IJBPAS, November, Special Issue, 2021, 10(11): 913-923

ISSN: 2277–4998

International Journal of Biology, Pharmacy and Allied Sciences (IJBPAS)

'A Bridge Between Laboratory and Reader'

www.ijbpas.com

SDN NETWORK LOAD BALANCING USING ENVIRONMENTAL CONGENITAL ACO METHODOLOGY

SRINIVAS JHADE[1*], SAKTHIVEL S[2], SUDHA R V[3], ROHIT KUMAR VERMA[4], RANJAN WALIA[5] AND LOKESH M R[6]

1: Associate Professor in Computer Science Engineering at KG Reddy College of Engineering & Technology, Chilkur (V), Moinabad (M), Ranga Reddy Dist. Hyderabad, Telangana State, India

2: Professor in Computer Science and Engineering at Sona College of Technology, Salem, Tamil Nadu, India

3: Assistant Professor, Vivekanandha College of Technology for Women, Elayampalayam, Tiruchengode, Namakkal Dt, Tamilnadu, India

4: Assistant Professor, Department of MCA, Himachal Pradesh University Regional Centre, Mohli, Khaniyara, Dharamshala-176218, District Kangra, Himachal Pradesh, India

5: Associate Professor in Electrical Engineering Department at Model Institute of Engineering and Technology, Jammu, J & K, India

6: Professor at Maharaja Institute of Technology Mysore, Mandya, Srirangapatna, Karnataka, India

***Corresponding Author: Srinivas Jhade; E Mail: srjhade@kgr.ac.in**

Received 22nd July 2021; Revised 25th Aug. 2021; Accepted 30th Sept. 2021; Available online 1st Nov. 2021

https://doi.org/10.31032/IJBPAS/2021/10.11.1079

ABSTRACT

The amount of information transmitted is continually risen while modern communication methods have been developed. To successfully meet that rising computational demands, the volume of datacentres networks (DCNs), which are materials built up of servers connected by sustainable switches, rapidly grown across the globe. Classical switches are ineffective in meeting the demands of DCNs. Over the latest days, the software-defined network (SDN) was already considered as a novel networking standard for managing DCNs, controlling network switches, and deploying new network services. One major issue with DCNs, therefore, was balancing the workload among computers. Machine learning (ML) technologies might be used to handle data transfer requirements as one possible solution to this issue. Deep learning (DL) is a new effective ML approach that generates predictions, classifications, and choices using vast quantities of data. Despite deep learning (DL) is gaining popularity in such a variety of disciplines, has few advantages in networks. Inside this article, a DL method for load-balancing SDN-based DCNs is presented. The different loading levels across connections were used to educate the DL networks. The overall reaction speed of a DL approach for load balancing is contrasted to that of several ML techniques, including an ANN, SVM, and logic regression method. The findings show perhaps ANN & DL methods have faster reaction times than support vector machines & logistic regression methods. Furthermore, DL efficiency is superior to ANN efficiency. As a consequence, DL seems to be an excellent load balancing solution.

Keywords: Machine learning, Load balance, SDN, Deep learning, Switches; Environment

INTRODUCTION

SDN has been used in DCNs to increase networking efficiency in the latest days **[1-2]**. online datacentres are implemented to handle the ports in **[3]**, and complete network management is obtained via SDN and Communication protocol. To satisfy a range of data centers requirements, **[4]** utilized the SDN NOX controllers to administer the DCNs. **[5]** looked at SDN-based DCNs & described the SDN frameworks that were used to simplify network connection and administration. **[6]** looked at data centers architecture that was centered on task scheduling. Scientists utilized data center networking with in-packet bloom filtering and OpenFlow processors that turn DCNs into software bugs. **[7]** suggested a two-tier SDN approach to alleviate the network congestion in DCNs by including dynamic load balance. SDN uses 6 aspects to increase transmitting data efficiency in DCNs. Capacity ratios, number of hops, delay, packet overrun, trustworthiness, and packet drop are some of these properties. SDN-based data traffic balance increases access permissions by effectively spreading the load over various routes. Forwarding congestion via task scheduling in SDN-based DCNs would provide reduced management expenses and stellar results.

Artificial Intelligence approaches emphasize the construction of intelligent robots which operate & react the same way that people do. Machine learning and deep learning are two main subs of Artificial Intelligence **[8]**. Computer learning features an adaptable method that enables computers to learn from instances and knowledge. Over time, this dynamic method is used to improve the Artificial Intelligence system's performance **[9]**. Deep Learning is formed by Machine Learning's multi-layered & multi-node architecture. The framework in Deep Learning trains to execute stratifications from visuals, words, or audio. Deep Learning produces high-performance outcomes in big datasets and has effectively been used to problems in object recognition analysis **[10]**. As a consequence, network scientists began to recognize the value of Deep learning and to investigate how it may be used to address networking issues. Since traffic updates in datacentres are nearly identical and the network structure is relatively stable, it is necessary to utilize Deep Learning methods across sender and receiver **[11]**. The reason for this study would be the first use of DL techniques in SDN-based DCNs. Researchers intend to investigate the use of DL techniques to redistribute loads in DCNs through studying the various network parameters.

Related Work

In **[12]** presents ant colony optimization (ACO) based load balance problems. To distribute load among the source to the destination point, the suggested scheme utilizes the ACO study guideline as well as the connection task scheduling technique. The network demand, latency, and packet drop were utilized as task scheduling variables to pick the next burden. In **[13]**, and adaptable isolated ant method for SDN transportation planning is presented, as well as a task scheduling method for transportation planning in SDN (A4SDN). Ants are employed in the A4SDN algorithms to discover sustenance throughout all possible pathways. The A4SDN computation result is evaluated to Dijkstra's algorithm. In comparison to Dijkstra's algorithm, A4SDN exhibited a 55 percent higher throughput performance. A4SDN is satisfied in terms of enhancing independent routing protocol, the helpful approach of network capacity, & enhanced connection speeds.

In **[14]**, researchers suggest a Q-Learn technique to task scheduling in SDN to limit the number of dissatisfied customers towards a 5G wireless connection. The Q learn method directs a participant's traffic to the access point which provides the highest incentive instead of the lowest mean value again for the justice criterion. To equalize the network congestion and maintain QoS, **[15]** includes a multi-particle swarm optimization method. To save energy, the multi-particle swarm optimization algorithm takes the most easily open and switches to the rest state whenever needed. **[16]** proposes a heuristics solution to the switching relocation issue.

To enhance the capacity of SDN, **[17]** presented a novel adaptive routing system that blends the evolutionary algorithms with ACO. In respect of a good way to develop searching performance, round-trip duration, and packet drop, the suggested method considerably outperforms the Round Robin and ACO algorithms. Such researches are focused on the use of optimization methods. Task scheduling is an essential method that necessitates real-time evaluation & adaptive processing of traffic data. Once taught, ANN and DL constitute important approaches for dynamic load management. As a result, the DL method is presented in this work to dynamically distribute the load.

Proposed methods

The k-ary fat tree28 design is a traditional architecture that arranges transistors into a tree shape are seen in **Figure 2**.

• From bottom to top, the k ary fat tree possesses 3 rigid foundational tiers: edges, aggregates, and core.

• The computers are accessible in the aggregating and core levels, as well as the edges layer is connected to the main computer.

• The web server at every layer as well as the overall physical servers that may communicate to clients are decided using the fat tree's k attribute values.

AI is a computing model aimed at teaching robots how to operate, respond, and adapt in the same way that people do. ML and DL methods are used in the AI sector. ML is a machine learning algorithm that learns from information and makes choices without being pattern recognition. In classification methods, LR, Artificial NN, and Deep Learning are widely utilized ML methods.

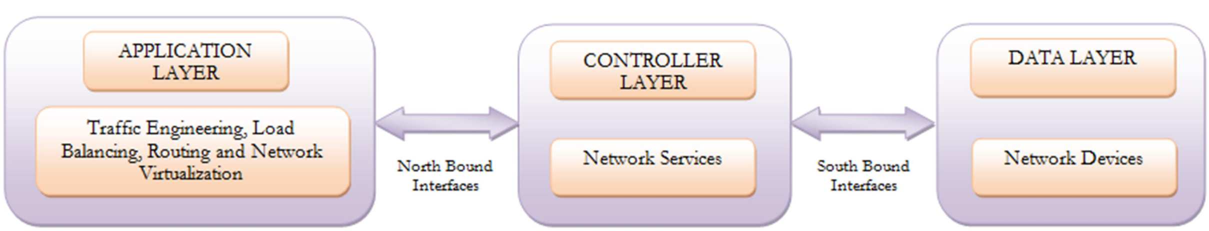

Figure 1: The generic SDN architecture

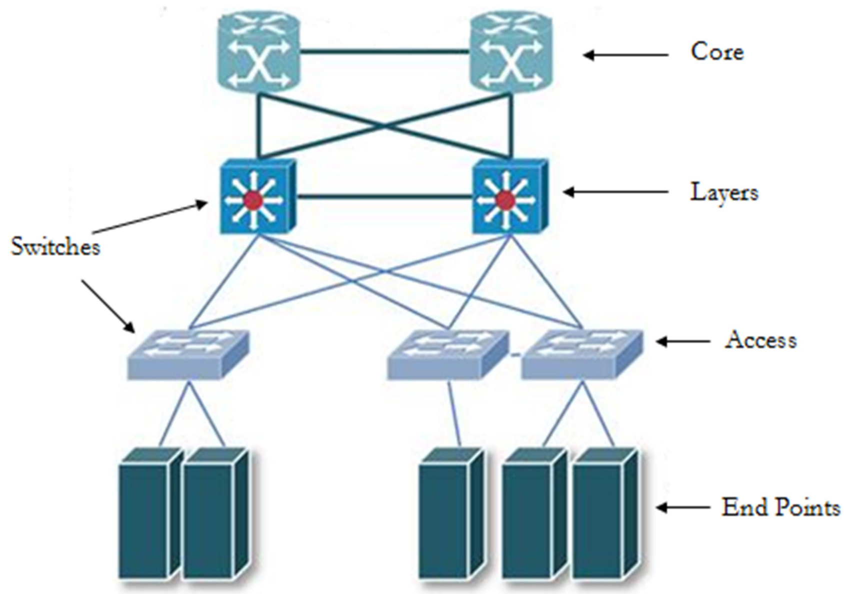

Figure 2: Structure of k ary in DC network

SVM31,32 is built on the LR technique that assigns training images to one of 2 groups, separated by distinct ways hyper-plane. The optimum hyper-plane, unlike LR, abuts the locations that indicate the greatest separation of the two sets. The SVM structure cannot be defined by a simple method. As a result, SVM is charged with locating the datasets that define the hyper-plane and derivatives lines parameters, and then categorization the incoming data variables as lying on either side of the hyperplane. ANN is a machine learning approach that is based on natural nerve networks. A node in ANN correlates to neurons in the biological nervous system. The weights represent the importance of the data collected in the nodes as well as its impact on the network. It makes a difference to ANN whether the mass is higher or lower. ANN methods use multi-layer modifications to learn representations and hierarchically retrieve information from raw data, making assumptions based on the goal of employing several levels of asymmetric processing elements. DL33,34 is a type of machine learning that employs concatenated levels to collect features from input information and then make a judgment. The core idea behind DL is to acquire data representations by layering abstractions on top of one other. Increasingly complex ideas at a greater level are often acquired by describing them in the perspective of less abstract concepts at reduced ranks. That sort of hierarchy method of learning is extremely successful since it enables a machine to comprehend and develop sophisticated structures straight from original data, making it applicable to a wide range of fields.

Several load-balancing techniques are available in SDN: static, dynamically, or combined of both. The forwarding policy is explicitly coded in the load balancer in the static load-balancing approach, however, network performance deteriorates as no such knowledge about the networks is available. Because of burden is redistributed dynamically per some parameters set in the load balancer, dynamic techniques are more effective than static ones.

Architecture and Experimental Results

The majority of the original study modeled assessments were run on a Linux operating system with just an Intel core i10 CPU and 10 Gb of Memory. The DCNs structure is selected as a k-ary fat-tree architecture, with the topology's k constant value system at 5. All deep learning algorithms are developed using Notepad.

Figure 4 depicts the flow graph art for the study's architectures. The k-ary fat-tree architecture is initially a built-in network simulator, as shown in **Figure 3**. Then, for task scheduling, two servers are chosen. Data transmission (tx) and having received rate (rx) are empowered in Floodlight to conduct task scheduling among two decided trails. The price among selected pathways is acquired when the flow production is completed using Iperf. Rest-API is used to obtain route information. The Rest-API is also used to get IP addresses, routers, MAC addresses, and port mapping information. The price between the two decided pathways is calculated in milliseconds and saved in a textual file using this information. Ulu.41 contains this supplementary component. The cost is gathered in Floodlight, and the particular parameters are adjusted to run Rest the API apps. The cost of the ways is calculated by

Cost = tx (transmitted) + rx (received), (5)

wherein tx represents the number sent throughout a data packet, rx represents the value retrieved throughout a data packet, and the price is the total of the tx and rx numbers. The information gathered is utilized to educate machine learning and deep learning methods. The processes are provided with data such as In-Port, Out-Port, Source IP, Destination IP, Source MAC, and Destination MAC. This data is provided every second by the software, which keeps it interactive. After continuously getting the optimum routing path using ML and DL techniques, this optimal route data is employed for task scheduling.

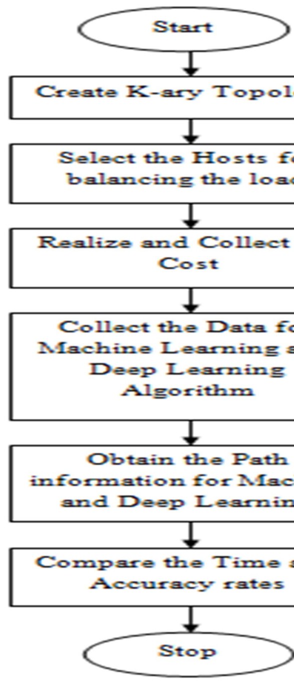

Figure 3: Architecture of the model for the experiments

LR, Support Vector Machine, Artificial NN, and Deep Learning are utilized as machine learning algorithms for training and validation. In the SVM algorithm, the C parameters are set to 1 or 100. A 2 model is used in ANN. There are five joints at the first stage and only one joint at the second stage. 0.0025 is preferred as the training data. The first and second layers, correspondingly, utilize rely and sigmoid convolution layers. The input parameters for the first and second routes are x0 and x1. The nth component of vectors is denoted by lower script n. The activating, weighting and bias amounts are represented by the variables a, w, and b, correspondingly. A four-layer model is utilized in DL. The DL paradigm for task scheduling. The first layer contains 20 nodes, the second layer has 7 nodes, the third layer has 6 joints, and the 4^{th} stage has 1 joint. Most levels except layer four employ ReLu as an activation function. The sigmoid activating function is utilized in the fourth layer. In the ANN model, x0 and x1 are the input parameters. A number linked with the lth layer is superscript **[12]**. The nth entry of vectors is denoted by lower script n. The activating function's inputs, also known as the pre-activation variable, is Z. The reaction rate of each method is computed after the ML algorithms have run as per the testing data.

In machine learning, efficiency assessment is crucial. As a result, the ROC curve is frequently used to assess a categorization problem's effectiveness. The

ROC curve is a probabilistic curve that indicates how well the model results are for distinct classifications. **Figures 4 and 5** show the ROC curves for SVM as C = 1.0 and C = 100.0, Artificial NN, and Deep Learning, correspondingly, for such two situations.

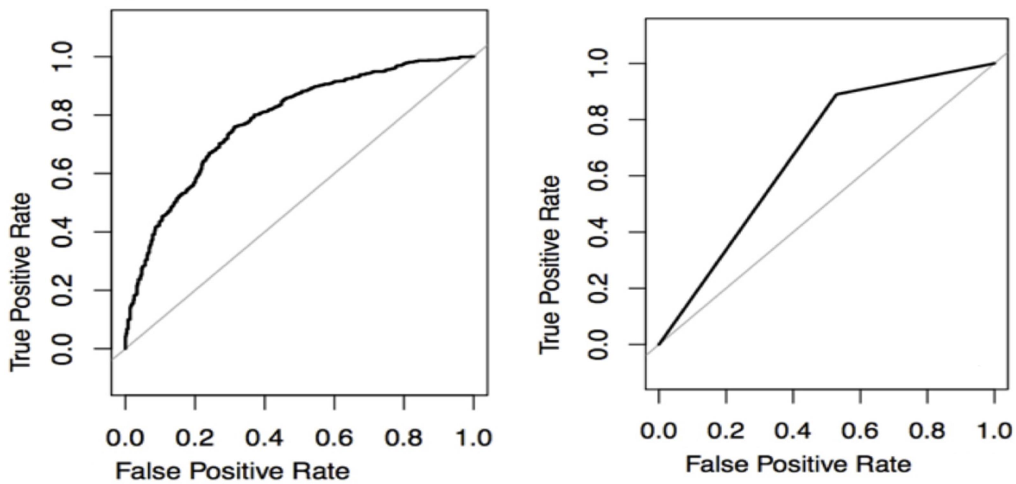

Figure 4: ROC curve for Support Vector Machine and Deep Learning in First Scenario

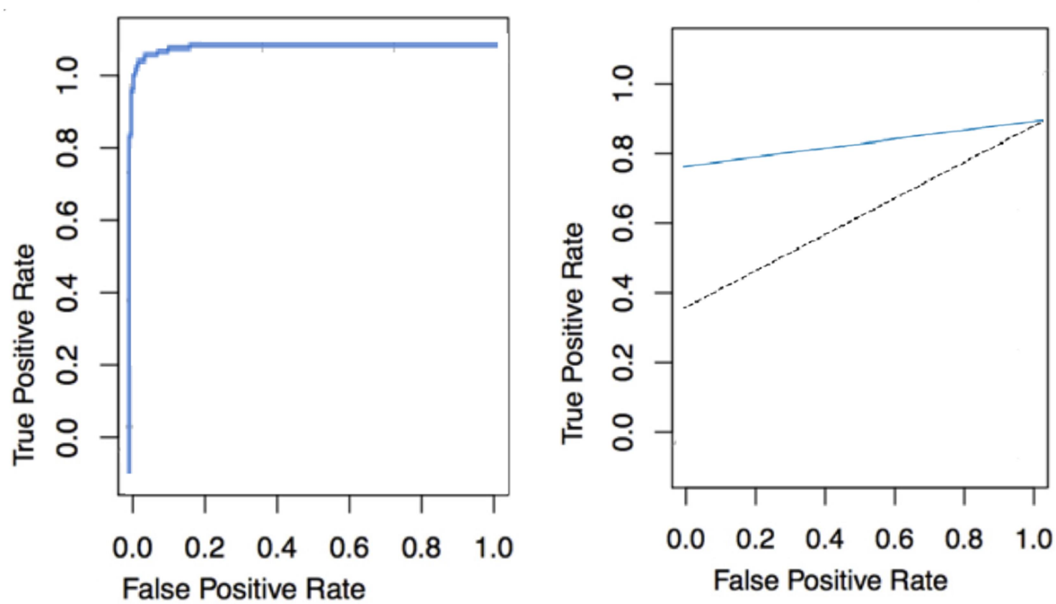

Figure 5: ROC curve for Support Vector Machine and Deep Learning in Second Scenario

By comparing the mean reaction times of the 1st & 2nd situations, Artificial NN and Deep Learning algorithms require less time than the other Ml algorithms. The ANN method is the classifier with the best mean reaction time of the two situations; nevertheless, the DL method is fairly constant in comparison to the ANN model. The LR algorithm comes after the DL method. In this investigation, the SVM method is the classification model with the poorest rate of success. In addition, as contrasted to SVM of C = 1.0, SVM of C = 100.0 has a significantly faster reaction time. In furthermore, the original data is subjected to tenfold cross-validation to measure and evaluate the classification program's efficiency. **Tables 1 and 2** show the classification results of the methods acquired from cross-validation for two different scenarios. The SVM with C = 100 methods is the most accurate classification system of overall accuracy. In the second order, the DL method is accompanied by SVM of C = 100.0 and 1.0, and the ANN method is the classification model with the poorest success efficiency. According to another technique, the DL technique is fairly accurate in terms of both median reaction time & correctness percentages. As a result, among the studied Ml techniques, the DL technique is regarded to become the better model in task scheduling over SDN.

Table 1: The precision rate at the first level

Support Vector Machine E=5	Support Vector Machine E = 50	Artificial Neural Network	Deep Learning
0.85	0.87	0.50	0.71

Table 2: The precision rate at the second level

Support Vector Machine E=5	Support Vector Machine E = 50	Artificial Neural Network	Deep Learning
0.83	0.87	0.56	0.81

CONCLUSION

That is important to achieve and sustain traffic balance between multiple data centers while performing DCNs on SDN. As a result, this research proposed for the first time the use of DL for task scheduling of SDN-supported DCN. On DCNs, DL is implemented by using variable transmission cost values between particular routes. In terms of reaction time and overall accuracy, the efficiency of the DL approach is contrasted to that of Ml techniques such as ANN, SVM, and LR. The findings demonstrate that the suggested DL-based traffic balance strategy outperforms the other ML-based task scheduling techniques. The DL-based task scheduling approach presented in this study would be investigated for all communication networks in SDN in the upcoming, taking into account various performance measures likely the width of the bans, and latency. A software center using low-cost single-board processors also will be built to handle large data. Additionally, single-board processors will be equipped with the SDN controller. This controller will be in charge of controlling flow in the datacenters as well as task scheduling. To choose the optimal route, this acquired data would be educated in SDN controllers using DL and ML techniques.

REFERENCES

[1] Xie, K., Huang, X., Hao, S., Ma, M., Zhang, P., & Hu, D. (2016). $\text {E}^{3} $ MC: Improving Energy Efficiency via Elastic Multi-Controller SDN in Data Center Networks. *IEEE Access*, *4*, 6780-6791.

[2] Mu, T. Y., Al-Fuqaha, A., Shuaib, K., Sallabi, F. M., &Qadir, J. (2018). SDN flow entry management using reinforcement learning. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, *13*(2), 1-23.

[3] Wang, Y. C., & You, S. Y. (2018). An efficient route management framework for load balance and overhead reduction in SDN-based data center networks. *IEEE Transactions on Network and Service Management*, *15*(4), 1422-1434.

[4] Dr. P. Sivakumar, "Analytical framework to build predictive and optimization function from manufacturing industry sensor data using cross-sectional sharing", Big Data,2021 (SCI)

[5] Dr.P.Sivakumar, “Improved Resource management and utilization based on a fog-cloud computing system with IoT incorporated with Classifier systems”, Microprocessors and Microsystems, Jan 2021 (SCI).

[6] Ranjeeth, S., Latchoumi, T. P., & Paul, P. V. (2020). Role of gender on academic performance based on different parameters: Data from secondary school education. Data in brief, 29, 105257.

[7] Venkata Pavan, M., Karnan, B., & Latchoumi, T. P. (2021). PLA-Cu reinforced composite filament: Preparation and flexural property printed at different machining conditions. Advanced Composite Materials, https://doi.org/10.1080/09243046.2021, 1918608.

[8] Srivastava, V., & Pandey, R. S. (2021). Load balancing for the software-defined network: a review. *International Journal of Computers and Applications*, 1-14.

[9] Wang, T., Liu, F., Guo, J., & Xu, H. (2016, April). Dynamic SDN controller assignment in data center networks: Stable matching with transfers. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications* (pp. 1-9). IEEE.

[10] Gershman, A. B., Sidiropoulos, N. D., Shahbazpanahi, S., Bengtsson, M., & Ottersten, B. (2010). Convex optimization-based beamforming. *IEEE Signal Processing Magazine*, *27*(3), 62-75.

[11] Choudhary, S., &Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD, and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science*, *167*, 1561-1573.

[12] Dorigo, M., Birattari, M., &Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39.

[13] Rout, S., Patra, S. S., Sahoo, B., & Jena, A. K. (2017, July). Load balancing in SDN using effective traffic engineering method. In *2017 International Conference on Signal Processing and Communication (ICSPC)* (pp. 452-456). IEEE.

[14] Tosounidis, V., Pavlidis, G., &Sakellariou, I. (2020, September). Deep Q-Learning for Load Balancing Traffic in SDN Networks. In *11th Hellenic Conference on Artificial Intelligence* (pp. 135-143).

[15] Trivedi, V., Varshney, P., &Ramteke, M. (2020). A simplified multi-objective particle swarm optimization algorithm. *Swarm Intelligence*, *14*(2), 83-116.

[16] Tripathi, P. K., Bandyopadhyay, S., & Pal, S. K. (2007). Multi-objective particle swarm optimization with time-variant inertia and acceleration coefficients. *Information scie-nces*, *177*(22), 5033-5049.

[17] Dai, Y., Lou, Y., & Lu, X. (2015, August). A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing. In *2015 7th international conference on intelligent human-machine systems and cybernetics* (Vol. 2, pp. 428-431). IEEE.