Hybrid Particle Swarm Optimization with Firefly based Resource Provisioning Technique for Data Fusion Fog-Cloud Computing Platforms

Joseph B. Awotunde[1,*], Hrudaya K. Tripathy[2], Anjan Bandyopadhyay[3]

[1] Faculty of Information and Communication Sciences, University of Ilorin, Nigeria

[2] School of Computer Engineering, Kalinga Institute of Industrial Technology, India

[3] Kalinga Institute of Industrial Technology (KIIIT) Bhubaneswar, Odisha, India

Emails: awotunde.jb@unilorin.edu.ng; hktripathyfcs@kiit.ac.in; anjan.bandyopadhyayfcs@kiit.ac.in

Abstract

The recent wide acceptance of cloud and virtualization technologies has made a number of Internet of Things (IoT) applications practical. Although these technologies are typically useful, they may introduce a high transmission latency in IoT environments, e.g., data fusion in smart cities. To address this issue, fog computing, a distributed decentralized computing layer between IoT hardware and the cloud layer, can be used. To facilitate the use of fog computing in IoT data fusion environments, this paper proposes a new Hybrid Particle Swarm Optimization with Firefly based Resource Provisioning Technique (HPSOFF-RPT) model for fog-cloud computing platforms. The HPSOFF-RPT model is designed to optimize resource allocation and distribution in IoT environments. The model uses the PSO and FF algorithms to provision resources in the fog-cloud environment. To evaluate performance, a wide-ranging simulation analysis is performed. The simulation results show that the proposed model improves performance compared to the existing optimization algorithms.

Keywords: Resource provisioning; Fog computing; Cloud computing; Hybrid metaheuristics; Data Fusion

1. Introduction

Cloud Computing is the use of computing resources and services over external networks, e.g., the Internet. The resources and services are typically offered by a third party, known as a cloud provider. As a result, resources data fusion is a big challenge for these environments. Upon the approval of a cloud computing request, the provider typically assigns several virtual machines (VMs) to the requestor [2].

Cloud provisioning is the allocation of cloud assets to a user [1]. This can be done in advance (i.e., static provisioning) or pay-as-you-go (i.e., dynamic provisioning). In static provisioning, the requestor is charged in advance (e.g., each month) as a number of assets are reserved for the requestor [2]. In dynamic provisioning, the assets are allocated to a requestor only when they are needed [3]. Unlike static provisioning, dynamic provisioning can optimize resource allocation and minimize costs. However, it may lead to resource overprovision (through allocating more than is necessary) or resource underprovision (through allocating less than is necessary). Hence, when dynamic allocation is used, resource overprovision and underprovision should be considered in the design of allocation methods.

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

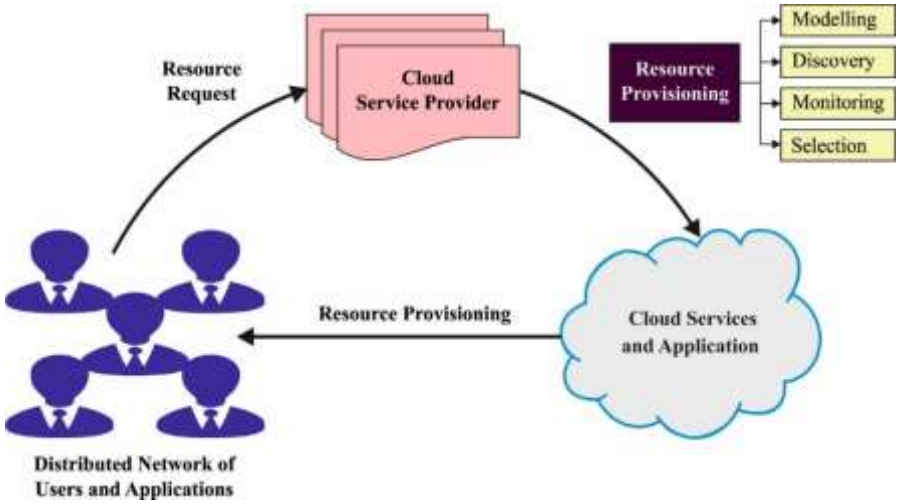

Figure 1: General resource provisioning process

As shown in Figure 1, the general resource provisioning process may go through a number of steps. It typically starts with a resource request and ends with resource provisioning. To maintain scalability, cloud providers often use auto-scaling procedures [5]. These procedures can be proactive and/or reactive. They scale allocation according to resource specifications (e.g., processing capability) and/or service specifications (e.g., service throughput) to improve performance. Resource/service-hosting devices often monitor usage and initiate scaling whenever it is needed. For example, the devices may reallocate resources when usage exceeds the allowance.

A major issue with responsive components is the reaction duration (the time needed for a component to react from trigger detection to asset allocation or reallocation), as it may overburden the system, especially when high fluctuations occur. Proactive components attempt to predict future asset requirements using a variety of methods (e.g., previous allocation records). Although most cloud providers use receptive models, a number of proactive methods have been proposed. These methods use time reinforcement learning, sequence investigation, queuing theory, or control theory [6].

To enhance the reaction duration and optimize resource allocation and distribution, this paper introduces a new Hybrid Particle Swarm Optimization with Firefly based Resource Provisioning Technique (HPSOFF-RPT) model for fog-Cloud Computing platforms. The model uses of the PSO and FF algorithms to provision resources.Appropriate classification of resources can be attained in which the needs of the client undergo mapping with the resources exist in the class. For accomplishing entire resource provisioning process, an easy weight matching process is utilized. For assuring the enhanced performance of the HPSOFF-RPT model, a wide ranging simulation analysis is performed.

2. Related Works

Khorsand et al. [11] offer a self-learning fuzzy technique for proactive resource provisioning in CC atmosphere, whereas key is predicted parameters of the probability distribution of incoming players in every time. Moreover, a self-learning fuzzy autoscaling decision-maker method was suggested for computing the appropriate quantity of sources to be allotted to every tier in the enormously multi-player online games through application of user SLA and predicted workload. In [12], RLPAS (Reinforcement Learning related Proactive Auto-Scaler) method was suggested, and it was depending on prevailing RL-SARSA method which studies the atmosphere in parallel and allots the resources. Ghobaei-Arani et al. [13] present an autonomous resource-provisioning structure for Massively Multiplayer Online Games (MMOGs) in a Cloud atmosphere. At first, a load prediction service expects the forthcoming game-entity distribution from historical trace data by making use of ANFIS prediction model. Afterward, the suggested a fuzzy DT method for estimating the appropriate quantity of sources to be allotted to every tier in an MMOG application by making use of user SLA and predicted workload.

Shahidinejad et al. [14] introduce a hybrid solution for managing the source provisioning problem with the help of workload analysis in a cloud atmosphere. This solution used the K-means and Imperialist Competition Algorithm (ICA) to cluster the workload given by users. Moreover, uses a DT technique for determining scaling decisions for effective resource provisioning. Ghobaei-Arani et al. [15] suggest a hybrid resource provisioning technique for cloud services which depends on a combination of the idea

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

of reinforcement learning (RL) and autonomic computing. Besides, provides a structure for autonomic source provisioning that was inspired by the cloud layer method. In [16], uses an enhanced Virtual Machine (VM) sharing protocol for optimizing VM consumption and achieving particular Quality of Service conditions from a diversity of users in WaaS cloud platforms. It introduces a Flexible Deadline-driven scheduling and resource Provisioning approach for Multiple workflows (FDPM) which could reduce the computing expenditures by adjusting VM sharing for making the workflow's execution cost low while attaining a user assigned deadline.

3. The HPSOFF-RPT Model

The HPSOFF-RPT model is proposed to optimize resource allocation and distribution in the data fusion applications. This section describes the HPSOFF-RPT algorithm and its applications.

3.1 The HPSOFF Algorithm

The HPSOFF-RPT model uses the PSO and FF algorithms to provision resources in the fog-cloud environment.

Kennedy and Eberhart in 1995 [17] first suggested a particle swarm optimization (PSO) approach. This model is made by the motivation of fishes grouping and birds flocking. Indeed, they utilized the model of bird flocking to resolve optimization problems. It implies that a collection of particles find the solution space for the optimal solution. Every particle has a velocity, position, and memory to store the optimal location at the initial stage. In every iteration, the particle which has the optimal location is considered the leader, and the remaining particle tends to obtain its location. Hence, the movement is influenced by 2 factors: the optimal location from the initial iteration to existing iteration and the leader location. Eqs (8) and (9) describes how particle moves through iteration:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot rand \cdot \left(p_{best}^{d} - x_{id}^{t}\right) + c_2 . rand\left(p_{gbest}^{d} - x_{id}^{t}\right), \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t}. \quad (2)$$

Here, v_{id} denotes the $d\text{-}th$ dimensions of the velocity of $i\text{-}th$ particles, x indicates the location of the particles, t represents the iteration count, c_1 and c_2 represents learning factors, $rand$ denotes a positive random number lies among zero and one in standard distribution, w denotes the inertia weight coefficient, p_{best} represents the optimal location of the particles at first to existing iterations, and p_{gbest} illustrates the location of the leader in all the iterations.

In this work, the PSO algorithm is combined with the FF algorithm to improve its performance. Yang [24] introduced a FF algorithm. The presented algorithm makes use of attractiveness and brightness of fireflies. Brightness is defined using the objective function values, whereas attractiveness depends on the brightness and it can be formulated by the subsequent equation [24], where $I(x)$ denotes attractiveness and $f(x)$ represents the value of objective function at position x. Note that the x in the existing section shouldn't be mistaken for the representation in the preceding subsection.

$$I(x) = \{\frac{1}{f(x)} \text{ '} if\ f(x) > 0\ 1 + |f(x)|\ , otherwise \quad (3)$$

If the distance from the light source increases, then the attractiveness of firefly reduces [24]:

$$I(r) = \frac{I_0}{1 + \gamma r^2} \quad (4)$$

In Eq. (4), $I(r)$ denotes light intensity at r distance, I_0 represents the light intensity at the source. The light is partially absorbed by their surroundings, the FA employs the γ variable that indicates the light absorption coefficient. The cumulative impact of the inverse square law for distance and the γ coefficient is estimated by the subsequent Gaussian formulation [24]:

$$I(r) = I_0 \cdot \quad (5)$$

Furthermore, every individual firefly exploits attractiveness β that is directly proportional to the light intensity and depends on the distance as follows.

$$\beta(r) = \beta_0 \cdot e^{-\gamma r^2} \quad (6)$$

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

In Eq. (6), variable β_0 defines attractiveness at distance $r = 0$. Note that, Eq. (6) is frequently replaced with Eq. (7) [24]:

$$\beta(r) = \frac{\beta_0}{1 + \gamma r^2} \qquad (7)$$

The FA search formula for a i random individual moves in $t + 1$ iterations to a novel position x_i to individual j with higher fitness as follows [24]:

$$x_i^{t+1} = x_i^t + \beta_0 \cdot e^{-\gamma r_i^2}, i(x_j^t - x_i^t) + \alpha^t(\kappa - 0.5) \qquad (8)$$

Let α be the randomization variable, the arbitrary number derived from a uniform or Gaussian distribution is represented by κ, and $r_{i,j}$ characterizes the distance between t i and j fireflies. Typical value establishes successful outcome for most problems for β_0 and α are 1 and [0, 1], correspondingly.

$r_{i,j}$ denotes the Cartesian distance that is evaluated as follows.

$$r_{i,j} = ||_{X_i - \chi_j}|| = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2} \qquad (9)$$

In Eq. (14), D indicates the problem parameter.

3.2 Application of HPSOFF for Resource Provisioning Process

In the beginning, consider n task and m resource as a set of tasks $T = \{t_1, t_2, t_3, \cdots, t_n\}$, and set of the fog resources $R = \{r_1, r_2, r_3, \cdots, r_m\}$, and moderate provision of resources was achieved according to certain provisioning principles. In comparison to other methods, the task and resource approaches are formulated as follows: The set of tasks given by user is controlled with n fog task. The $p\text{-}th$ tasks are offered by T_p, and feature is determined by $1D$ vector $T_p = \{t_{id}, t_{len}, t_{comp}, t_{netw}, t_{stor}, t_{dat}\}$; whereby t_{id} characterizes a task value; t_{len} symbolizes a task length; t_{comp} is a computation power; t_{netw} is a bandwidth limit; t_{stor} is a memory requirement of the resource; and t_{dat} indicates that dataset must be calculated using the task.

In fog calculation, external resources must be allotted by the virtualized resource. By assuming that m resource is contemporary is a set of fog resources, the $q\text{-}th$ resources are denoted by R_q, and feature is determined by $1D$ vector $R_q = \{r_{id}, r_{comp}, r_{netw}, r_{stor}\}$. Here, r_{id} symbolizes a resource value; r_{comp} is computation energy; r_{netw} is bandwidth usage; and r_{stor} is memory capacity of the resource, correspondingly.

For reporting the entity connection, it is given by a resource provisioning network architecture in fog computation.

When the fog resource was separated, the resource scale of a provisioning method was constrained. User needs must be categorized by different classes. The appropriate resource category was discovered whereby user requirement was mapped to the resource from a class. To finish the provision of resources, it is employed by the weight matching and it is evaluated as follows.

$$grade = \frac{\sum \|req_p - res_p\|\omega_p}{\sum \omega_p} \qquad (10)$$

In Eq. (10), req_p denotes attribute of user requirement, res_p demonstrates the attribute, and ω_p denotes the weight.

Many applications were deployed with different resource requirements to serve the data fusion applications. In the event of diverse tasks, it is classified by storage, computation, and bandwidth. Using the task with three features conquers a different weight. Therefore, the attributes needed by the user were collectively computed, and maximal score is obtained by the user as experimental results of resource provision. The procedure included in FCM-FPA algorithm shown below.

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

Algorithm 1: Pseudocode of FCM-FPA

Input: set of resources $\{res_1, res_2, \ldots, res_m\}$, set of tasks $\{task_1, task_2, \ldots, task_n\}$, m, ε, N

Output: Grade, matching outcomes

actual processing of data

attains the normalized matrixes R, T

carry out $(N, K) \rightarrow u$ and c

$$[rs] = size(R)$$

u =psofcm

while $t < M$ do

$$t \leftarrow t + 1$$
$$u' \leftarrow u$$

evaluate u_m and v

for$k = 1 \rightarrow K$ do

for $p = 1 \rightarrow r$ do

dist $(k, p) = Dist\big(data(p, :), v(k, :)\big)$

end

end

evaluate u' and objfun

if norm($u' - u$, inf) $< \varepsilon$ then

break

end if

$$u \leftarrow u'$$

end while

attains the three resource classes

evaluate grade (req_p, res_p, ω_p)

$$grade = sum\,(norm\,(req_p - res_p) * \omega_p)/sum(\omega_p)$$

return matching outcomes M: $\{R_p \rightarrow T_q\}$

4. Experimental Validation

The performance evaluation of the HPSOFF-RPT model is performed using IRIS and WINE datasets. The details related to the IRIS dataset are portrayed in Table 1 with 150 samples and three class labels.

Table 1: IRIS dataset details

IRIS Dataset		
Description	**Class**	**No. of Samples**

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

Class 0	Iris-setosa	50
Class 1	Iris-versicolor	50
Class 2	Iris-virginica	50
Total No. of Samples		**150**

Fig. 2 illustrates the confusion matrix offered by the HPSOFF-RPT model on iris dataset. The figure indicated that the HPSOFF-RPT model has identified 44 samples under class 0, 48 samples under class 1, and 47 samples under class 2.

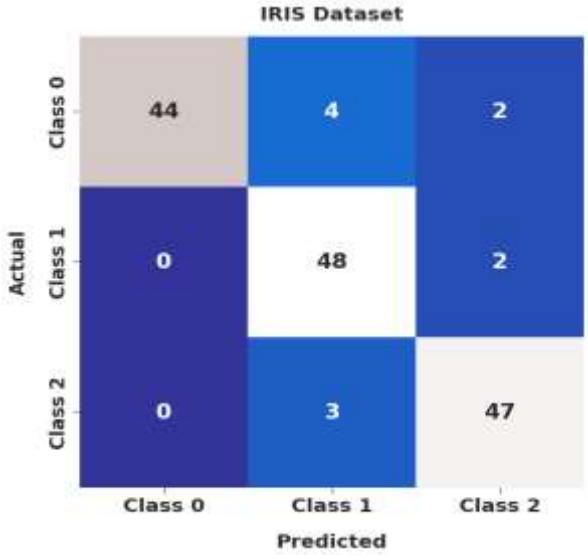

Figure 3: Classifier outcomes of HPSOFF-RPT model on IRIS dataset

Table 2 and Fig. 3 offer brief classifier results of the HPSOFF-RPT model on the IRIS dataset. The results reported that the HPSOFF-RPT model has shown enhanced results under all classes. For instance, on class 0, the HPSOFF-RPT model has reported $accu_y$ and F_{score} of 96% and 93.62% respectively. Also, on class 1, the HPSOFF-RPT model has reported $accu_y$ and F_{score} of 94% and 91.43% respectively. Moreover, on class 2, the HPSOFF-RPT model has gained $accu_y$ and F_{score} of 95.33% and 93.07% respectively.

Table 2: Result analysis of HPSOFF-RPT model on IRIS dataset

Labels	**Accuracy**	**F-Score**
Class 0	96.00	93.62
Class 1	94.00	91.43
Class 2	95.33	93.07
Average	**95.11**	**92.70**

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

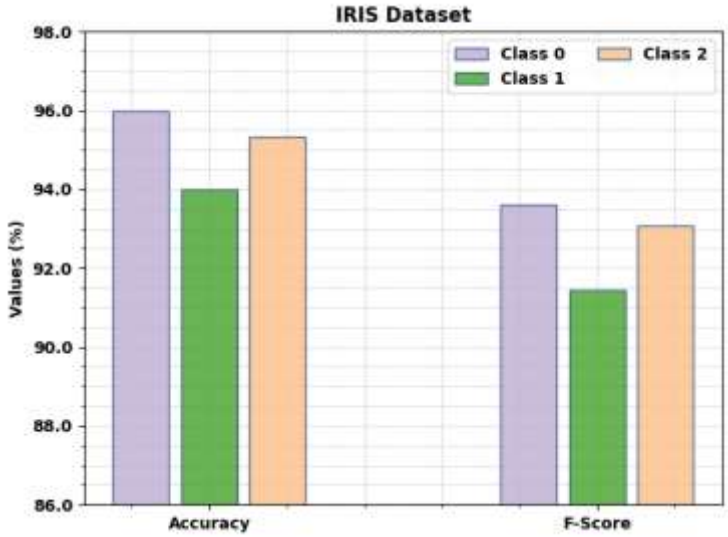

Figure 3: Classifier outcomes of HPSOFF-RPT model on IRIS dataset

Table 3 and Fig. 4 illustrate the comparison study of the HPSOFF-RPT model with other models on IRIS dataset. With respect to correctly clustered (CC), the HPSOFF-RPT model has provided higher CC of 139 whereas the FCM, FCAP, and FCM-FPA models have shown lower CC of 134, 137, and 138 respectively. Besides, with respect to error clustered (EC), the HPSOFF-RPT model has obtained decreased EC of 11 whereas the FCM, FCAP, and FCM-FPA models have reposted increased EC of 16, 13, and 12 respectively. Along with that, based on accuracy, the HPSOFF-RPT model has provided higher accuracy of 95.11% whereas the FCM, FCAP, and FCM-FPA models have demonstrated reduced accuracy of 89.39%, 92.09%, and 94.41% respectively.

Table 3: Comparison study of HPSOFF-RPT model on IRIS dataset

Methods	Correctly Clustered	Error Clustered	Accuracy (%)
HPSOFF-RPT	139	11	95.11
FCM	134	16	89.39
FCAP	137	13	92.09
FCM-FPA	138	12	94.41

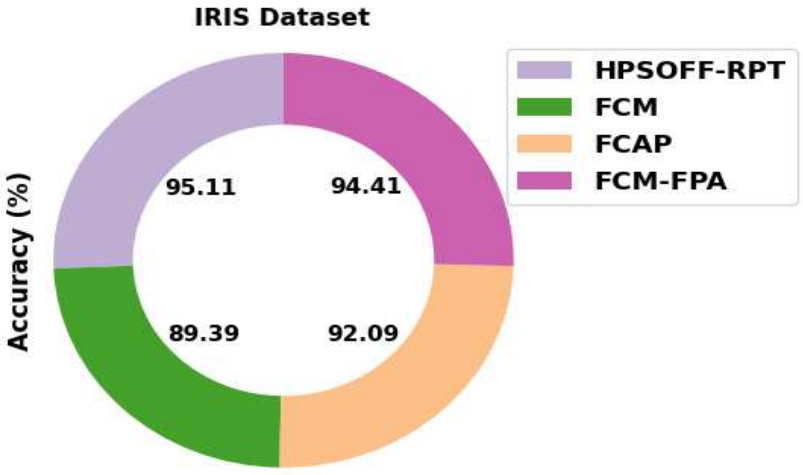

Figure 4: Comparative Classifier outcomes of HPSOFF-RPT model on IRIS dataset

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

Table 4: WINE dataset details

Wine Dataset	
Class	No. of Samples
Class 0	59
Class 1	71
Class 2	48
Total No. of Samples	**178**

The details related to the WINE dataset are portrayed in Table 4 with 178 samples and three class labels.

Fig. 5 demonstrates the confusion matrix presented by the HPSOFF-RPT model on WINE dataset. The figure denoted the HPSOFF-RPT method has identified 40 samples under class 0, 63 samples under class 1, and 39 samples under class 2.

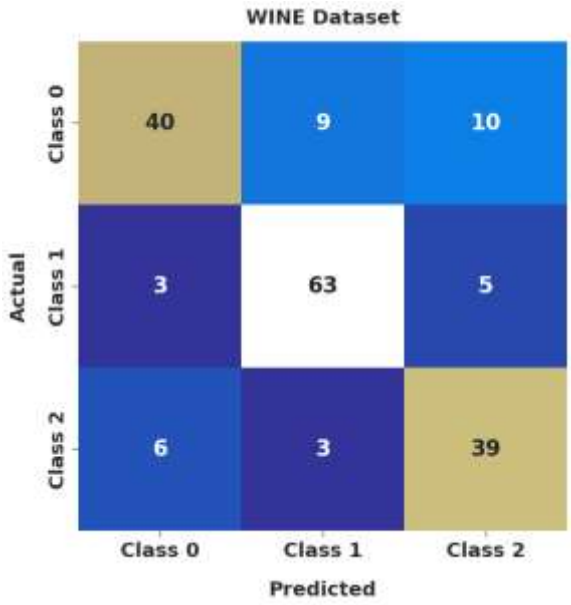

Figure 5: Confusion matrix of HPSOFF-RPT model on WINE dataset

Table 5: Result analysis of HPSOFF-RPT model on WINE dataset

Labels	**Accuracy**	**F-Score**
Class 0	84.27	74.07
Class 1	88.76	86.30
Class 2	86.52	76.47
Average	**86.52**	**78.95**

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

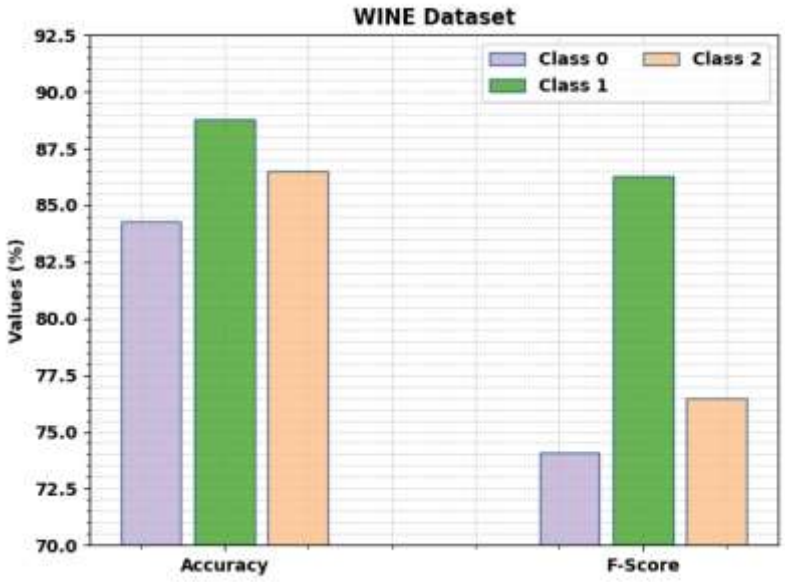

Figure 6: Classifier outcomes of HPSOFF-RPT model on WINE dataset

Table 5 and Fig. 6 offer brief classifier results of the HPSOFF-RPT model on the WINE dataset. The results reported that the HPSOFF-RPT approach has shown enhanced results under all classes. For example, on class 0, the HPSOFF-RPT technique has reported $accu_y$ and F_{score} of 84.27% and 74.07% respectively. Also, on class 1, the HPSOFF-RPT model has reported $accu_y$ and F_{score} of 88.76% and 86.30% respectively. Moreover, on class 2, the HPSOFF-RPT model has gained $accu_y$ and F_{score} of 86.52% and 76.47% correspondingly.

Table 6: Comparison study of HPSOFF-RPT model on WINE dataset

Methods	Correctly Clustered	Error Clustered	Accuracy (%)
HPSOFF-RPT	142	36	86.52
FCM	122	56	68.37
FCAP	129	49	72.43
FCM-FPA	140	38	82.08

Table 6 and Fig. 71 show the comparison study of the HPSOFF-RPT model with other models on WINE dataset. With respect to CC, the HPSOFF-RPT model has provided higher CC of 142 whereas the FCM, FCAP, and FCM-FPA algorithms have shown lower CC of 122, 129, and 140 respectively. Besides, with respect to error clustered (EC), the HPSOFF-RPT model has obtained decreased EC of 36 whereas the FCM, FCAP, and FCM-FPA models have reposted increased EC of 56, 49, and 38 correspondingly. Along with that, based on accuracy, the HPSOFF-RPT model has provided higher accuracy of 86.52% whereas the FCM, FCAP, and FCM-FPA models have established reduced accuracy of 68.37%, 72.43%, and 82.08% correspondingly.

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

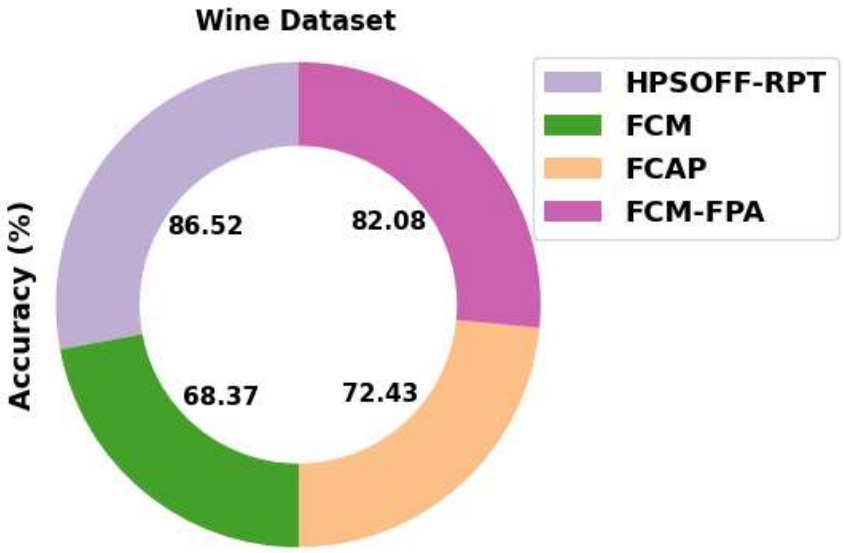

Figure 7: Classifier outcomes of HPSOFF-RPT and recent models on WINE dataset

5. Conclusion

In this study, a new HPSOFF-RPT technique has been proposed to optimize resource allocation and utilization for data fusion applications. The technique uses the PSO and FF algorithms to provision resources in fog-cloud environments. In addition, it uses a weight matching process to facilitate resource allocation and reallocation. The HPSOFF-RPT model has been evaluated using a wide-ranging simulation analysis. The evaluation results show that the HPSOFF-RPT model enhances performance in comparison with other optimization algorithms.

6. References

[1] Duc, T.L., Leiva, R.G., Casari, P. and Östberg, P.O., 2019. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Computing Surveys (CSUR), 52(5), pp.1-39.

[2] Varshney, S., Sandhu, R. and Gupta, P.K., 2019, April. QoS based resource provisioning in cloud computing environment: a technical survey. In International conference on advances in computing and data sciences (pp. 711-723). Springer, Singapore.

[3] Vinothiyalakshmi, P. and Anitha, R., 2021. Efficient dynamic resource provisioning based on credibility in cloud computing. Wireless Networks, 27(3), pp.2217-2229.

[4] Suresh, A. and Varatharajan, R., 2019. Competent resource provisioning and distribution techniques for cloud computing environment. Cluster Computing, 22(5), pp.11039-11046.

[5] Aslanpour, M.S., Dashti, S.E., Ghobaei-Arani, M. and Rahmanian, A.A., 2018. Resource provisioning for cloud applications: a 3-D, provident and flexible approach. The Journal of Supercomputing, 74(12), pp.6470-6501.

[6] Debbi, H., 2021. Modeling and Performance Analysis of Resource Provisioning in Cloud Computing using Probabilistic Model Checking. Informatica, 45(4).

[7] Saxena, D. and Singh, A.K., 2022. OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments. The Journal of Supercomputing, pp.1-22.

[8] Senturk, I.F., Balakrishnan, P., Abu-Doleh, A., Kaya, K., Malluhi, Q. and Çatalyürek, Ü.V., 2018. A resource provisioning framework for bioinformatics applications in multi-cloud environments. Future Generation Computer Systems, 78, pp.379-391.

[9] Shakarami, A., Shakarami, H., Ghobaei-Arani, M., Nikougoftar, E. and Faraji-Mehmandar, M., 2022. Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review. Journal of Systems Architecture, 122, p.102362.

[10] Santos, J., Wauters, T., Volckaert, B. and De Turck, F., 2021. Towards end-to-end resource provisioning in fog computing over low power wide area networks. Journal of Network and Computer Applications, 175, p.102915.

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022

[11] Khorsand, R., Ghobaei-Arani, M. and Ramezanpour, M., 2019. A self-learning fuzzy approach for proactive resource provisioning in cloud environment. Software: Practice and Experience, 49(11), pp.1618-1642.
[12] Bibal Benifa, J.V. and Dejey, D., 2019. Rlpas: Reinforcement learning-based proactive auto-scaler for resource provisioning in cloud environment. Mobile Networks and Applications, 24(4), pp.1348-1363.
[13] Ghobaei-Arani, M., Khorsand, R. and Ramezanpour, M., 2019. An autonomous resource provisioning framework for massively multiplayer online games in cloud environment. Journal of Network and Computer Applications, 142, pp.76-97
[14] Shahidinejad, A., Ghobaei-Arani, M. and Masdari, M., 2021. Resource provisioning using workload clustering in cloud computing environment: a hybrid approach. Cluster Computing, 24(1), pp.319-342
[15] Ghobaei-Arani, M., Jabbehdari, S. and Pourmina, M.A., 2018. An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach. Future Generation Computer Systems, 78, pp.191-210
[16] Rajasekar, P. and Palanichamy, Y., 2022. A flexible deadline-driven resource provisioning and scheduling algorithm for multiple workflows with VM sharing protocol on WaaS-cloud. The Journal of Supercomputing, pp.1-31
[17] Kennedy, J. and Eberhart, R., 1995, November. Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.
[18] Yang, X.S., 2010. Firefly algorithm, stochastic test functions and design optimisation. arXiv preprint arXiv:1003.1409.

Doi: https://doi.org/10.54216/FPA.080203
Received: May 11, 2022 Accepted: September 17, 2022