Eng Int Syst (2021) 6: 395–401

Engineering
Intelligent Systems

Performance Analysis of Cloud Computing Resource Scheduling Optimization Based on IPSO Algorithm

Wu Chunqiong*

Yango University, Fuzhou Fujian 350015, China

With the emergence of cloud computing technology, users and researchers are constantly exploring ways to use network technology to simulate intelligent system operations, introduce more practical application functions, and assist with the design of cloud computing platforms. The design and development of cloud computing platforms require a sound knowledge of computer science and engineering, although several theories have been advanced by research scholars in this field. Cloud computing network platforms have been able to provide an increasing number of data resources, and the number of users who use cloud computing network platforms for data processing is also increasing. In the long-term, the current structure of the cloud computing network platform will no longer be able to meet users' requirements for a good quality network platform. Based on the analysis of the particle swarm algorithm, this article addresses the current problems related to the use of the algorithm, and adjusts the bad particles and parameters in a particle swarm so that the algorithm can better meet the construction needs of a cloud computing platform.

Keywords: IPSO algorithm; Cloud computing; Resource scheduling; Optimized performance

1. INTRODUCTION

Advancements in computer technology have ushered in the era of 'big data'. People are increasingly inclined to use data to analyze various problems encountered in teaching, and to construct different educational data based on big data analysis. There is the need for a model that can provide support and guarantee the improvement of specific research activities. By examining the statistics of different education data and analysis results produced by related models, the relationship between different variables in these activities and the intensity of mutual influence can be determined. A certain degree of efficiency in data processing and analysis can be achieved by using computers for some data processing tasks and constructing a certain data analysis method. People's requirements for data processing technology are constantly changing, leading to the rapid rise of cloud computing technology in the network data processing process [1]. This technology can effectively combine virtual technology and real network technology to effectively analyze and process large amounts of data. During the operation of the cloud computing platform, the development and use of virtualization technology provides many opportunities for the development of cloud computing. In addition, the development and application of Internet technology is also the foundation for the long-term development of cloud computing technology. When using virtualization technology, it is important to determine the carrying capacity of the cloud computing platform to ensure the effective use of the technology, which is also the main research aim of most researchers. When using the resources of a cloud computing platform, people

*Email: wu_chunqiong@163.com

do not need to pay additional fees to purchase the life of the platform. People can use the platform for simulation at any time for the purpose of data analysis. By analyzing the distribution of resources in cloud computing platforms, we can determine the problems related to this distribution [2]. In order to improve the user experience, resources should be paid for as this will improve the effectiveness of resource allocation. At present, the resources in the cloud computing platform are being utilized to an unreasonable extent, which wastes of resources, increases the burden on the normal operation of the system, and negatively affects the user experience. Long-term development is very unfavorable.

With the continuous advancement of science and technology, cloud computing technology is now being utilized in an increasing number of domains including medicine and e-commerce, not to mention the many other fields where computer technology has become an inextricable part of people's daily lives. The application of cloud computing technology provides people with more comprehensive data analysis and evaluation, making it more convenient for people to choose safe services in universities [3]. The current methods used for resource allocation and scheduling in cloud computing cannot meet the quality of service that users demand of cloud computing technology, and it is also difficult to determine whether cloud computing technology will be unbalance by an overloaded of operations. In order to improve user satisfaction with the quality of service offered by cloud computing technology and the improvement of the efficiency of the cloud computing platform, it is necessary to strengthen the technical capabilities of cloud computing in resource scheduling and allocation to make it more efficient and scientific. Therefore, the study of cloud computing Dynamic resource scheduling has become a hot issue. The resource scheduling of cloud computing is mainly for the purpose of meeting the service needs of users. It is necessary to continuously optimize resource scheduling to effectively improve overall revenue and efficiency [4]. In order to meet the above requirements, it is necessary to strengthen the effectiveness, timeliness and scientific nature of cloud computing in resource scheduling, so that cloud computing can integrate many resources and meet the needs of users in a timely manner. In order to improve the resource allocation speed of cloud computing, it is necessary to use more efficient and scientific cloud computing resource scheduling algorithms, so as to effectively improve the service level of cloud computing. Given users' requirements for service quality, this paper proposes a cloud computing resource scheduling algorithm based on the particle swarm algorithm, which can improve the scheduling of cloud computing resources, and address resource-allocation problems.

2. IPSO ALGORITHM DESIGN

2.1 The Basis of The IPSO Algorithm

In 1995, inspired by the foraging behavior of birds, researchers proposed a particle swarm optimization algorithm, one of the intelligent optimization methods. In the simulation of the IPSO algorithm, the foraging behavior of a group of birds is viewed as a group of particles without mass and volume flying in the search space to find the best position [5]. The search for the best food position can be regarded as the process of solving the target problem, and each particle represents a potential solution to the problem being solved. In the initial state, all particles are assigned a random position x and velocity v. In each generation, the position of the particle will be determined by the local optimal position of the group particle and the global optimal position in the memory of the group particle.

The position of the k-th particle can be expressed as:

$$X_k = (X_{k,1}, X_{k,2}, X_{k,3} \ldots X_{k,N}), k \in [1, m] \tag{1}$$

The velocity of the k-th particle can be expressed as:

$$V_k = (V_{k,1}, V_{k,2}, V_{k,3} \ldots V_{k,N}), k \in [1, m], V_k \in [-V_{\max}, V_{\max}] \tag{2}$$

When particles are moving in the air, they may be affected by inertia, which will change their direction and speed. To a certain extent, the inertia weight directly determines whether the particles can find the optimal position. In order to eliminate the influence of too many factors, researchers believe that a linearly decreasing strategy can be used to reduce the influence of inertia weight on particle movement. The calculation formula is:

$$P^k_{best} = (P_{k1}, P_{k2}, P_{k3} \ldots P_{kN}), k = 1, 2, 3 \ldots M \tag{3}$$

The overall optimal solution can be expressed as:

$$G^k_{best} = (P_{g1}, P_{g2}, P_{g3} \ldots P_{gN}) \tag{4}$$

The particle swarm algorithm has several advantages: it converges quickly and has a relatively small number of parameters, making it is easy to grasp the results of the calculation. When using the differential evolution algorithm, it is found that when the calculation process runs to the later stage, the diversity of species in the population will not change, and the order between the crossed parameters can be guaranteed [6]. The application of this algorithm is more complicated, as it needs to calculate the motion states of various particles. Secondly, the algorithm does not directly give the specific position of the particle during the calculation process, but defines a region with the most probable position as the center, which also fits the uncertainty of the particle's movement in the quantum domain. The updated speed and position change of particles can be expressed as:

$$V_{ki} = \eta V_{ki} + \alpha\lambda(P_{kn} - V_{kn}) + \beta\mu(P_{gn} - V_{gn}) \tag{5}$$

$$X_{kn} = X_{kn} + V_{gn} \tag{6}$$

2.2 Implementation of IPSO Algorithm

2.2.1 The Implementation Process of Improved Particle Swarm Algorithm

In this study, we analyze the shortcomings of the original particle swarm algorithm and, subsequently, eliminate several bad particles from the particle swarm. Calculations are adjusted according to the parameters in the formula. The steps used to improve the particle swarm algorithm are shown in Figure 1 below.

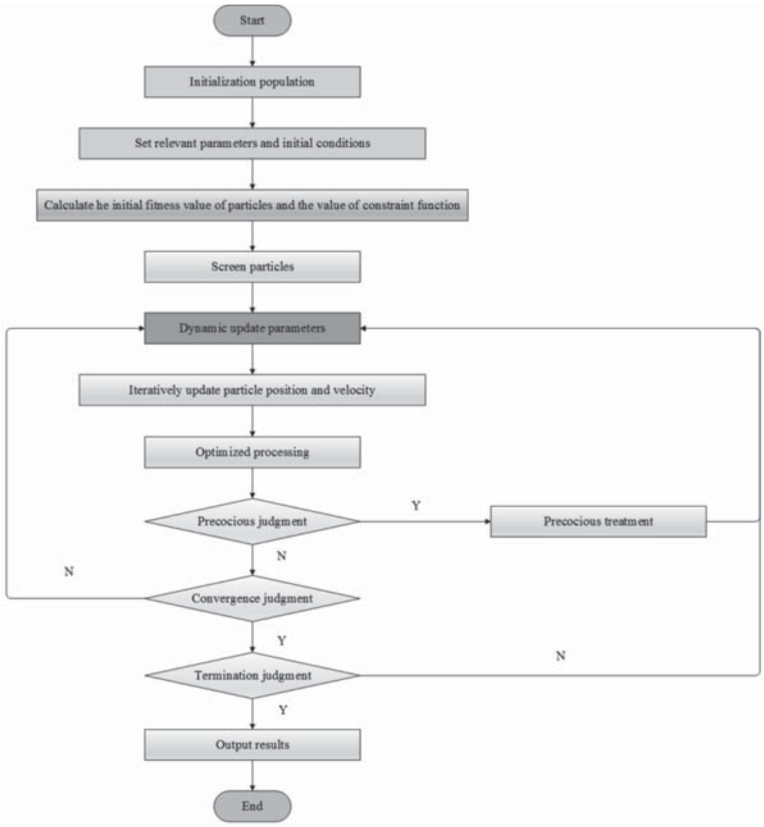

Figure 1 Flow chart of improved particle swarm algorithm implementation.

2.2.2 Performance Analysis of Improved Particle Swarm Algorithm

For analysis, it is necessary to determine the performance of the improved particle swarm algorithm and of the basic particle swarm algorithm [7]. Hence, this research compares and analyzes the solution of the objective function in the original particle swarm algorithm and the optimal solution obtained by the improved particle swarm algorithm by establishing an objective function. Simulation software is used to simulate and analyze the efficiency of the two particle swarm algorithms.

The objective function F1 is expressed as:

$$F1\colon \min\ f_1(x) = \sum_{i=1}^{30} x_i^2/4000 - \prod_{i=1}^{20} \cos\left(\frac{x}{\sqrt{i}}\right) + 1,\ x_i \in [-600600] \tag{7}$$

The objective function F2 is expressed as:

$$F2\colon \min\ f_2(x) = 0.5 - \frac{\left(\sin\left(\sqrt{x_1^2+x_2^2}\right)\right)^2 - 0.5}{1+(x_1^2+x_2^2)/1000},\quad x_1, x_2 \in [-100100] \tag{8}$$

The test results are shown in Table 1 below.

The data presented in Table 1 and Table 2 shows that the improved particle swarm algorithm has better performance than the basic particle swarm algorithm in solving the optimal solution of the objective function.

The data presented in Table 3 and Figure 3 above shows that the 10 experiments conducted in this study used the first objective function to eliminate a total of 60 particles in the example swarm algorithm, and the second objective function eliminated a total of 62 particles. The efficiency of the function in eliminating particles is basically the same, which to a certain extent shows that the elimination system proposed in this study is more reasonable [8]. The results indicated that the application of the objective function to eliminate some bad particles can indeed reduce the calculation range of the particle swarm algorithm and improve the accuracy of the calculation.

3. DESIGN AND OPTIMIZATION OF CLOUD COMPUTING RESOURCE SCHEDULING SYSTEM BASED ON IPSO ALGORITHM

3.1 The Implementation Process of Improved Particle Swarm Algorithm in Cloud Computing Resource Scheduling

According to the structural adjustment and the functional relationship established by cloud computing for resource scheduling optimization, the steps of particle swarm algorithm for cloud computing resource scheduling adjustment are as follows: The first step is to take the user's service demand as the objective function, and then using the particle swarm algorithm for cloud computing resource scheduling, search

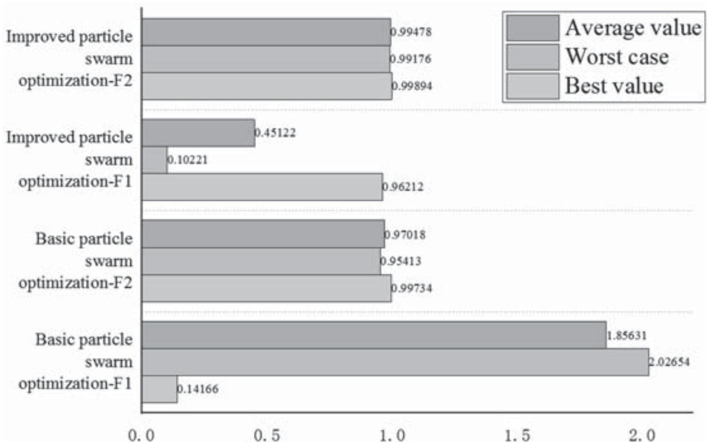

Figure 2 Comparison of results before and after particle swarm optimization.

Table 1 Basic particle swarm algorithm calculation results.

Function	The optimal value	Worst value	Average value	Average number of iterations of convergence
F1	0.14166	2.02654	1.85631	928
F2	0.99734	0.95413	0.97018	1000

Table 2 Improved particle swarm algorithm calculation results.

Function	The optimal value	Worst value	Average value	Average number of iterations of convergence
F1	0.96212	0.10221	0.45122	456
F2	0.99894	0.99176	0.99478	413

Table 3 Number of eliminated particles.

Experiment number	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_10
F1	5	6	6	6	6	7	7	6	6	5
F2	6	5	6	7	7	7	6	6	6	6

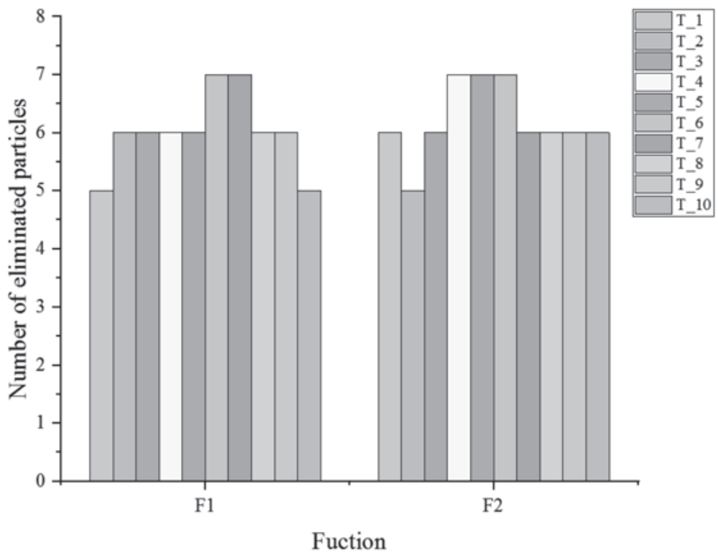

Figure 3 Comparison of the number of eliminated particles.

for relevant particles and make adjustments; the second step is to determine the problems that occur in cloud computing resource scheduling, and initialize these particles so that the particles after initialization are optimal; the third step is to use the function to calculate the initial fitness of the particles, and eliminate the inconsistent particles; in the fourth step, update the existing particle swarm so that it can optimize the individual and the whole; the fifth step determines whether it is premature, and return to the previous one if required [9]. Otherwise, go to the next step which determines whether

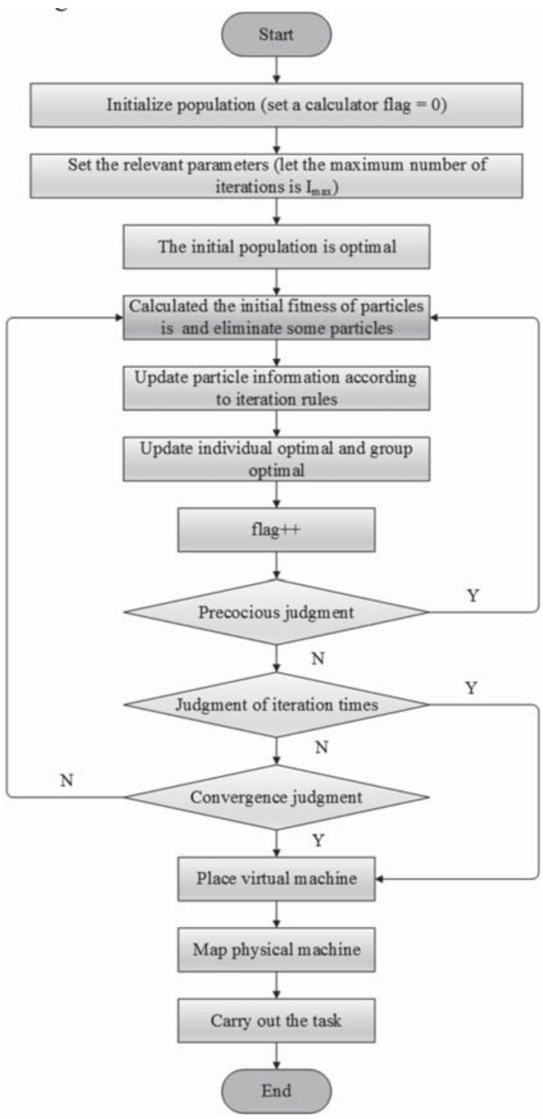

Figure 4 The implementation process of improved particle swarm algorithm in cloud computing resource scheduling.

the cloud computing resource scheduling has the maximum number of generations, if yes, go to the sixth step, otherwise go to the next step; the seventh step is to determine whether the cloud computing resource scheduling has converged, if yes, go to the first The fourth step; the eighth step is to know the particle information of the optimal solution; the ninth step is to perform simulation tests based on the value of the optimal solution; the tenth step is to learn the machine being used according to the registration information of the resource; the eleventh step It is to recover resources after the machine completes the task; the twelfth step is to integrate the above process and get the result [10]. The steps required for cloud computing resource scheduling are depicted by the flowchart in Figure 4 below.

3.2 Design of Simulation Experiment System

3.2.1 Simulation Experiment System

The software and toolkits used to build a simulation platform in this study are listed in Table 4.

To conduct an in-depth analysis, it is necessary to check the classified data. For the purpose of analysis, a network platform is built to simulate the data running environment [11]. It has been found that the Cloud Sim platform is the main platform suitable for data operation and simulation, and the efficiency of data simulation and simulation can be improved through this platform. Users do not need to pay extra fees to purchase the life of the platform; the platform can be used for simulation at any time for data analysis. Moreover, in this study, this platform was chosen for the construction of the simulation environment mainly because its source code can be used and referenced by others. Users can obtain the simulation results in the shortest time [12]. Furthermore, it provides users with wired and wireless usage models, offers users more principles, and facilitates users' data analysis and testing.

To a certain extent, CloudSim can reduce the time required to build the platform as this type of simulation platform is relatively easy to construct, because it takes into account all the operational characteristics of the cloud computing platform, and can perform the resource allocation of this platform.

Table 4 Software and toolkit required for CloudSim platform.

Tool (software) name	Introduction
JDK1.7	Compiler Environment
C1oudSim2.1	Simulate cloud computing environment
NetBeans7.1	Development tools

Table 5 Experimental hardware environment.

Project	Configuration value
Operating system	Windows10 Home Edition
CPU	Intel Core i7-7500UCPU@2.70GHz
RAM	8GB
Hard disk	256GB
JDK version	JDK1.8
Cloud sin version	Cloudsim3.0.3

Table 6 Experimental parameter settings.

Parameter name	Value
Number of tasks	200, 400, 600, 800, 1000
Number of virtual machines	10
Number of iterations	150

3.3 Experimental Environment Configuration

3.3.1 Experimental Hardware Environment

The simulation experiment for this research needs to be conducted on a personal computer. The configuration is shown in Table 5 below.

If a personal computer is being used to build an experimental simulation platform, firstly, CloudSim must be downloaded from the Internet. It is best to choose the version with the largest number of users, and then download the compressed file of the corresponding version on the official website to the computer [13]. The compressed file needs to be decompressed first, and an appropriate decompression path (which cannot include Chinese characters) should be selected during the decompression process. After the compressed file has been decompressed, the environment of the simulation platform can be established, and appropriate variables can be chosen according to the requirements of the experimental analysis.

The steps are:

(1) Initialize C1oudSim, set the number of users of the simulation platform and the completion time of the simulation task according to the needs of experimental analysis, and then the simulation can be carried out after determining all the variables;

(2) Create a new data processing center and management agent in the platform;

(3) To determine the situation of virtual machines, specify the number of virtual machines and the memory and external storage of virtual machines are too small, and the parameters of the agent ID and virtual machine ID that accept virtual machine services are set;

(4) Create specific tasks according to the needs of the experiment;

(5) Map the created task to the virtual machine;

(6) Start the formal simulation operation and obtain the final calculation results;

(7) End the entire simulation process.

3.4 Experimental Results and Analysis

In order to illustrate the performance of the proposed algorithm through experimental verification, the basic particle swarm optimization algorithm and the adaptive twin-swarm improved particle swarm algorithm are applied to cloud computing task scheduling for comparison and analysis.

3.4.1 The Impact of The Number of Tasks on Performance

The number of simulation tasks in this study ranged from 200 to 1,000. The specific conditions and experimental data of the virtual machine are shown in Table 6 below.

3.4.2 The Impact of Resource Quantity on Performance

In the case of the same number of tasks, the number of virtual machines is increased from 10 to 50. Through specific experiments and analysis, we know that two different algorithms will formulate different strategies for the resource allocation of cloud computing platforms. The specific parameter changes are shown in Table 7 below.

The laws of motion of particles in the quantum field are quite different from those of objects in real life. The two significant laws of particle motion in the quantum field are aggregation and uncertainty. The uncertainty of particle motion is due to quantum motion. They are often jumpy, so it is generally difficult to determine the precise path of motion

Table 7 Experimental parameter settings.

Parameter name	Value
Number of tasks	1000
Number of virtual machines	10, 20, 30, 40, 50
Number of iterations	150

[14–15]. We can only obtain the probability function of different paths according to the law of particle motion. In the particle swarm optimization algorithm, researchers generally use the substitution method to see the whole particle as comprising multiple examples. Then, the motion equations of these particles are listed according to the different parameters, properties and trajectories of these wholes and then, according to the known particle position and moving image, the particle's movement law and the next position are predicted [16]. In the prediction process, the prediction result is affected by the position of the particle, as well as its velocity and acceleration during movement.

4. CONCLUSION

Cloud computing has many shortcomings in terms of resource scheduling, so we should start from the cloud computing resource scheduling algorithm, optimize and improve the particle swarm algorithm, and design a brand-new cloud computing resource scheduling model. The particle swarm algorithm and cloud computing resource scheduling model are used for the analysis of the user's service quality requirements. At the same time, it is not necessary to consider whether the cloud computing technology will be out of balance under overload conditions. Finally, the results are obtained through continuous experimentation. The improved and optimized particle swarm algorithm has a positive effect on cloud computing resource scheduling as it can improve performance. It is necessary to improve and expand the simulation platform of cloud computing. At the same time, it is necessary to compare the role of the basic particle swarm optimization algorithm and the adaptive dual subswarm improved particle swarm algorithm in cloud computing task scheduling. The experimental results of this study show that the new algorithm proposed in this paper can optimize and upgrade cloud computing resource scheduling to a certain extent, and significantly improve scheduling efficiency.

ACKNOWLEDGEMENTS

Engineering Research Center for Business Intelligence in Big Data for Fujian Province, project document number 90 of Education and Scientific Research of the Education Department of Fujian Province in 2018.

REFERENCES

1. T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 7 (2002), 971–987.
2. Y. Ouyang, N. Sang, A facial expression recognition method by fusing multiple sparse representation-based classifiers. In: International Symposium on Neural Networks (2013), 479–488.
3. O.M. Parkhi, A. Vedaldi, A. Zisserman, et al., Deep face recognition. bmvc 1 (2015), 6.
4. D. Patel, X. Hong, G. Zhao, Selective deep features for micro-expression recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR) (2016), 2258–2263.
5. M. Peng, C. Wang, T. Chen, G. Liu, X. Fu, Dual temporal scale convolutional neural network for micro-expression recognition. Front. Psychol. 8 (2017), 1745.
6. T. Pfister, X. Li, G. Zhao, M. Pietikäinen, Recognising spontaneous facial micro-expressions. In: 2011 International Conference on Computer Vision (2011), 1449–1456.
7. M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, Computer Vision Using Local Binary Patterns, vol. 40. Springer Science & Business Media, Berlin (2011), 712–731.
8. S. Polikovsky, Y. Kameda, Y. Ohta, Facial micro-expressions recognition using high speed camera and 3d-gradient descriptor. IET (2009), 824–836.
9. F. Qu, S.J. Wang, W.J. Yan, et al., Cas(me)2: a database for spontaneous macro-expression and micro-expression spotting and recognition. IEEE Trans. Affect. Comput. 9(4) (2017), 424–436.
10. S.P.T. Reddy, S.T. Karri, S.R. Dubey, S. Mukherjee, Spontaneous facial micro-expression recognition using 3d spatiotemporal convolutional neural networks. arXiv preprint arXiv 19(4) (2019), 1390–1412.
11. M. Shreve, S. Godavarthy, D. Goldgof, S. Sarkar, Macro-and micro-expression spotting in long videos using spatio-temporal strain. In: Face and Gesture (2011), 51–56.
12. P.Y. Simard, D. Steinkraus, J.C. Platt, et al., Best practices for convolutional neural networks applied to visual document analysis. In: Icdar, vol. 3 (2003), 249–253.
13. Y. Song, L.P. Morency, R. Davis, Learning a sparse codebook of facial and body microexpressions for emotion recognition. In: Proceedings of the 15th ACM on International Conference on Multimodal Interaction (2013), 237–244.
14. C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015), 1–9.
15. M. Takalkar, M. Xu, Q. Wu, Z. Chaczko, A survey: facial micro-expression recognition. Multimedia Tools and Applications 77(15) (2018), 19301–19325.
16. M.A. Takalkar, M. Xu, Image based facial micro-expression recognition using deep learning on small datasets. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (2017), 1–7.