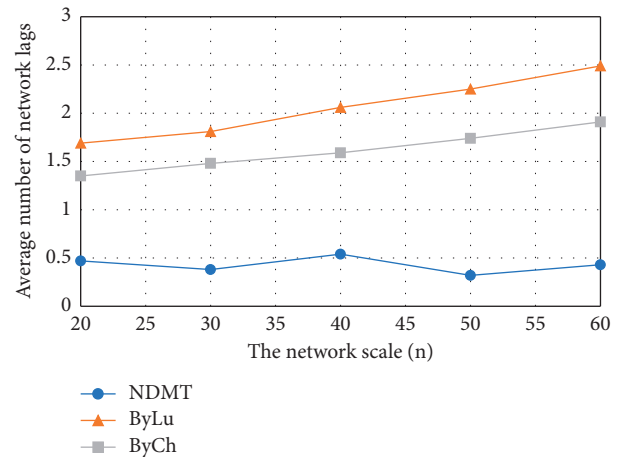


FIGURE 8: The average numbers of network lags among NDMT, ByLu, and ByCh.

analysis and the performance comparison. Meanwhile, the scheduling time, bandwidth utilization, and QoE of user are considered three performance evaluation metrics, and we observe that the proposed NDMT outperforms two baselines. Based on the nice experiment results, we think that the proposed NDMT can be regarded as a feasible and efficient reference for the following 2022 FIFA World Cup held in Qatar. In the future, we plan to improve NDMT from the following two aspects. At first, more datasets are collected and used to train the DNN; then, NDMT is deployed at a real network environment to further verify its performance.

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 no conflicts of interest.

References

- [1] Z. Yan, P. Zhang, A. V. Vasilakos et al., "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, no. 3, pp. 120–134, 2014.
- [2] L. Quijano-Sanchez, I. Cantador, M. E. Cortes-Cediel et al., "Recommender systems for smart cities," *Information Systems*, vol. 92, pp. 1–22, 2020.
- [3] Cisco Annual Internet Report (2018–2023) White Paper, <http://www.cisco.com/c/en/us/solutions>.
- [4] F. Pacheco, E. Exposito, and M. Gineste, "A framework to classify heterogeneous Internet traffic with Machine Learning and Deep Learning techniques for satellite communications," *Computer Networks*, vol. 173, pp. 1–21, 2020.
- [5] Y. Koike and A. Inoue, "High-speed graded-index plastic optical fibers and their simple interconnects for 4K/8K video transmission," *Journal of Lightwave Technology*, vol. 34, no. 6, pp. 1551–1555, 2016.
- [6] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, and A. El Saddik, "Edge caching and computing in 5G for mobile AR/VR and tactile Internet," *IEEE MultiMedia*, vol. 26, no. 1, pp. 21–30, 2019.
- [7] Y. Liu, S. Liu, Y. Wang, and H. Zhao, "Video coding and processing: a survey," *Neurocomputing*, vol. 408, pp. 331–344, 2020.
- [8] S. Cui, M. R. Asghar, and G. Russello, "Multi-CDN: towards privacy in content delivery networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 984–999, 2020.
- [9] N. Anjum, D. Karamshuk, M. Shikh-Bahaei, and N. Sastry, "Survey on peer-assisted content delivery networks," *Computer Networks*, vol. 116, pp. 79–95, 2017.
- [10] H. Lin, S. Zeadally, Z. Chen et al., "A survey on computation offloading modeling for edge computing," *Journal of Network and Computer Applications*, vol. 169, pp. 1–13, 2020.
- [11] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective," *Computer Networks*, vol. 182, pp. 1–15, 2020.
- [12] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [13] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [14] M. Zamzam, T. Elshabrawy, and M. Ashour, "Resource management using machine learning in mobile edge computing: a survey," in *Proceedings of the 2019 International Conference on Intelligent Computing and Information Systems*, pp. 112–117, Cairo, Egypt, December 2019.
- [15] G. Madraki and R. P. Judd, "Recalculating the length of the longest path in perturbed directed acyclic Graph," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1560–1565, 2019.
- [16] J. D. Ullman, "NP-complete scheduling problems," *Journal of Computer and System Sciences*, vol. 10, no. 3, pp. 384–393, 1975.
- [17] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [18] S. Cheng, L. Ma, H. Lu et al., "Evolutionary computation for solving search-based data analytics problems," *Artificial Intelligence Review*, 2020, in press.
- [19] M. M. Drugan, "Reinforcement learning versus evolutionary computation: a survey on hybrid algorithms," *Swarm and Evolutionary Computation*, vol. 44, pp. 228–246, 2019.
- [20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on Graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020, in press.
- [21] F. Wei, S. Chen, and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," *China Communications*, vol. 15, no. 11, pp. 149–157, 2018.
- [22] S. Li, D. Zhai, P. Du et al., "Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks," *Science China Information Sciences*, vol. 62, pp. 1–3, 2019.
- [23] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.
- [24] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.
- [25] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.
- [26] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 668–682, 2019.
- [27] X. Zhang and S. Debroy, "Adaptive task offloading over wireless in mobile edge computing," in *Proceedings of the ACM/IEEE Symposium on Edge Computing*, pp. 323–325, Washington, DC, USA, 2019.
- [28] J. Fang and W. Zeng, "Offloading strategy for edge computing tasks based on cache mechanism," in *Proceedings of the International Conference on Computing and Artificial Intelligence*, pp. 129–134, Cairo, Egypt, April 2020.
- [29] G. Wang, Q. Li, and X. Yu, "Computation task offloading for minimizing energy consumption with mobile edge computing," *Lecture Notes in Electrical Engineering*, vol. 571, pp. 2117–2123, 2020.
- [30] R. Wang, Y. Cao, A. Noor, T. A. Alamoudi, and R. Nour, "Agent-enabled task offloading in UAV-aided mobile edge computing," *Computer Communications*, vol. 149, pp. 324–331, 2020.
- [31] S. Josilo and G. Dan, "Computation offloading scheduling for periodic tasks in mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 667–680, 2020.
- [32] N. Zhang, S. Guo, Y. Dong et al., "Joint task offloading and data caching in mobile edge computing networks," *Computer Networks*, vol. 182, pp. 1–10, 2020.
- [33] J. Li, H. Gao, T. Lv et al., "Deep reinforcement learning-based task offloading and resource allocation for mobile edge computing," in *Proceedings of the International Conference on Machine Learning and Intelligent Communications*, pp. 33–42, Hangzhou, China, July 2018.
- [34] L. Huang, X. Feng, C. Zhang, L. Qian, and Y. Wu, "Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing," *Digital Communications and Networks*, vol. 5, no. 1, pp. 10–17, 2019.

- [35] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 102, pp. 847–861, 2020.
- [36] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4005–4018, 2019.
- [37] Z. Ning, P. Dong, and X. Wang, "Deep reinforcement learning for vehicular edge computing: an intelligent offloading system," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, pp. 1–24, 2019.
- [38] H. Meng, D. Chao, R. Huo et al., "Deep reinforcement learning based delay-sensitive task scheduling and resource management algorithm for multi-user mobile-edge computing systems," in *Proceedings of the International Conference on Mathematics and Artificial Intelligence*, pp. 66–70, Chengdu China, April 2019.
- [39] K. Wang, X. Yu, W. Lin et al., "Computing aware scheduling in mobile edge computing system," *Wireless Networks*, vol. 25, pp. 1–17, 2019.
- [40] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.
- [41] H. Topcuoglu, S. Hariri, and M. Min-You Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.
- [42] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, p. 1. in press, 2020.
- [43] TensorFlow, <https://tensorflow.google.cn>.
- [44] J. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, Cornell University, Ithaca, NY, USA, 2016.

Research Article

Performance Optimization Mechanism of Adolescent Physical Training Based on Reinforcement Learning and Markov Model

Mingze Wei  and Lei Yuan 

Jilin University, Changchun 130012, China

Correspondence should be addressed to Lei Yuan; yuanlei@jlu.edu.cn

Received 20 September 2020; Revised 3 October 2020; Accepted 20 October 2020; Published 2 November 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Mingze Wei and Lei Yuan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Upon the teenagers' failure to obtain the plenty of physical exercises at the growth and development stage, the related central nervous system is prone to degeneration and the physical fitness starts to decline gradually. In fact, through monitoring the exercise process real-time and quantifying the exercise data, the adolescent physical training can be effectively conducted. For such process, it involves two issues, i.e., the real-time data monitoring and data quantification evaluation. Therefore, this paper proposes a novel method based on Reinforcement Learning (RL) and Markov model to monitor and evaluate the physical training effect. Meanwhile, the RL is used to optimize the adaptive bit rate of surveillance video and help the real-time data monitoring; the Markov model is employed to evaluate the health condition on the physical training. Finally, we develop a real-time monitoring system on exercise data and compare with the state-of-the-art mechanisms based on this system platform. The experimental results indicate that the proposed performance optimization mechanism can be more efficient to conduct the physical training. Particularly the average evaluation deviation rate based on Markov model is controlled within 0.16%.

1. Introduction

The physical fitness of teenagers has attracted the global attention for a long time because it has the considerably important influence on the rise and fall of each country. In fact, the related central nervous system gradually starts to degenerate, and the physical fitness also will decline with it, when the teenagers fail to obtain the plenty of physical exercises at the growth and development stage. Furthermore, according to what one hears and sees as well as the reliable news, the sudden death events among high school and college students happen frequently, which makes people turn their attention to the fitness problem during the process of physical training [1]. To put it crudely, the adolescent physical training is of great importance while the exercise (including public/private, individual/population) must be done properly.

The researches show that the generated physical training data not only reflects the real trajectories of exercisers but also implies the abundant and valuable information related

to the whole exercise process [2]. To be specific, the information includes time, speed, acceleration, steps, and energy consumption. Among them, the energy consumption is an important metric and it can release two key signals, i.e., the amount of exercise and the exercise intensity. Given this, the amount of exercise and exercise intensity information can be obtained easily via monitoring the consumed energy, and then the physical training can be conducted and adjusted, which can be regarded as the healthy and reasonable exercise. In addition, according to the monitored energy consumption, the unforeseen circumstances due to the overtraining can be discovered in a timely manner, avoiding the worse tragedies as much as possible.

According to the above statements, we can observe that the video surveillance plays the nonnegligible role during the process of monitoring the physical training data. However, the traditional video surveillance shows some limitations. On the one hand, the development of computation speed is unable to keep pace with the increasing of application data; on the other hand, the inherent transmission overhead is

very large, and the provided network bandwidth and the transmission of sliced segments are also not always matched [3]. Therefore, it is very necessary to optimize the Adaptive Bit Rate (ABR) [4] and guarantee obtaining the real-time monitoring data.

The typical ABR algorithm includes caching stage and stable stage [5]. At the first stage, the ABR algorithm usually tends to fill up the cache as quickly as possible; at the second stage, the ABR algorithm usually tries the best to improve the quality of video segment and prevent the cache overflow. At present, there have been some ABR optimization algorithms, including the traditional ones and the Artificial Intelligence (AI-) based ones. To the best of our knowledge, the traditional ABR optimization algorithms cannot obtain the real-time network status to adapt to the dynamic network environment. On the contrary, the AI-based ABR optimization algorithms can adaptively adjust the network parameters and obtain the relatively optimal video transmission [6]. In terms of AI, the Reinforcement Learning (RL) [7] is the most popular representative. For RL, it can retrieve the demanded data by information exchanging between the intelligent agents and the external environment, without preparing the additional training datasets before training. Compared with the other RL-based ABR optimization algorithms, the Q-learning-based ABR optimization algorithms have better experience quality. However, the current Q-learning-based ABR optimization algorithms fail to encode for the continuous state values and cannot complete the fast convergence in terms of the large state space. As a conclusion, this paper improve the Q-learning to optimize the ABR algorithm.

In addition to the data monitoring, the quantification evaluation for the physical training data is also very significant. Specifically, the physical training features can be extracted by analyzing the monitored exercise data [8]; on this basis, the embedded laws on the physical training can be explored and the corresponding physical fitness evaluation model can be built, such as exercise effect, exercise tolerance, and improvement situation. Based on the evaluation model, the differentiated physical training educations can be effectively developed.

With the above considerations, this paper proposes a novel method based on RL and Markov model to monitor and evaluate the adolescent physical training effect. The major contributions are summarized as follows. (i) The Q-learning-based RL is exploited to optimize the ABR of surveillance video by combining the nearest neighbor algorithm. (ii) The Markov model is used to evaluate the health condition on the physical training by considering the energy consumption metric. (iii) The real-time monitoring system on exercise data is implemented, and the performance optimization effect on the adolescent physical training is demonstrated based on the system platform.

The rest of the paper is organized as follows. Section 2 reviews the related work. The improved Q-learning-based ABR optimization is proposed in Section 3. Section 4 gives the physical fitness evaluation model. The experimental results are shown in Section 5. Section 6 concludes this paper.

2. Related Work

The physical training has always been concerned and some most cutting-edge works have been developed. Buckinx et al. evaluated the effect of citrulline supplementation combined with high-intensity interval training on physical performance in healthy older adults [9]. Ana et al. proposed the multicomponent exercise training method by combining with the nutritional counselling to improve the physical education [10]. Konstantinos et al. presented a study to compare the effectiveness of virtual and physical training for teaching a bimanual assembly task and in a novel approach, where the task complexity was introduced as an indicator of assembly errors during final assembly [11]. Roland Van Den et al. studied the training order of explosive strength and plyometrics training on different physical abilities in adolescent handball players [12]. Rodrigo et al. investigated the plyometric training on the soccer players' physical fitness by considering muscle power, speed, and change-of-direction speed tasks [13]. Simpson et al. enhanced the physical performance in professional rugby league players via the optimised force-velocity training [14]. Unquestionably, the above references showed the professional research on the physical training. In spite of this, they did not provide the networked physical training mode, i.e., regardless of the transmission of data generated from the video surveillance. To this end, Sun and Zou concentrated on the video transmission and improved the performance of extended training by using mobile edge computing [3]. However, [3, 9–14] still did not pay attention to the ABR optimization during the transmission process of physical training data.

The ABR optimization plays an important role in the networked physical training mode. The traditional optimization algorithms usually are heuristics. Cicco et al. announced two policies to optimize ABR, i.e., gradual increasing, but accelerate decreasing for the bit rate [15]. The heuristic ABR optimization existed the suboptimal problem and the shock problem of transmission quality, which had the considerable influence on the experience quality. As a result, Mok et al. paid attention to the improvement of quality of experience [16], which could guarantee that the transmission quality kept the stable level. Furthermore, the traditional ABR optimization algorithms also had a limitation, i.e., could not build the predictable and describable mathematical models for the concrete problems. For such purpose, some researchers used the control theory to optimize the ABR, where the controller was responsible for handling the input parameters. For example, Xiong et al. proposed the adaptive control model based on fuzzy logic, which could effectively meet the dynamic network change [17]. Besides, Vergados et al. used the fuzzy logic to design the adaptive policy by inputting the varying information on caching [18], preventing the cache overflow. Although [17, 18] achieved the good effect on the ABR optimization, they could not obtain the real-time network status to adapt to the dynamic network environment. To this end, some AI-based ABR optimization algorithms were proposed. For example, Chien et al. mapped the feature values related to network bandwidth to the bit rate of video by using the

random forest classification decision tree [19]. Basso et al. [20] trained the classification model and estimated the bit rate based on the classification model. In fact, these ABR optimization algorithms like [19, 20] needed the ready-made dataset used for training. Instead, the RL-based ABR optimization algorithms could easily obtain the physical training data without preparing the additional datasets. Among them, the Q-learning-based ABR optimization algorithms could obtain the relatively best experience quality [21]. In spite of this, it was very difficult for the current Q-learning-based ABR optimization algorithms to encode for the continuous state values and realize the fast convergence in case of the large state space.

The evaluation model building of physical training data is very significant because it can effectively conduct and adjust the physical training. ElSamahy et al. presented a computer-based system for safe physical fitness evaluation for subjects undergoing aerobic physical stress, in which a proportional-integral fuzzy controller was applied to control the applied physical stress to ensure not exceeding the predefined target heart rate to satisfy safety [22]. Zhong and Hu designed a WebGIS-based interactive platform to collect and analyze national physical fitness-related indicators, including realizing the seven functional modules [23]. Heldens et al. studied the care data evaluation model to address the association between performance parameters of physical fitness and postoperative outcomes in patients undergoing colorectal surgery [24]. Ma proposed a kind of multilevel estimation and fuzzy evaluation of physical fitness and health effect of college students in regular institutions of higher learning based on classification and regression tree algorithm [25]. Qu et al. considered the physical fitness evaluation in children with congenital heart diseases versus healthy population [26]. Although the above references built the evaluation models for the physical fitness, they did not address the adolescent physical training. Regarding this, Guo et al. proposed a machine learning-based physical fitness evaluation model oriented to wearable running monitoring for teenagers, in which a variant of the gradient boosting machine combined with advanced feature selection and Bayesian hyperparameter optimization was employed to build a physical fitness evaluation model [27]. In spite of this, [27] did not concentrate the ABR optimization, which cannot complete the optimal performance optimization for the adolescent physical training.

3. Q-Learning-Based RL for ABR Optimization

The RL-based ABR optimization algorithms show the tradeoff between state space division and convergence speed. To be specific, if the state space is divided in the more fine-grained way, the more adequate states are generated and further the system behaviors can be more precisely described, while this causes the slow convergence problem. On the contrary, if the division granularity is relatively large, the number of states becomes small with it, while the convergence speed can be accelerated. Besides, these states in the ABR optimization problem are usually continuous, and the current Q-learning-based ABR optimization algorithms only

make the simple discrete processing for these states. Therefore, this section plans to combine the nearest neighbor algorithm to address the abovementioned problems.

3.1. ABR Decision Model. Suppose that each code rate involves N video segments, denoted by $seg_1, seg_2, \dots, seg_N$, respectively. The client can select the corresponding segment from some code rate according to the network status information, such as network bandwidth, caching condition, and so on. In fact, the video segment selection can be regarded as the sequential decision process, and the decision objective is to guarantee the stable video display with the high code rate on the condition where the network bandwidth keeps the dynamic change. Given this, this paper assumes that there is an intelligent agent to determine how to download the video segment. Mathematically, for any seg_i , we can observe the information like network bandwidth (denoted by BD_i), caching state (denoted by CH_i), and the previous segment's quality (denoted by q_{i-1}), and the corresponding environment state is defined as $s_i = (BD_i, CH_i, q_{i-1})$.

The intelligent agent selects the specific video segment from different code rates according to the perceived information. For the selection behavior regarding seg_i , it can be called as an intelligent action, denoted by a_i . After a_i is completed, CH_i will be updated by referring to BD_i and the selected code rate (denoted by $br(a_i)$). Let T_i denote the duration time of seg_i , and the corresponding download time is defined as follows:

$$Tdl_i = \frac{T_i * br(a_i)}{BD_i} \quad (1)$$

Furthermore, for seg_{i+1} , its corresponding cache is defined as follows:

$$CH_{i+1} = \max\{0, CH_i - Tdl_i\} + T_i, \quad (2)$$

which suggests that the segment lag time equals the caching consumption time. To the best of our knowledge, the strategy conducted by the intelligent agent usually is not optimal. Thus, we give the reward function (denoted by R_i) to conduct the intelligent agent reaching the optimal level. In spite of this, the design of reward function should follow the requirements of service quality, which mainly includes the following three factors: the quality of video segment, the quality varying value between two frames, and the risk coefficient with respect to the cache overflow. Therein, the higher evaluation on service quality comes from the higher first factor and the smaller second/third factor. With the above three factors consideration, R_i is defined as follows:

$$R(i) = q_i - \gamma|q_i - q_{i+1}| - \varphi_i, \quad (3)$$

where γ is a regulatory parameter to adjust the difference between q_i and q_{i+1} ; φ_i is used to measure the risk coefficient with respect to the cache overflow, which has two functions: guaranteeing that the caching keeps the safe level and avoiding such actions that cause the low caching since they trigger the duplicate caching events easily. As a result, φ_i is defined as follows:

$$\varphi_i = \min\{\alpha(\max\{0, Tdl_i - CH_i\}), 1\} + \beta(\max\{CH_i - CH_{i+1}, 0\})^2, \quad (4)$$

where the two parts of the right-hand side of equation (4) represent safe caching level and duplicate caching event, respectively; α and β are two regulatory parameters to, respectively, determine whether safe caching level and duplicate caching event are considered.

In fact, the ABR optimization problem based on the sequential segment selections by the intelligent agent can be built as the Markov decision model [28], which is expressed by four attributes, i.e., state space, action space, conditional transition probability, and instant reward function, denoted by S , A , P , and R , respectively. As mentioned earlier, the state space includes network bandwidth, caching state, and the previous segment's quality. The action space is the set of all available code rates. Particularly, the strategy conducted by the intelligent agent is the mapping relationship from state space to action space, i.e., $F: S \rightarrow A$. Furthermore, the corresponding s_i will change with it and be converted into s_{i+1} when a_i is finished, and the situation is called the conditional transition probability, denoted by $P(s_{i+1}|s_i, a_i)$. Moreover, the intelligent agent's decision objective is to obtain the optimal long-term benefits according to some strategy (F), and the long-term benefits function is defined as follows:

$$R(s_0; F) = \lim_{N \rightarrow \infty} E \left[\sum_{i=0}^{N-1} \lambda_i * R(i) \right], \quad (5)$$

where s_0 is the initial state; λ is the discount parameter and is between 0 and 1. When $\lambda = 0$, it means the action only pays attention to the current benefits irrespective of the expected benefits. With the increasing of λ , the action starts to pay attention to the expected benefits. Suppose that F^* is the optimal strategy while guaranteeing that the long-term benefits are the maximal, and we have

$$F^* = \arg \max_{\forall F} R(s_0; F). \quad (6)$$

3.2. Nearest Neighbor Algorithm for Q-Learning

3.2.1. K-Nearest Neighbor Proposal. In order to obtain the optimal strategy of formula (6), the long-term benefits function is modified with the recursive form, as follows

$$R(s_0; F) = r_0 + \lambda \sum_{s_1 \in S} P(s_1|s_0, a_0) * R(s_1; F). \quad (7)$$

As we know, formula (7) can be solved by the dynamic linear programming algorithm to obtain the optimal strategy. However, the dynamic linear programming has high computation complexity, and thus Kröse [29] uses the Q-learning method to obtain the optimal strategy with the relatively low computation complexity. In terms of Q-learning, it maintains one Q-table which includes the entries on mapping from the state to the action. As above mentioned, the Q-learning has two limitations; thus, this paper prepares to use K-nearest neighbor algorithm to optimize it.

The K-nearest neighbor algorithm is first proposed by Cover and Hart [30] in 1968, and it belongs to the instance-based learning method. In the K-nearest neighbor algorithm, the Euclidean distance between two samples is usually used to measure the similarity, where the larger Euclidean distance means the lower similarity. For example, suppose that $X = \{x_1, x_2, \dots, x_m\}$ and $Y = \{y_1, y_2, \dots, y_m\}$ are two data samples in terms of m -dimensional space, and the Euclidean distance between X and Y is defined as

$$\sqrt{\sum_{i=1}^m (x_i - y_i)^2}. \quad (8)$$

3.2.2. ABR Optimization. We employ the Q-learning based on K-nearest neighbor algorithm to optimize the ABR, and the corresponding state division is shown in Figure 1. We can observe that when and only when the middle value of each interval is determined, the state is found. Under such condition, the state's Q-value can be obtained by referring to the neighbouring Q-value(s). Suppose that d_i is the Euclidean distance between two states, and it is defined as follows:

$$d_i = \sqrt{(d_i^{\text{BD}})^2 + (d_i^{\text{CH}})^2 + (d_i^q)^2}, \quad (9)$$

where d_i^{BD} , d_i^{CH} , and d_i^q are the Euclidean distances regarding two network bandwidths, two cache states, and two previous segment's qualities respectively, and their computations are similar to formula (8). After obtaining these Euclidean distances between s_i and the state in each Q-table, the first K-nearest distances are selected and their corresponding Q-values are used to compute the array of Q-values for s_i , defined as follows:

$$Q(s_i, :) = \sum_{i=1}^K w_i * Q(s'_i, a_i), \quad (10)$$

here, s'_i is the neighbouring state with s_i ; w_i is the Q-value proportion of s'_i . If the corresponding d_i is larger, w_i is set to be smaller. Given an intermediate variable $\rho_i = 1/d_i$, w_i is defined as follows:

$$w_i = \frac{\rho_i}{\sum_{i=1}^K \rho_i}. \quad (11)$$

Furthermore, after the action is finished, it requires to update the Q-table according to the returned instant reward and the new state. If the current state is found in the state division table, i.e., $s_i \in S$, the updating of Q-value is defined as follows:

$$Q(s_i, a_i) = Q(s_i, a_i) + \eta \left[r_{i+1} + \lambda \max_{\forall a_i} Q(s_i + 1, a_i) - Q(s_i, a_i) \right]. \quad (12)$$

On the contrary, if the current state cannot be found in the state division table, i.e., $s_i \notin S$, the updating of Q-value is defined as follows:

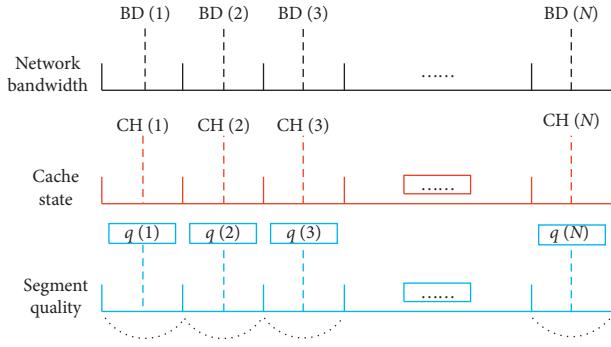


FIGURE 1: The state division in terms of the Q-learning based on K -nearest neighbor algorithm.

$$Q(s'_{i+1}, a_{i+1}) = Q(s'_i, a_i) + \eta * \Upsilon w_i, \quad (13)$$

$$\Upsilon = \sum_{i=1}^K R(i) + \xi \max_{a_i} \left\{ \sum_{a'_i} w'_i * Q(s''_i, a'_i) - \sum_{a_i} w_i * Q(s'_i, a_i) \right\}, \quad (14)$$

where Υ is an intermediate variable; ξ is a regularity parameter; s''_i is the next K state after s'_i .

According to the above statements, the pseudocode of ABR optimization based on Q-learning with consideration of K -nearest neighbor algorithm is described in Algorithm 1.

4. Physical Fitness Evaluation

The quantification evaluation for the physical training data is also very significant. In fact, the physical training process is intricate and has the strong randomness, which leads to the difficulty of quantification evaluation. The traditional physical evaluation models (e.g., [22–26]) usually consider the relatively simple factors with the subjectivity, which have a few limitations. Thus, it is required to find a proper model to evaluate the physical training.

4.1. Thought Incubation. The physical training process has the feature of randomness, that is, the subsequent exercise state only depends on the current exercise state and has no connection with the history exercise states. This conforms to the Markov process; thus, this paper uses the Markov model to simulate the physical training process and further present the related evaluation models, including individual exercise modelling and population exercise modelling. The main ideas are summarized as follows:

- (i) *Individual Exercise Modelling.* (i) The generated physical training data is given a rank; (ii) the matrix of transition probability is obtained by the varying data rank; (iii) the vector of stability probability is computed by referring to the stability of Markov process, to predict the stable state; and (iv) the subsequent physical training is conducted based on the exercise limit.

- (ii) *Population Exercise Modelling.* The first two steps are similar to those in the individual exercise modelling. The third step is to compare the generated population data and adjust the improvement degree to adapt to the whole physical training effect.

Furthermore, the generated energy during the physical training process usually reflects the situation of physical fitness; thus, this paper regards the energy consumption as the evaluation metric. Particularly for the adolescent physical training, the consumed energy gradually increases at the initial stage. After reaching the relatively stable level, it begins to decrease rapidly until the teenagers lose their strength. In terms of the individual exercise modelling, we adopt the energy consumption rate as the evaluation metric, which is defined as follows:

$$cr_t = \frac{EE_t}{ET_t}. \quad (15)$$

Among them, $t \in [1, t_{max}]$ is the time period used for data collection and t_{max} is the maximal number of periods; EE_t is the consumed energy of the t th data collection; and ET_t is the spent time to complete EE_t . In terms of the population exercise modelling, we adopt the improvement degree with respect to the energy consumption transition as the evaluation metric, which is defined as follows:

$$eK_{ij} = ep_{ij} * ed_{ij}^c. \quad (16)$$

Among them, ep_{ij} is the transition probability; ed_{ij} is the energy consumption span difference, and it can be computed by formula (8); and c is the regularity parameter.

4.2. Modelling for Two Situations

4.2.1. Individual Modelling. Based on the Markov model and the ABR optimization algorithm (see Section 3), the consumed energy data can be monitored and computed easily. After that, the sequence on energy consumption rates can be obtained by formula (15). For the sequence, the individual evaluation model based on the Markov process is described as follows:

- (i) *State Space Division.* The maximal value and the minimal value in the sequence are found, denoted by cr_{max} and cr_{min} , respectively. Suppose that there are θ divided state intervals, and the length of interval is defined as

$$\Delta cr = \frac{cr_{max} - cr_{min}}{\theta}. \quad (17)$$

On this basis, the divided intervals are $[cr_{min}, cr_{min} + \Delta cr), \dots, [cr_{min} + (\theta - 1)\Delta cr, cr_{max}]$.

- (ii) *Transition Probability Matrix Computation.* For the continuous time periods, their transition probabilities are computed, and one matrix used to record these transition probabilities can be obtained and denoted by eP .

```

Input: State space and action space
Output: Q-value
(1) Initialize Q-table;
(2) for each state, do
(3)   Compute  $Q(s_i, \cdot)$  with formula (10);
(4)   Confirm  $br(a_i)$  by  $Q(s_i, \cdot)$ ;
(5)   Request to download  $seg_i$ ;
(6)   Update  $CH_i$  with formula (2);
(7)   Compute  $R(i)$  with formula (3);
(8)   if  $s_i \in S$ , then
(9)     Update Q-value with formula (12);
(10)  else
(11)    Update Q-value with formula (13);
(12)  endfor

```

ALGORITHM 1: ABR optimization algorithm.

(iii) *Stable-State Vector Determination*. When t_{\max} energy consumption rates are completed, we give an initial state vector denoted by $eS_{t_{\max}}$ to satisfy

$$eS_{t_{\max}+1} = eS_{t_{\max}} * eP. \quad (18)$$

According to the stability of Markov chain, we can obtain a state vector $eS^* = \{es_1, es_2, \dots, es_\theta\}$ to satisfy $eS^* = eS_{t_{\max}} eP$, where eS^* is called the stable-state vector.

(iv) *Limited Energy Consumption Rate Computation*. For θ state intervals, their maximal values are selected, where er_i^{\max} denotes the maximal value of the i th interval, and the limited energy consumption rate is defined as

$$cr^{\lim} = \sum_{i=1}^{\theta} er_i^{\max} * es_i. \quad (19)$$

To sum up, if cr_i is larger than cr^{\lim} , it means that the current physical training is dangerous and the system will notify the teenagers to slow down the physical training.

4.2.2. *Population Modelling*. Suppose that one population includes Nump teenager individuals, and the average energy consumption for the population is defined as follows:

$$EP_{\text{ave}} = \frac{1}{\text{Nump}} \sum_{i=1}^{\text{Nump}} Ei_i, \quad (20)$$

where Ei_i is the consumed energy by the arbitrary teenager individual in the population. In a similar way, the sequence on the average energy consumptions can be obtained by formula (20), denoted by $EP_{\text{ave}}^1, EP_{\text{ave}}^2, \dots, EP_{\text{ave}}^{t_{\max}}$; that is, there are t_{\max} collection periods. For the sequence, the population evaluation model based on the Markov process is described as follows:

(i) *State Space Division*. The maximal value and the minimal value in the sequence are found, denoted by

EP_{ave}^{\max} and EP_{ave}^{\min} , respectively. Suppose that there are θ divided state intervals, and the length of interval is defined as

$$\Delta EP_{\text{ave}} = \frac{EP_{\text{ave}}^{\max} - EP_{\text{ave}}^{\min}}{\theta}. \quad (21)$$

Then, the divided intervals are $[EP_{\text{ave}}^{\min}, EP_{\text{ave}}^{\min} + \Delta EP_{\text{ave}}), \dots, [EP_{\text{ave}}^{\min} + (\theta - 1)\Delta EP_{\text{ave}}, EP_{\text{ave}}^{\max}]$.

- (ii) *Transition Probability Matrix Computation*. It is similar to the operation in the individual evaluation model.
- (iii) *Transition Improvement Degree Computation*. It can be obtained by formula (16).

In total, if eK_{ij} is larger than $\sum eK_{ij}/(\theta - 1)$, it means that the physical training effect has been improved and the system will notify the population to enhance the physical training.

5. Simulation Results

In this section, we pay attention to the simulation experiments. At first, we develop the data monitoring system. Then, we test the physical training evaluation models. Finally, the whole performance optimization on the adolescent physical training is verified. Meanwhile, the last two parts are based on the developed system platform. In particular, regarding the simulation settings, we make the different simulations and find one proper combination.

5.1. *System Implementation*. The real-time data monitoring system depends on computer technology, communication technology, and sports science, which provides the real-time exercise monitoring services according to collecting the data information with respect to the physical training. In terms of the adolescent physical training, the data monitoring system platform architecture is shown in Figure 2. We observe that the system platform includes four modules, i.e., data collection, data receiving and data sending, data analysis and handling, and data display. Among them, the last module

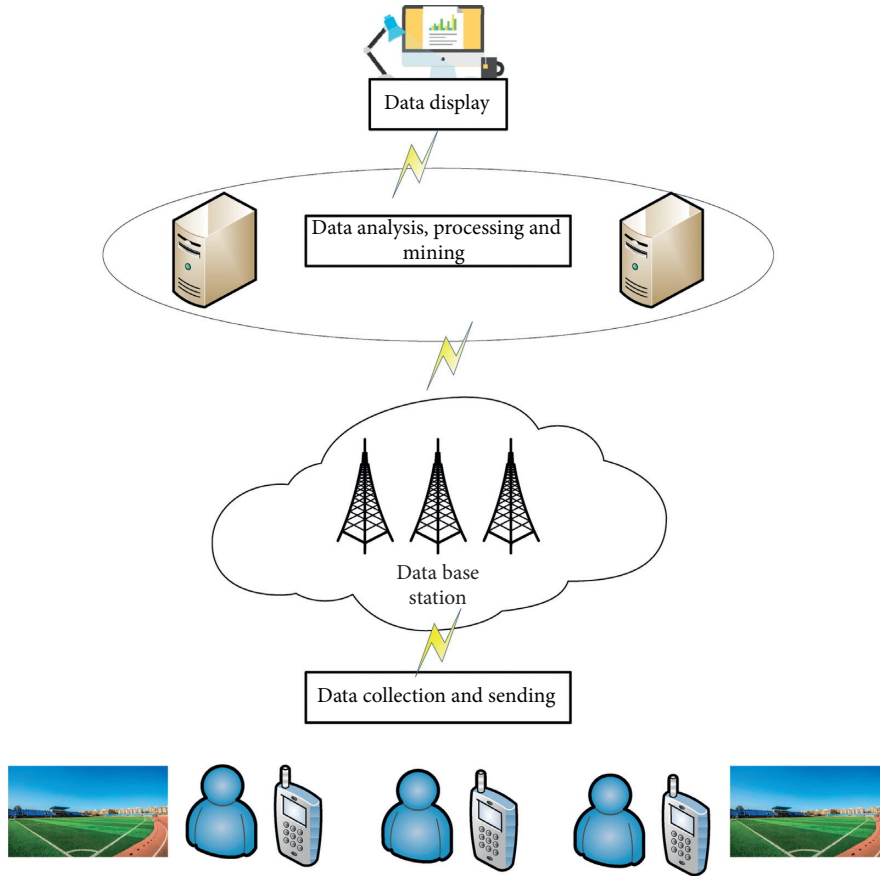


FIGURE 2: The data monitoring system platform architecture.

can provide the reference for the adolescent physical training directly according to the monitored data.

5.2. Model Evaluation. This section will evaluate two models, i.e., individual evaluation model and population evaluation model. The involved parameters are set as follows: $c = 3$, $t_{\max} = 300$, $\theta = 24$, and time period for 30 s. In addition, we use the deviation rate to measure whether the evaluation models can be acceptable. For the individual evaluation model, we test 1000 teenagers for 12 times experiments, where the frequency is once a day. The experimental results on the conducted physical training conditions are shown in Table 1. For the population evaluation model, we also test 1000 populations for 12 times experiments, where one population includes 20 teenagers and the frequency is once a day. The corresponding results on the population physical training conditions are shown in Table 2. Among them, the deviation rate is defined as the ratio of the number of improper conducts and the total number of experiments.

As seen from Tables 1 and 2, the deviation rate for each group experiment is always smaller than 0.3%. To be specific, the average deviation rate in terms of the individual evaluation model is 0.158% and that for the population evaluation model is 0.092%, and the two values can be controlled within the 0.16%, which implies that it is efficient to use the Markov model to evaluate the adolescent physical training.

Furthermore, it implies that the Markov model has better evaluation effect in terms of the population physical training situation because $0.158\% < 0.092\%$.

5.3. Performance Verification. This section will verify the optimization preformation of adolescent physical training by comparing with two benchmarks, i.e., [3, 27] published by Internet Technology Letters (ITL) and Computer Networks (CN), respectively. Meanwhile, the whole transmission time and packet loss rate are adopted as two performance verification metrics. The involved parameters are set as follows: $\gamma = 0.45$, $\alpha = 0.6$, $\beta = 0.4$, $\lambda = 0.9$, $K = 6$, $\eta = 0.35$, and $\xi = 0.4$. In addition, the number of simulations is set as 10. The experimental results on the whole transmission time and packet loss rate are shown in Tables 3 and 4, respectively.

It can be seen from Tables 3 and 4 that this paper always consumes the smallest whole transmission time and the lowest packet loss rate. This implies that this paper has the optimal optimization performance for the adolescent physical training. This is because this paper uses RL to obtain the relatively optimal solution and uses the Markov model to obtain the relatively accurate training effect. In addition, regarding the two metrics, we show the corresponding dispersion coefficients to evaluate the stability, as shown in Figure 3. We observe that this paper always has the smallest dispersion coefficient due to the stability guarantee of using

TABLE 1: The experimental results on the individual physical training conditions.

| Experiment no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------------------|-----|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| #Correct conduct | 998 | 1000 | 997 | 999 | 999 | 998 | 999 | 999 | 999 | 997 | 998 | 1000 |
| #Improper conduct | 2 | 0 | 3 | 1 | 1 | 2 | 1 | 1 | 1 | 3 | 2 | 0 |
| Deviation rate (%) | 0.2 | 0 | 0.3 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.1 | 0.3 | 0.2 | 0 |

TABLE 2: The experimental results on the population physical training conditions.

| Experiment no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------------------|------|-----|-----|------|-----|------|-----|-----|------|-----|------|-----|
| #Correct conduct | 1000 | 998 | 999 | 1000 | 997 | 1000 | 999 | 999 | 1000 | 999 | 1000 | 998 |
| #Improper conduct | 0 | 2 | 1 | 0 | 3 | 0 | 1 | 1 | 0 | 1 | 0 | 2 |
| Deviation rate (%) | 0 | 0.2 | 0.1 | 0 | 0.3 | 0 | 0.1 | 0.1 | 0 | 0.1 | 0 | 0.2 |

TABLE 3: The experimental results on the whole transmission time (ms).

| Experiment no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| This paper | 43.26 | 44.61 | 43.97 | 45.67 | 43.09 | 44.56 | 44.13 | 45.02 | 43.86 | 42.61 |
| ITL | 56.67 | 55.37 | 58.64 | 57.26 | 55.97 | 56.43 | 59.06 | 58.34 | 53.49 | 55.73 |
| CN | 83.18 | 82.67 | 84.62 | 85.64 | 85.31 | 84.64 | 83.59 | 84.22 | 82.98 | 84.55 |

TABLE 4: The experimental results on the packet loss rate (%).

| Experiment no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| This paper | 0.156 | 0.173 | 0.146 | 0.152 | 0.139 | 0.109 | 0.154 | 0.167 | 0.141 | 0.121 |
| ITL | 0.648 | 0.694 | 0.703 | 0.625 | 0.713 | 0.633 | 0.687 | 0.701 | 0.692 | 0.681 |
| CN | 0.355 | 0.386 | 0.321 | 0.364 | 0.339 | 0.309 | 0.359 | 0.316 | 0.356 | 0.392 |

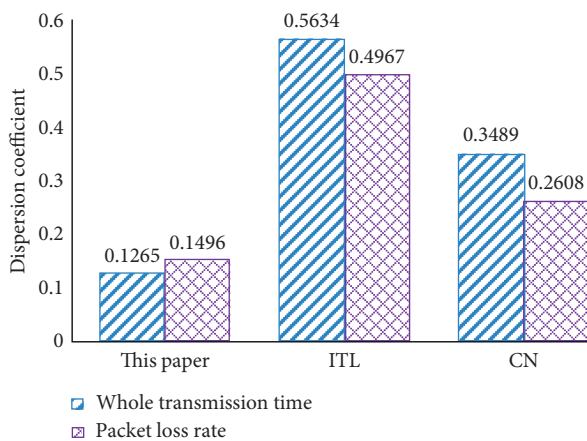


FIGURE 3: The dispersion coefficients on two metrics.

RL, which implies that the performance mechanism is the most stable.

6. Conclusions

The physical fitness of teenagers has attracted the global attention for a long time because it has a considerably important influence on the rise and fall of each country. This paper proposes to optimize the adolescent physical training based on RL and Markov model. Because the RL-based ABR optimization algorithms shows the tradeoff between state space division and convergence speed, this paper improves

the Q-learning by using the K -nearest neighbor algorithm. In addition, we also present the evaluation models on the physical fitness, including individual exercise modelling and population exercise modelling, based on the Markov model. Moreover, we make the simulation experiments based on the developed data monitoring system platform, and the results have demonstrated that this paper has always the optimal optimization performance for the adolescent physical training with the most stable state. In the future, we will deploy more functions in our system platform, such as adaptive recognition and falling warning. Besides, we also make the large-scale experiments based on the real testbed instead of the system platform.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work was supported by the Special Funds Project for Basic Scientific Research in Central Universities (no. 451170306081) and the Research Project on the Realization PATH of High-Quality Development of School Competitive Sports in Jilin Province (no. 2020C088). In addition, the

authors also thank Yuanshuang Li who is an expert in the field of video transmission to provide the great help for ABR knowledge.

References

- [1] Y. Jin and X. Wang, "Study on safety mode of dragon boat sports physical fitness training based on machine learning," *Safety Science*, vol. 120, pp. 1–5, 2019.
- [2] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel, "A conceptual framework and taxonomy of techniques for analyzing movement," *Journal of Visual Languages & Computing*, vol. 22, no. 3, pp. 213–232, 2011.
- [3] L. Sun and Y. Zou, "Mobile edge computing based video surveillance model for improving the performance of extended training," *Internet Technology Letters*, 2020.
- [4] A. Yang, X. Wu, Y. Qiao, and Y.-n. Sun, "Bit-rate adaptive optical performance monitoring method for fiber communication systems," *Optics Communications*, vol. 284, no. 1, pp. 436–440, 2011.
- [5] A. Rao, Y. Lim, C. Barakat et al., "Network characteristics of video streaming traffic," in *Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies*, pp. 1–12, Tokyo Japan, December 2011.
- [6] B. K. Mohanta, D. Jena, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Addressing security and privacy issues of IoT using blockchain technology," *IEEE Internet of Things Journal*, vol. 111, pp. 1–17, 2020.
- [7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, *An Introduction to Reinforcement Learning*, IEEE Press, Hoboken, NJ, USA, 2005.
- [8] H. Yuan, J. Wang, J. Liu, and S. Li, "Research of zigbee and big data analysis based pulse monitoring system for efficient physical training," *Procedia Computer Science*, vol. 80, pp. 2357–2361, 2016.
- [9] F. Buckinx, L. P. Carvalho, and V. Marcangeli, "High intensity interval training combined with L-citrulline supplementation: effects on physical performance in healthy older adults," *Experimental Gerontology*, vol. 140, 2020.
- [10] C. Ana, P. Beatriz, G. Paula et al., "Multicomponent exercise training combined with nutritional counselling improves physical function, biochemical and anthropometric profiles in obese children: a pilot study," *Nutrients*, vol. 12, no. 9, pp. 1–15, 2020.
- [11] K. Konstantinos, C. Francesco, M. Panagiotis, and K. Simon, "Effectiveness of virtual versus physical training: the case of assembly tasks, trainer's verbal assistance, and task complexity," *IEEE Computer Graphics and Applications*, vol. 40, no. 5, pp. 41–56, 2020.
- [12] T. Roland Van Den, R. Truls Valland, and O. Dustin, "Comparison of effects of training order of explosive strength and plyometrics training on different physical abilities in adolescent handball players," *Biology of Sport*, vol. 37, no. 3, pp. 239–246, 2020.
- [13] R. Rodrigo, A. Cristian, G. Felipe et al., "Effects of combined surfaces vs. single-surface plyometric training on soccer players' physical fitness," *Journal of Strength and Conditioning Research*, vol. 34, no. 9, pp. 2644–2653, 2020.
- [14] A. Simpson, M. Waldron, E. Cushion et al., "Optimised force-velocity training during pre-season enhances physical performance in professional rugby league players," *Journal of Sports Sciences*, 2020.
- [15] L. D. Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, pp. 145–156, San Jose, CA, USA, February 2011.
- [16] R. K. P. Mok, X. Luo, E. Chan, and R. K. C. Chang, "QDASH: a QoE-aware DASH system," in *Proceedings of the 3rd Annual ACM Conference on Multimedia Systems*, pp. 11–22, Chapel Hill, NC, USA, February 2012.
- [17] P. Xiong, J. Shen, Q. Wang et al., "NBS: a network-bandwidth-aware streaming version switcher for mobile streaming applications under fuzzy logic control," in *Proceedings of the IEEE First International Conference on Mobile Services*, pp. 48–55, Honolulu, HI, USA, June 2012.
- [18] D. J. Vergados, A. Michalas, A. Sgora et al., "A control-based algorithm for rate adaption in MPEG-DASH," in *Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications*, pp. 1–5, Chania, Greece, July 2014.
- [19] Y. Chien, C. Lin, and M. Chen, "Machine learning based rate adaptation with elastic feature selection for HTTP-based streaming," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1–6, Turin, Italy, June 2015.
- [20] S. Basso, A. Servetti, E. Masala et al., "Measuring DASH streaming performance from the end users perspective using neubot," in *Proceedings of the 5th Annual ACM Conference on Multimedia Systems*, pp. 1–6, Singapore, March 2014.
- [21] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [22] E. ElSamahy, A. Genedy, M. A. Abbass et al., "A computer-based system for safe physical fitness evaluation," in *Proceedings of the 4th International Conference on Biomedical Engineering and Informatics*, pp. 1443–1447, Shanghai, China, October 2011.
- [23] Y. Zhong and W. Hu, "The research on WebGIS-based information integration and data analysis platform for China's Physical Fitness and the National Fitness Program," in *Proceedings of the 2nd IEEE International Conference on Computer and Communications*, pp. 135–139, Chengdu, China, October 2016.
- [24] A. F. J. M. Heldens, B. C. Bongers, A. F. Lenssen, L. P. S. Stassen, W. F. Buhre, and N. L. U. van Meeteren, "The association between performance parameters of physical fitness and postoperative outcomes in patients undergoing colorectal surgery: an evaluation of care data," *European Journal of Surgical Oncology*, vol. 43, no. 11, pp. 2084–2092, 2017.
- [25] L. Ma, "Multi-level estimation and fuzzy evaluation of physical fitness and health effect of college students in regular institutions of higher learning based on classification and regression tree algorithm," in *Proceedings of the 2018 International Conference on Virtual Reality and Intelligent Systems*, pp. 201–205, Changsha, China, August 2018.
- [26] J. Qu, H. Shi, X. Chen et al., "Evaluation of physical fitness in children with congenital heart diseases versus healthy population," *Seminars in Thoracic and Cardiovascular Surgery*, 2020.
- [27] J. Guo, L. Yang, R. Bie et al., "An XGBoost-based physical fitness evaluation model using advanced feature selection and Bayesian hyper-parameter optimization for wearable running monitoring," *Computer Networks*, vol. 151, pp. 166–180, 2019.
- [28] G. Kalnoor and G. Subrahmanyam, "A review on applications of Markov decision process model and energy efficiency in wireless sensor networks," *Procedia Computer Science*, vol. 167, pp. 2308–2317, 2020.

- [29] B. J. A. Kröse, “Learning from delayed rewards,” *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 233–235, 1995.
- [30] G. Chirici, M. Mura, D. McInerney et al., “A meta-analysis and review of the literature on the k -nearest neighbors technique for forestry applications that use remotely sensed data,” *Remote Sensing of Environment*, vol. 176, pp. 282–294, 2016.

Research Article

Human Falling Detection Algorithm Based on Multisensor Data Fusion with SVM

Daohua Pan ^{1,2} Hongwei Liu ¹ Dongming Qu,³ and Zhan Zhang¹

¹School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, Heilongjiang, China

²Department of Electronic and Information Engineering, Heilongjiang Vocational College for Nationalities, Harbin 150066, Heilongjiang, China

³Department of Financial Technology, China Construction Bank, Harbin 150001, Heilongjiang, China

Correspondence should be addressed to Daohua Pan; pandaohua@ftcl.hit.edu.cn and Hongwei Liu; liuhw@hit.edu.cn

Received 23 September 2020; Revised 14 October 2020; Accepted 17 October 2020; Published 31 October 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Daohua Pan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Falling is a common phenomenon in the life of the elderly, and it is also one of the 10 main causes of serious health injuries and death of the elderly. In order to prevent falling of the elderly, a real-time fall prediction system is installed on the wearable intelligent device, which can timely trigger the alarm and reduce the accidental injury caused by falls. At present, most algorithms based on single-sensor data cannot accurately describe the fall state, while the fall detection algorithm based on multisensor data integration can improve the sensitivity and specificity of prediction. In this study, we design a fall detection system based on multisensor data fusion and analyze the four stages of falls using the data of 100 volunteers simulating falls and daily activities. In this paper, data fusion method is used to extract three characteristic parameters representing human body acceleration and posture change, and the effectiveness of the multisensor data fusion algorithm is verified. The sensitivity is 96.67%, and the specificity is 97%. It is found that the recognition rate is the highest when the training set contains the largest number of samples in the training set. Therefore, after training the model based on a large amount of effective data, its recognition ability can be improved, and the prevention of fall possibility will gradually increase. In order to compare the applicability of random forest and support vector machine (SVM) in the development of wearable intelligent devices, two fall posture recognition models were established, respectively, and the training time and recognition time of the models are compared. The results show that SVM is more suitable for the development of wearable intelligent devices.

1. Introduction

Population aging problem is increasingly prominent, and the technology to help adapt to the aging has been widely concerned. Due to the aging of physiological structure and the decline of physical function for the elderly, the probability and frequency of accidental falls in the elderly are very high. The development of human fall detection technology has practical application value. In daily life, especially in wet places such as the kitchen, bathroom, and toilet, elderly people slip easily. If the elderly suffer from cerebral hemorrhage, heart disease, and other diseases, falls may also threaten their life. Therefore, the detection of falls is particularly important for the elderly. As one of the key components of intelligent devices, sensors have been widely

used in the field of smart portability. With the expansion of the application range, more and more high requirements are put forward for the preparation of sensors with high sensitivity, high precision, and flexibility. At present, the fall detection based on wearable sensor is relatively accurate, which usually does not affect the detection results due to environmental changes. It is flexible and easy to use. It can also install sensors such as heart rate and blood pressure to monitor the physical condition of the elderly in real time. The multisensor attitude detection system combines multiple data together to get the prediction results and real-time output. Compared with the detection system based on a single sensor, the cooperative use of multiple sensors can further improve the accuracy and reliability of prediction.

The research of sensors and fall prevention is deep. Wang's research introduced the design and implementation of RT fall (Real-Time fall). The system was a real-time, noncontact, low-cost, and accurate indoor fall detection system using commercial Wi-Fi equipment. Using the phase and amplitude of fine-grained channel state information (CSI) that can be accessed in commercial Wi-Fi devices, RT fall achieved the goal of real-time automatic segmentation and fault detection for the first time. The sharp drop of power spectrum of falls was found in time-frequency domain, which provided a new idea for new feature extraction and precise fall detection. However, Wang's research was biased towards theory and had no practical support [1]. He established a motion model based on three-dimensional acceleration and gyroscope to analyze the difference between activities of daily living (ADLs) and falls. At the same time, He introduced k-NN algorithm and sliding window technology and developed an intelligent fall detection and alarm system composed of wearable motion sensor board and smart phone, but there was a problem of low sensitivity [2]. Thanh had developed a low-cost fall detection system to accurately detect accidental falls in the elderly. He used fall detection algorithm to compare acceleration with lower and higher fall thresholds to accurately detect falls. However, the system performance therein was not good, and the energy requirements are high, and the practicability was not high [3]. Kepski proposed a fall detection algorithm based on depth map sequence and wireless inertial sensor data worn by the monitored person. Kepski's experimental verification was carried out on a freely obtained dataset consisting of synchronized depth and acceleration data. Kepski had done a lot of experiments in the scene. A static camera faced the scene, and an active camera observed the same scene from above. Experiments on human detection, tracking, and real-time fall detection verified the effectiveness and reliability of the proposed method. However, the cost of this method was high, so it was not suitable for large-scale promotion [4].

In this study, the effectiveness of the fall detection algorithm based on multisensor data fusion is verified by using the data of 100 volunteers simulating falls and daily activities and whether the increase of sample number in machine learning can increase the recognition rate of the system. The comparison of random forest algorithm and support vector machine algorithm, which is more suitable for wearable devices, to predict the trend of falls in recognition rate and recognition time is of great significance for the elderly to drop down.

2. Wearable Sensor and Fall Detection Algorithm

2.1. Wearable Sensor. Wearable devices integrate sensors into monitoring devices in real time. When the human body falls down, the alarm can be triggered in real time. The advantages of this method are obvious, not limited by time and space, and convenient transportation and low cost.

2.1.1. Flexible Materials. Common sensors are composed of rigid and noncurved substrates, such as those with silicon substrates. The current sensors have been widely used in our real life, but there are also some shortcomings, such as stiffness, sensitivity, and flexibility. Flexible sensor has flexibility; because of its wear, it needs to use flexible materials, such as some elastic staples, threads, or fiber fabrics; organic polymer also has corresponding applications [5]. The main substrates used are PET and REN because they have the advantages of transparency and low cost compared with other organic polymers. In the development of wearable health medical devices, PDMS, PEN, PI, and P are often used in the development of flexible sensors. The sensor electrode is composed of new driving materials, such as carbon nanomaterials and metal nanoparticles. The starting materials are graphene, carbon nanotube (CNT), carbon fiber [6], silver, gold, and nickel. PDMS, PET, PEN, and PI are commonly used as insulating substrates for the development of flexible sensors. The differences between these polymer materials mainly lie in Young's factor and refractive index. We usually choose the appropriate manufacturing process according to the size of flexible sensors [7].

2.1.2. Types of Sensing Indicators. Wearable sensors are mainly used to detect human normal indicators, so they have been widely used in our lives. Generally speaking, through the detection of human normal indicators, such as blood pressure, cholesterol content, and pH value, testing them can put forward corresponding reference suggestions for human health. The main application tools in life are electrochemical sensors, pressure and stress sensors, and magnetic field sensors.

2.1.3. System Composition and Characteristics. At present, wearable sensors to detect falls usually use three-axis acceleration sensors [8]. By integrating the acceleration sensor into relevant wearable devices, the human body wears it and collects motion information from the human body. After analyzing and processing the collected information, it can be judged whether the human body falls [9]. For example, using the possibility and working principle of the three-axis acceleration ADXL345, the three coordinates of ADXL345 correspond to the left and right, the front and rear directions of the human body, and the change of acceleration in each direction. When the human body falls, the acceleration in all three directions will change. Acceleration changes in three directions should be calculated and appropriate thresholds should be set to determine whether the elderly fall [10]. The fall detection based on wearable sensor is relatively accurate and generally does not affect the detection results due to environmental changes, and it is flexible and easy to use. Sensors such as heart rate and blood pressure can also be installed to monitor the physical condition of the elderly in real time. For the sensor beam fall detection algorithm, the commonly used algorithm should specifically include the

methods based on the following contents: time domain analysis and human fall behavior recognition method based on threshold analysis. The human fall process algorithm based on time series analysis can be used for the prediction and detection of human falls not only to predict whether human falls will occur but also to identify human falls [11].

2.2. Fall Detection Algorithm Based on Multisensor Data Fusion

2.2.1. Analysis of Fall Process and Design of Detection Algorithm.

According to the impact strength to the ground, falls can be divided into “hard fall” and “soft fall.” “Hard fall” refers to a fall resulting from physical illness due to impact. “Soft fall” refers to a fall that does not cause harm due to low impact strength, because the elderly fall slowly against the wall. Due to the greater threat of hard fall to the body, this paper focuses on the analysis of this kind of fall. According to the principle of mechanics, when the projection of the center of gravity to the ground cannot fall into the supporting surface, the mechanism of body force balance will be broken. If the center of gravity or supporting surface is not adjusted in time, a fall may occur. This imbalance state will be manifested in two forms: the body lands at certain acceleration, and the body posture angle changes [12]. This fall can be divided into four stages:

- (1) The fall begins: the fall does not occur, but the balance was destroyed, and the body acceleration and position angle do not change at this time.
- (2) Down phase: from the body losing balance to the end of contact with the ground, the acceleration value changes, and the stop angle and angular velocity of the object rotating around the axis change.
- (3) Impact phase: from the moment when the body contacts the ground until it stops on the ground, in this stage, the acceleration value in the middle of the body will change dramatically and bear the maximum impact, which is also the stage of causing damage [13].
- (4) Immobility phase: the postimpact phase is usually characterized by lying on the ground for a period of time.

The whole process (including four steps) of fall detection algorithm based on multisensor and multiparameter data fusion is summarized as follows: SV and BVA values determine whether the fall has started. When the data of five consecutive points exceed the limit a , it shall be regarded as the beginning of decline. At this time, the stop angle values of Pitch1, Roll1, and Yaw1 are recorded to determine the descent phase. When the data of five consecutive points exceeds the B limit, the descent phase occurs and the timer is activated. BVA and SVD were used to determine the impact stage. When the data of five consecutive points exceeds the limit C , the collision phase occurs. Disable the timer and read the value to check whether it is less than 1 second. Use BVA, Pitch, Roll, and Yaw to determine the immobility phase. When 400 consecutive data points are within a

specific range D , they are in the immobility phase, and the position angle values of Pitch2, Roll2, and Yaw2 are recorded [14]. The two recorded stop angle values are used to determine the type of fall.

2.2.2. Analysis of Control System Design Requirements.

When it is determined that a human body has fallen, the system must be able to send a text message to a specific person through GPRS to send an alarm. Therefore, the basic program of the controller is as follows: the inertial sensor continuously detects the data and sends it to the MCU for processing. According to the postposition detection algorithm, the MCU uses the processed data to determine the current rear position. The fall recognition based on threshold value is proposed. If it fails, decide whether to perform alarm operation, such as voice unit, engine unit, and GPRS unit [15, 16]. In addition, if the current device is connected to the mobile phone through Bluetooth, the mobile phone application can send parameter adjustment instructions to the device, such as angle adjustment command, engine vibration command, and size command voice unit [17, 18]. When using, if the hardware adds other updated functions or there are problems when using the device, you can remotely update the latest program in GPRS through the server. Through the analysis of workflow, the design controller must have the following four functions:

- (1) Communication function, advanced computer communication function, and mobile application server.
- (2) Storage operation, parameter control command, and bad attitude record storage.
- (3) The operation of receiving and processing data, receiving data from inertial sensors, and processing to detect falls.
- (4) Input and output functions, including ADC power detection, IO port operation, engine, and voice drive.

2.2.3. Data Fusion Algorithm.

The classical data fusion algorithms include multidimensional scaling (MDS) and Isomap, while the traditional data fusion algorithms include principal component analysis (PCA). MDS algorithm displays the points in the high-dimensional coordinate system to the low-dimensional coordinate system and ensures the minimum distance between the two points. After the data is reduced, the best high-dimensional local space is selected for dimensionality reduction. In MDS algorithm, the distance table of sample m in the original space is D , and the distance table of lower space is Z . After selecting the initial point, the gradient method is used to solve the low-dimensional mapping, so that DZ is the minimum. MDS does not need to make assumptions on data and can process large-scale data, but the calculation cycle is long, and the samples will fall into local optimum. Isomap uses the same basic algorithm as MDS, while Isomap uses geographic distance to reflect the low dimension of the actual multistructure calculation. For the n -dimensional matrix, after the space input distance DX

(I, J) between two points is specified by Isomap, the shortest path $DG(I, J)$ is calculated by Dijkstra algorithm, which is used to estimate the geological distance. Finally, DG uterine distance was used as the input of MDS to obtain the result of reducing the size [19, 20]. When the number of N vectors increases, the distance between the two points is closer to the geographical distance, but it will last longer; if n is too small, the geological distance will not be accurate.

PCA algorithm is called basic element analysis, which uses linear transformation to transform the original data into a new coordinate system to compress the proportion of the original data [21]. For the initial M -dimension and X -dimension table, the algorithm must find the interface table and then find the attributes through the combination table. Then, according to the attributes, they are sorted by row from top to bottom, and the first row K is taken to form the matrix P ; $y = PX$ is the result of dimension reduction of dimension K . The algorithm runs fast, but it is difficult to deal with nonlinear data.

2.3. Support Vector Machine and Random Forest. Mechanical learning is a multidisciplinary interdisciplinary subject that integrates probability theory, statistics, approximation theory, and other related theories. Its research scope includes data mining, computer vision, and speech recognition [22]. At present, there are many kinds of mechanical learning algorithms. For wearable multisensor devices, support vector machine or random forest is recommended.

2.3.1. Support Vector Machine. This is a two-layer recognition method. For a dataset, the linear classifier with the largest separation distance in its feature space is found. The purpose is to maximize the distance between the two categories and develop it into a method to solve problems. When the data is linearly inseparable, the current data must be transformed into high-dimensional space for effective classification. In the support vector machine algorithm, the function that can complete the dimension mapping is called the core function, which can solve the most complex problems, operate the internal products in the dimension space, and enable the data to be classified in the dimension space [23]. The basic principle is to transform the two types of data to be classified into an internal carrying volume and a dimension space through core function transformation. At present, the commonly used kernel functions include linear kernel function, polynomial kernel function, radial basis function (RBF) kernel function, and Fourier kernel function [24].

As shown below, formula (1) is a linear kernel function. For linear kernel function, there is no special parameter to be set.

$$K(x, y) = x \cdot y. \quad (1)$$

Formula (2) is a polynomial kernel function. For polynomial kernel function, D is used to set the highest degree of this term of polynomial kernel function. The

default value is 3. C is used to set coef0 in kernel function, and the default value is 0.

$$K(x, y) = (x \cdot y + c)^d. \quad (2)$$

Formula (3) is a radial basis function. For radial basis function, gamma is used to set gamma parameter in kernel function. The default value is $1/K$ (K is the number of categories).

$$K(x, y) = \exp(-\gamma \|x - y\|^2). \quad (3)$$

Formula (4) is a Fourier kernel function. For the Fourier kernel function, q is used to set the gamma parameter in the kernel function. The default value is $1/K$ (K is the number of categories).

$$K(x, y) = \frac{1 - q^2}{2(1 - 2q \cos(x - y) + q^2)}. \quad (4)$$

Among these kernel functions, the operation of kernel basis is more complex than that of polynomial kernel, which may have a better influence on the relationship between classification and classification and can deal with the characteristics of linear indivisible data. Therefore, this paper selects the fundamental kernel function as the core function of support vector machine algorithm.

2.3.2. Random Forest. This is a multidecision tree classifier. Its output category is created after comparing the output results of all decision trees. The advantage of random forest is that it can create a high-precision classifier, process a large number of input variables, evaluate the significance of variables, and balance the error classification of unbalanced material sets. Based on the decision tree theory, N data samples are randomly selected from the original dataset to form a sample subset, which is determined by the sample subset. According to the above steps, create other decision trees to form a random forest. When the forest category is evaluated most, it is determined according to the data of the new classification tree. The random forest algorithm is an improved algorithm based on the decision tree algorithm. It contains multiple decision trees, and the creation of each decision tree depends on the samples collected independently each time. The process of random forest algorithm should be divided into the following steps:

- (1) Select random data sample n from the original dataset.
- (2) The random K attributes are derived from all attributes, and a decision tree is created based on the samples selected in (L) based on these attributes.
- (3) Repeat (1) and (2) for a total of M times to generate a decision tree and create a random forest.
- (4) When classifying new data, each tree will make a decision, and all the decision tree votes are used to determine the final classification result.

Compared with other mechanical learning algorithms, random forest has the following advantages:

- (1) It can process dimensional data and identify relationships between them.
- (2) The classification model has certain stability for effective evaluation of missing data.
- (3) Random forest has a strong generalization effect.
- (4) The random sampling makes the classification model avoid overpositioning.

3. Experimental Design and Parameter Selection

3.1. Experimental Methods. This paper designs a fall detection system based on multisensor data fusion and analyzes the four stages of fall. By using the method of data fusion, three characteristic parameters representing human acceleration and posture change are extracted, and a fall detection algorithm based on threshold crisis is designed.

To be safe, we recruited 100 college students to simulate the experiment of falling. Their average age was 21 years, and their average height was 172 cm. A total of 100 volunteers were tested on a 10 cm protective mat for forward, backward, and lateral three times. The multiple sensors were located at the waist. Each volunteer also completed a series of daily actions such as walking (five steps), standing (3 s), sitting (5 s), squatting down (3 s), lying (10 s), going upstairs (ten steps), and going downstairs (ten steps). The 3D acceleration and angle data were collected from the fall detection system to verify the effectiveness of the algorithm. Manual inspection was required to ensure that the collected data should be available. According to the collected data, the data waveform of each one, energy is calculated. After pre-processing, the corresponding features are derived from each station data, and these features are taken as the original sample set.

3.2. Selection of Characteristic Parameters. According to the analysis of falling process, the following acceleration characteristic parameters were extracted:

$$SV = \sqrt{A_X^2 + A_Y^2 + A_Z^2}. \quad (5)$$

Among them, A_X , A_Y , and A_Z are the outputs of accelerometer in three directions. SV reflects the instantaneous acceleration of the body in the process of falling, which is composed of dynamic and static parts. Dynamic acceleration (SVD) is the high-frequency part of SV, which reflects the intensity of acceleration change, and can be used for impact judgment, which is obtained by formula (6); static acceleration refers to the acceleration of body at rest, usually 1 g.

$$SVD = \sqrt{(A_X^{(1)})^2 + (A_Y^{(1)})^2 + (A_Z^{(1)})^2}. \quad (6)$$

Among them, $A_X^{(1)}$, $A_Y^{(1)}$, and $A_Z^{(1)}$ are the data of A_X , A_Y , and A_Z after passing through the third-order high-pass Butterworth filter with cut-off frequency of 0.2 Hz.

The vertical acceleration dynamic quantity (BVA) of the body is obtained by

$$BVA = \frac{SV^2 - SVD^2 - G^2}{2G}. \quad (7)$$

The peak value of BVA appears during impact, which can be used to explain the oscillation behavior during impact. BVA can be used to determine the impact.

3.3. Validation of Algorithm Effectiveness. In order to verify the effectiveness of the proposed algorithm, experiments are carried out. Firstly, the relevant threshold is designed; then the data of volunteers' actions are collected and processed by the above algorithm, the performance of characteristic parameters is analyzed, and the accuracy rate SE and recognition rate SP are calculated. SE is defined in formula (8) and SP is defined in formula (9).

$$SE = \frac{D}{S} \times 100\%, \quad (8)$$

where D is the number of successful detections and S is the total number of falls.

$$SP = \left(1 - \frac{A}{B}\right) \times 100\%. \quad (9)$$

Among them, A is the number of times that was mistakenly detected as falling and B is the total number of daily actions.

Sensitivity is the capacity to detect a fall. It is given by the ratio between the number of detected falls and the total falls that occurred:

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \quad (10)$$

Specificity is the capacity to avoid false positives. Intuitively it is the capacity to detect a fall only if it really occurs:

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad (11)$$

where the meanings of TP, FN, FP, and TN are shown in Table 1.

4. Analysis of Experimental Results

4.1. Performance of Multifeature Parameter Fall Detection Algorithm in Falls. The performance of the algorithm is shown in Table 2 and Figure 1. If a feature is detected, it will be accumulated once.

It can be seen from Table 2 and Figure 1 that the algorithm proposed in this paper has successfully detected 175 times (180 times in total) and falls (SE = 96.69%). It has good performance in front and back falls, but only 90% of them fall in the horizontal direction, which is mainly due to the high threshold value and the missing detection in the fall. The main reason is that the fall effect is relatively 'slow' due to the support force when the foot falls to the side. Both SVD and BVA can track the fall stage. Because the stop angle changes before and after the fall, it recovers. Combined with these four stages, the accuracy of the algorithm is as high as

TABLE 1: Possible outputs of a fall detection system.

| | A fall occurs | A fall does not occur |
|------------------------|---------------------|-----------------------|
| A fall is detected | True positive (TP) | False positive (FP) |
| A fall is not detected | False negative (FN) | True negative (TN) |

TABLE 2: Performance of this algorithm in fall.

| Action | | Characteristic parameters | | | |
|--------------|------------------|---------------------------|-----|-----|---------------------|
| | | SV | SVD | BVA | Number of successes |
| Fall forward | The fall begins | 54 | | 60 | 60 |
| | Down phase | | | | 60 |
| | Impact phase | | 60 | 60 | 60 |
| | Immobility phase | | | 58 | 60 |
| Fall back | The fall begins | 59 | | 60 | 60 |
| | Down phase | | | | 60 |
| | Impact phase | | 60 | 60 | 60 |
| | Immobility phase | | 58 | | 60 |
| Sideways | The fall begins | 50 | | 60 | 60 |
| | Down phase | | | | 60 |
| | Impact phase | | 60 | 60 | 60 |
| | Immobility phase | | | 60 | 60 |

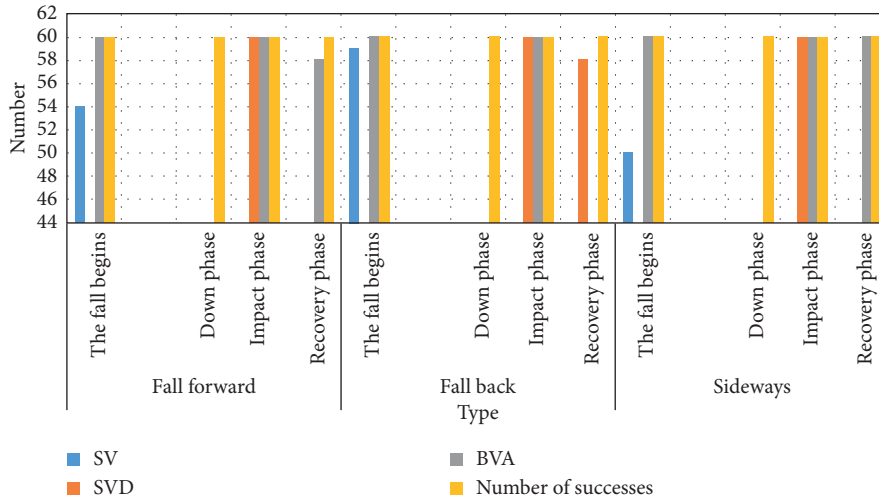


FIGURE 1: Performance of this algorithm in fall.

100%. The multifunctional parameters under multiple sensors can significantly improve the measurement accuracy.

4.2. Probability of Each Parameter Exceeding the Limit in Daily Activities. In the daily activities of the algorithm, the probability of each parameter exceeding the limit value is shown in Table 3 and Figure 2.

From Table 3 and Figure 2, we can see that the probability of each parameter in the algorithm exceeds the limit in daily activities; then a group of daily activities is equivalent to 10 falls in this study, and the false detection rate S_{ij} of each characteristic parameter is calculated, as shown in formula (10).

$$S_{ij} = \frac{N_{ij}}{10 \times 10} \times 100\%. \quad (12)$$

Among them, i = fall start, fall, impact, and recovery; j = SV, BVA, and SVD; N_{ij} is the number of false detections. As a result of 6 times of false detection, S_{ij} is 6% and SP value is 94%. It can be seen from Table 2 that the false alarm rate of acceleration related characteristic parameters is higher than that of angle. SV has the highest false alarm rate, while BVA has low false alarm rate. This is because BVA is the dynamic quantity of SV, and the daily activity changes slowly. Therefore, it is appropriate to increase the weight of BVA parameters in daily behavior assessment. Combined with Sections 4.1 and 4.2, we have falls and daily activities data of

TABLE 3: Performance of this algorithm in daily behavior.

| Action | | Characteristic parameters | | | |
|----------------|-----------------|---------------------------|---------|---------|-------------------------|
| | | SV (%) | SVD (%) | BVA (%) | Number of successes (%) |
| Daily behavior | The fall begins | 13 | | 2 | 13 |
| | Down phase | | 3 | 1 | 2 |
| | Impact phase | | 1 | 8 | 5 |

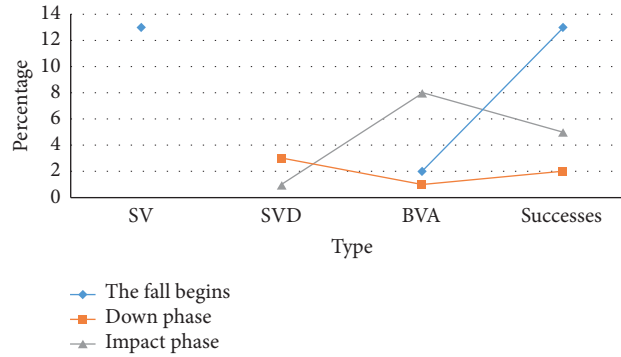


FIGURE 2: Performance of this algorithm in daily behavior.

100 volunteers to verify the algorithm, with a sensitivity of 96.67% and a specificity of 97%.

4.3. Multisensor Fall Detection Based on Support Vector Machine. Through the 4.1 data acquisition experiment, four kinds of human posture data are obtained, and five kinds of posture features are extracted. These features reflect different points from different angles. According to the human classification of different postures, a fall recognition model based on support vector machine (SVM) algorithm is proposed by controlling the corresponding optimal parameters. The specific steps are as follows:

- (1) Human body attitude sensors (including accelerometers and gyroscopes) are used for data acquisition, so as to collect multiple human body data at the same time, including human body triaxial acceleration and angular velocity.
- (2) Preprocess the collected original data, using the average filtering method to filter the human posture data, in order to obtain better and clearer data.
- (3) The features are derived from all the processed kinds of human body position data (sitting, lying, falling, sliding, squatting down, walking, standing, and going upstairs and downstairs). The features of different positions are taken, including five characteristics of combined acceleration and vertical component of combined acceleration.
- (4) In order to build the model based on the derived human pose set, ten times grid crossover and optimization algorithm should be used to determine the best training parameters. To evaluate the model, introduce the test set data to train the autumn attitude recognition model, and calculate the accuracy of the model in the test set.

According to the above steps, 86 groups of features are divided into training set and test set. After determining the optimal parameters, each dataset is randomly mixed and verified five times to ensure that the model has good generalization ability. The experiment was divided into three groups, and the total number of experiments in each group was different, but the number of falls was the same. The numbers of training sets and test sets and the identification results are shown in Figure 3.

As can be seen from the results in Figure 3, when the training set contains different data samples, the recognition results are slightly different. When the training set contains 65 samples and 8 groups of falls, the recognition rate should be 82%. In the training process, the set contains 70 samples, of which 8 groups are falls, and the recognition rate is 86.23%. When the training set contains 75 samples, of which 8 groups are falls, the recognition rate should be as high as 92.15%, and the recognition rate will improve with the increase of the number of samples in the selected education set, which indicates that the recognition ability of the trained model will gradually improve based on a large amount of effective data, thus increasing the possibility of prediction and prevention of falls.

4.4. Comparative Analysis and Selection of Models. In order to compare the applicability of support vector machine and random forest algorithm, two fall detection models were created and compared. 60 datasets are selected as the training set, including 50 normal positions and 10 falling positions. On the basis of the model, the training time of the model and the recognition time of the fall posture are calculated. The results are shown in Figure 4.

As can be seen from Figure 4, the training time of the model based on random forest is 5.5 seconds, the recognition time is 1.45 seconds, and the recognition rate is

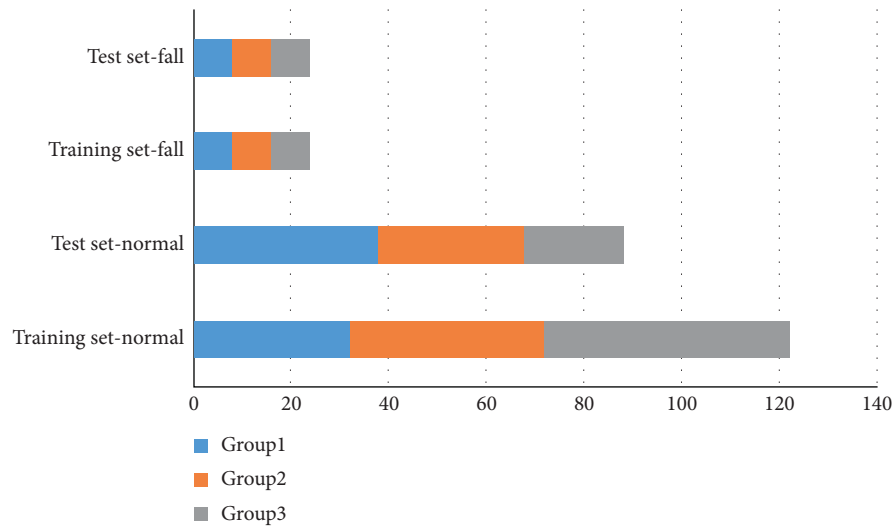


FIGURE 3: Comparison of fall posture recognition results.

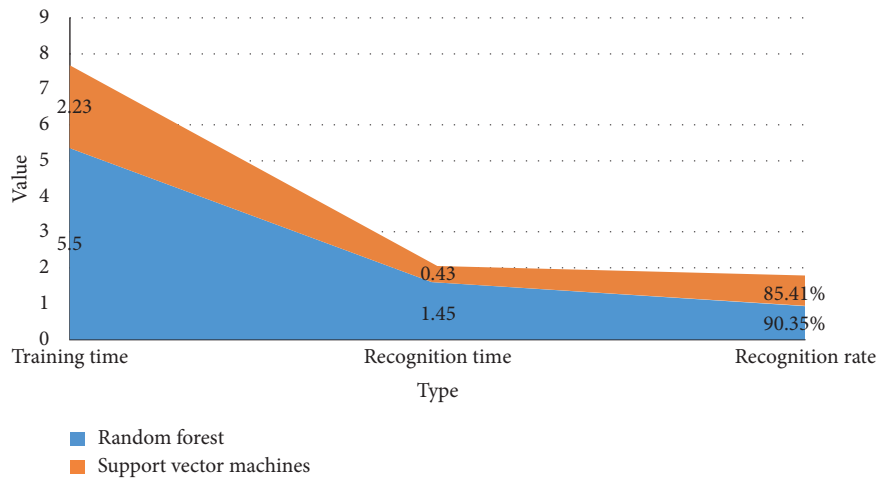


FIGURE 4: Comparison of the recognition results of the two algorithms.

90.35%; the training time of fall posture recognition model based on support vector machine is 2.23 seconds, the recognition time is 0.43 seconds, and the recognition rate is 85.41%. It can be seen that this machine can make more time-based response to fall recognition than the time-vector-based recognition algorithm.

5. Conclusion

Fall is a common phenomenon in the life of the elderly, and it is also one of the main causes of disease and death of the elderly. The real-time fall detection system can trigger the alarm in time, reduce the waiting time for treatment, and reduce the accidental injury caused by falls. However, in most collision detection systems, only accelerometers are used to design the detection system. However, the algorithm based on single data cannot fully describe the information about body posture changes when falling down. The ultimate goal of this study is to predict the human body's downward trend sensor based on multidata synthesis fall detection

algorithm, so as to effectively detect the occurrence of falls and avoid falls.

In this paper, a data acquisition experiment is designed to preprocess and extract features of human posture data. Then, on the basis of support vector machine algorithm and random forest algorithm, the human posture is recognized, and finally the effective recognition of human posture is realized. In this study, 100 volunteers are collected to simulate falls and daily activities to verify the effectiveness of the algorithm. The sensitivity was 96.67%, and the specificity was 97%, indicating good performance. In the experiment, when the training set contains the largest number of samples and the numbers of fall processing experiments are equal, the recognition rate should reach the maximum value of 92.15%. The accuracy of the model is based on a large amount of valid data. After training, the recognition ability of the system will gradually improve, so as to improve the ability to predict falls. Support vector machine has less training time and recognition time, which brings more warning time and is conducive to prevention. Therefore,

SVM is more suitable for the system development based on multisensor data fusion.

A fall detection algorithm based on multisensor data synthesis is studied in this paper to improve the accuracy of fall prediction. Wearable devices are very suitable for the elderly and have a very broad application prospect. This study is of great significance to its development.

Data Availability

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

Conflicts of Interest

None of the authors have any conflicts of interest.

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (863 Program) under Grant 2013AA01A215.

References

- [1] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "RT-fall: a real-time and contactless fall detection system with commodity WiFi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2017.
- [2] J. He, C. Hu, and X. Wang, "A smart device enabled system for autonomous fall detection and alert," *International Journal of Distributed Sensor Networks*, vol. 12, no. 2, Article ID 2308183, 2016.
- [3] P. V. Thanh, D. T. Tran, D. C. Nguyen et al., "Development of a real-time, simple and high-accuracy fall detection system for elderly using 3-DOF accelerometers," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3329–3342, 2019.
- [4] M. Kepski and B. Kwolek, "Event-driven system for fall detection using body-worn accelerometer and depth sensor," *IET Computer Vision*, vol. 12, no. 1, pp. 48–58, 2018.
- [5] S. Amir, Z. Liang, Z. Wang et al., "Novel flexible wearable sensor materials and signal processing for vital sign and human activity monitoring," *Sensors*, vol. 17, no. 7, p. 1622, 2017.
- [6] H. Chen, W. Chen, S. Bao et al., "Design of an integrated wearable multi-sensor platform based on flexible materials for neonatal monitoring," *IEEE Access*, vol. 8, pp. 23732–23747, 2020.
- [7] A. Shishido and N. Akamatsu, "Mechanical properties of flexible materials," *Kobunshi*, vol. 66, no. 7, pp. 362–365, 2017.
- [8] G. Li, T. Liu, J. Yi, H. Wang, J. Li, and Y. Inoue, "The lower limbs kinematics analysis by wearable sensor shoes," *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2627–2638, 2016.
- [9] F. Lin, A. Wang, Z. Yan, R. T. Machiko, and X. Wenyao, "A wearable sensor device for unobtrusive gait monitoring in daily life," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2281–2291, 2017.
- [10] C. Raphael, P. Philipp, and B. Anne-Laure, "Random forest versus logistic regression: a large-scale benchmark experiment," *Bmc Bioinformatics*, vol. 19, no. 1, p. 270, 2018.
- [11] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.
- [12] X. Huile, L. Jinyi, H. Haibo et al., "Wearable sensor-based human activity recognition method with multi-features extracted from hilbert-huang transform," *Sensors*, vol. 16, no. 12, p. 2048, 2016.
- [13] S. R. Steinhubl, D. Feye, A. C. Levine et al., "Validation of a portable, deployable system for continuous vital sign monitoring using a multiparametric wearable sensor and personalised analytics in an Ebola treatment centre," *Bmj Global Health*, vol. 1, no. 1, Article ID e000070, 2016.
- [14] I. Carpinella, D. Cattaneo, G. Bonora et al., "Wearable sensor-based biofeedback training for balance and gait in Parkinson's disease: a pilot randomized controlled trial," *Archives of Physical Medicine & Rehabilitation*, vol. 98, no. 4, pp. 622–630, 2016.
- [15] Y. Lei, X. Daxi, G. Liquan et al., "A compressed sensing-based wearable sensor network for quantitative assessment of stroke patients," *Sensors*, vol. 16, no. 2, p. 202, 2016.
- [16] W. Liangtian, H. Guangjie, W. Hao et al., "Wearable sensor localization considering mixed distributed sources in health monitoring systems," *Sensors*, vol. 16, no. 3, p. 368, 2016.
- [17] O. Martin, D. Joe, W. Tomas et al., "Mobile app to streamline the development of wearable sensor-based exercise biofeedback systems: system development and evaluation," *Jmir Rehabilitation & Assistive Technologies*, vol. 4, no. 2, p. e9, 2017.
- [18] H. Lee, H. Chung, H. Ko et al., "Dedicated cardiac rehabilitation wearable sensor and its clinical potential," *PLoS One*, vol. 12, no. 10, Article ID e0187108, 2017.
- [19] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2017.
- [20] Z. Obermeyer and E. J. Emanuel, "Predicting the future-big data, machine learning, and clinical medicine," *New England Journal of Medicine*, vol. 375, no. 13, pp. 1216–1219, 2016.
- [21] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.
- [22] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.
- [23] Z.-M. Yang, H.-J. Wu, C.-N. Li, and Y.-H. Shao, "Least squares recursive projection twin support vector machine for multi-class classification," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 3, pp. 411–426, 2016.
- [24] Y. Yuan, "Canonical duality solution for alternating support vector machine," *Journal of Industrial & Management Optimization*, vol. 8, no. 3, pp. 611–621, 2017.