Adaptive Task Scheduling in Fog Computing Using Learning Automata and RBF Neural Networks for Optimized Performance and Energy Efficiency

Reza Ebrahim Pourian[1], Saeed Roshani[2], Sobhan Roshani[3]

[1]Department of Computer Engineering, Ker.C., Islamic Azad University, Kermanshah, Iran
Email: r.e.pourian@iau.ac.ir

[2]Department of Electrical Engineering, Ker.C., Islamic Azad University, Kermanshah, Iran
Email: s_roshani@iau.ac.ir

[3]Department of Electrical Engineering, Ker.C., Islamic Azad University, Kermanshah, Iran
*Corresponding Author Email: s.roshani@iau.ac.ir

Article History

Received: Sep. 13. 2025

Revised: Oct. 22, 2025

Accepted: Nov. 16, 2025

Abstract

Fog Computing (FC) acts as an intermediate computational layer between the cloud and Internet of Things (IoT) devices, designed to enhance service quality by processing tasks closer to the data source. However, effectively managing energy consumption (EC) remains a critical challenge due to the complexities of task scheduling. This paper proposes an enhanced task scheduling approach based on learning automata (LA) and neural network modeling to minimize fitness, makespan (MK), and associated costs in fog environments. Furthermore, an additional radial basis function (RBF) model is introduced to predict interdependencies among MK, fitness, and cost relative to virtual machine (VM) configurations. A Comparative analysis demonstrates the superior performance of the proposed LA-driven scheduling model over existing methods, achieving more efficient resource allocation and environmental impact reduction across key metrics. This study advances FC task scheduling techniques, highlighting the potential of integrated neural network models to optimize energy-aware computation.

Keywords- Cost, energy consumption, fitness, fog computing, learning automata, Makespan, RBF neural networks.

I. INTRODUCTION

The rapid proliferation of IoT devices and smart systems has led to increased demand for real-time data processing across numerous fields, including healthcare, transportation, and smart cities. The networks generate vast amounts of data, which traditionally require extensive cloud resources for processing [1-3]. However, transmitting data to remote cloud centers incurs latency, bandwidth issues, and energy costs, often compromising the quality of service (QoS) in time-sensitive applications. FC has emerged as an efficient model that processes data closer to the data source, bridging the gap between end devices and the cloud [4-7]. Fog environments introduce complexities in task scheduling, which significantly impact energy usage, MK, and operational costs [8]. With computational power distributed across numerous virtual machines (VMs) in a fog network, an effective scheduling mechanism must balance task allocation to optimize energy consumption, reduce delays, and minimize expenses [9, 10]. Task scheduling in FC is inherently complex and an NP-hard problem, where ineffective scheduling can increase energy demand and carbon emissions due to prolonged machine use. Addressing these scheduling [11, 12] constraints requires intelligent resource allocation that dynamically aligns with real-time processing demands and resource availability [13-16]. Recently, neural networks have emerged as powerful tools for addressing a range of complex problems, including optimization, pattern recognition, and predictive modeling [17-19]. Their ability to learn non-linear relationships makes them suitable for diverse applications, from engineering and data processing to resource management in distributed systems [20, 21]. Recent studies demonstrate the effectiveness of neural networks in optimizing decision-making processes, particularly in dynamic, resource-constrained environments such as fog computing [22, 23]. Learning-based approaches have shown substantial promise for optimizing task scheduling in fog networks [24]. In this paper, we introduce an LA-based scheduling model enhanced by radial basis function (RBF) neural network modeling. This model leverages the RBF network's ability to learn complex data patterns, enabling efficient scheduling decisions that minimize MK and energy consumption. Known for their effectiveness at capturing non-linear relationships, RBF networks enhance the precision of scheduling predictions, enabling more adaptive, resource-conscious task allocation within fog networks. The literature on task scheduling in fog computing has explored various heuristic and learning-based strategies, such as particle swarm optimization (PSO) [25, 26], grey wolf optimizer (GWO) [27, 28], ant colony [29, 30], and bat-inspired algorithm (BIA) [31, 32] to minimize MK and energy cost.

In recent years, various hybrid scheduling techniques have been proposed to optimize task allocation in fog computing environments. However, most existing models focus on specific objectives, such as minimizing makespan or energy consumption, without effectively balancing multiple factors such as cost and performance in dynamic and resource-constrained environments. This paper introduces a novel approach that combines learning automata (LA) with RBF neural networks to address these limitations. The LA component provides adaptive task allocation based on real-time feedback, while the RBF network enhances the model's ability to predict and optimize key parameters such as makespan, energy consumption, and cost. By integrating these two methods, our approach offers a more comprehensive solution for efficient and cost-effective task scheduling in fog computing, filling a crucial gap in existing research. Results show that the proposed algorithm significantly reduces energy consumption, MK, and cost while maintaining low carbon dioxide emissions. Moreover, the RBF network demonstrated strong predictive accuracy for key metrics, enabling the LA-RBF scheduler to make more informed decisions regarding VM allocations.

II. LEARNING AUTOMATA

LA is a reinforcement-based adaptive decision-making model that optimizes actions by learning from environmental feedback. In an LA framework, an automaton selects actions based on a probability distribution, which is adjusted iteratively according to the rewards or penalties it receives from its environment. This adaptability allows LA to operate effectively in dynamic and uncertain conditions, making it suitable for complex scheduling problems like those found in FC. By continuously refining its decision policy through a trial-and-error approach, LA can minimize penalties, which translates to improved performance in computational tasks. This quality is particularly beneficial in FC environments, where the LA model can dynamically allocate tasks to fog nodes, taking into account factors like energy constraints, MK, and available resources to maximize system efficiency. In the proposed scheduling approach, LA plays a pivotal role in guiding task assignments by leveraging a variable-action-set model. This configuration enables the automaton to adaptively manage task scheduling based on resource availability and specific task requirements. Additionally, the LA-based scheduler optimally integrates with neural networks, allowing it to predict scheduling performance across parameters such as VM length and cost. By combining LA with RBF models, the proposed approach gains an intelligent decision-making capability that can enhance scheduling efficiency, reduce latency, and lower energy consumption. This hybrid framework positions LA as a key component in the optimization of fog environments, aligning scheduling decisions with energy efficiency and cost-effectiveness goals.

III. TASK SCHEDULING IN FOG COMPUTING

Task scheduling is a critical component in the design and optimization of fog computing environments, where computational resources are distributed across numerous nodes. Unlike centralized cloud systems, fog computing places computational power closer to end-users, reducing latency and enabling more responsive services. However, effective task scheduling in fog environments poses challenges due to the resource limitations of fog nodes and the diverse requirements of tasks. Ineffective scheduling can lead to suboptimal resource utilization, increased energy consumption, and prolonged MK, which negatively impact both performance and environmental goals. This complexity is further compounded by the need to maintain quality of service (QoS) while accommodating the fluctuating demands of IoT-driven networks. In this study, task scheduling is approached as a multi-objective optimization problem, aiming to balance critical parameters such as EC, MK, and cost. The proposed scheduling model leverages an LA-based approach, augmented with neural networks, to dynamically assign tasks to fog nodes based on available resources and task requirements. The LA scheduler operates in tandem with RBF networks, where the ANN model captures complex task allocation patterns, and predicts MK, fitness, and cost with respect to VM length. This integrated approach enables the system to make informed, adaptive scheduling decisions that optimize resource allocation and reduce energy usage, ultimately enhancing the efficiency and sustainability of the fog computing infrastructure.

A. Makespan

MK represents the total time required to complete all scheduled tasks within the system, measured from the start of the first task to the completion of the last. In a fog computing environment, reducing MK is essential to ensure prompt task execution and enhanced responsiveness, especially in applications that require real-time data processing. By minimizing MK, the system achieves a higher throughput, effectively handling more tasks within a given timeframe. There is also an allocation matrix called A_{ij} . When the task T_i is assigned to a virtual machine Vm_j , its value is 1. Otherwise, it is 0 [33].

$$Et_j = \sum_{i=1}^{m} A_{ij} \times e_{ij}\ ,\ e_{ij} = \frac{w_i}{s_j}\ ,\ MK = MAX\ (Et_j),\ j = 1,2,...,n \tag{1}$$

B. Fitness

Fitness is a composite metric that considers both energy consumption and MK to evaluate the overall efficiency of the scheduling model. This parameter is crucial for balancing the trade-offs between computational speed and energy usage. By optimizing for fitness, the model aims to maintain a balance between rapid task completion and sustainable energy consumption, thereby supporting both performance and environmental objectives. In the proposed model, fitness serves as a key indicator of how effectively tasks are assigned to nodes within the fog network.

$$Fitness = \tau \times total\_energy + (1-\tau) \times MK\ ,\tau = 0.8 \tag{2}$$

C. Cost

Cost is an important metric that reflects the resource expenditure associated with task execution on virtual machines. It includes factors such as the cost of CPU cycles, memory usage, and bandwidth utilization. Lowering the cost parameter directly impacts the operational efficiency of the fog system by reducing the expense of task scheduling across distributed nodes. Cost minimization ensures that the model remains economically viable while delivering high-quality services, making it a critical consideration for resource allocation strategies.

$$Total\_\cos t = \sum_{j=1}^{n} \cos t_j \times Et_j \tag{3}$$

D. Energy consumption

VMs can be in an active or idle state. The majority of their energy consumption is when they are active. Therefore, the energy consumed by the VM is equal to the sum of the energy consumption in the active and idle states. The idle time of the virtual machine is equal to the subtraction of the execution time of the virtual machine from MK. The idle time of the virtual machine is $MK - Et_j$ Also, Et_j is the total processing time of T_i on the VM_j . The energy consumed by each machine is calculated as follows [34].

$$Energy\ (VM_j) = (Et_j \times \beta_j + (MK - Et_j) \times \alpha_j) \times s_j\ ,\ \beta_j = 10^{-8} \times (S_j^{\ 2})\ ,\ \alpha_j = 0.6 \times \beta_j \tag{4}$$

TABLE I. MODEL NOTATIONS

Symbol	Definition
MK	Makespan
T_i	Task
α	Weight coefficient for M in fitness function
β	Weight coefficient for energy in fitness function
Energy	Total energy consumption
Et_j	The total processing time of T_i
Cost	Total operational cost
SD	Standard deviation
ρ	p-value in statistical testing
τ	Weight for energy consumption in the fitness function=0.8
A_{ij}	Allocation matrix
ej	Execution time of task j on a reference machine
S_j	Machine speed - Processing capability of machine j
wj	Task length - Computational size of task l (number of instructions)

IV. PROPOSED ALGORITHM

The proposed algorithm aims to assign tasks $\xi = \{\xi_1, \xi_2, ..., \xi_m\}$ efficiently to fog nodes $\psi = \{\psi_1, \psi_2, ..., \psi_n\}$, while minimizing both MK and EC within the fog environment. Given the limited resources and high energy demands of fog nodes, effective task scheduling becomes essential to achieving optimal resource utilization and energy efficiency. To address this, an LA approach is employed, which bases task allocation decisions on predicted processing times and the specific resource requirements of each task. The LA-based scheduler, denoted as $\underset{T}{A}$,operates by selecting actions that represent task scheduling decisions. These actions, defined as $\Upsilon_f^t = \{\overset{i}{\underset{T}{A}} \mid \forall i \in \xi = \{\xi_1, \xi_2, ..., \xi_n\}\}$,allow the scheduler to assign tasks based on current system conditions. To facilitate this, an action probability vector $\underset{T}{P} = \{\overset{i}{\underset{s}{P}} \mid \forall \in \{1, 2, ..., N_m\}\}$ is maintained, where each $\overset{i}{\underset{s}{P}}$ reflects the likelihood of selecting a particular task for processing. Initially, all tasks are assigned equal probability, allowing each task an equal chance to access fog resources. At each iteration, the automaton $\underset{T}{A}$ randomly selects an action $\overset{i}{\underset{s}{\Upsilon}}$ from the action probability vector, effectively permitting task ξ_i to enter the fog computing system for processing. If no tasks remain or if the system lacks adequate resources to fulfill a task, the algorithm terminates. Otherwise, the algorithm evaluates whether the resource requirements for processing ξ_i are within the available resources of the fog system and checks if the predicted execution time for ξ_i is the shortest among all pending tasks. If both conditions are met, the algorithm rewards the selected task, penalizes other tasks in the APV, and updates the probability vector accordingly. In cases where these conditions are not satisfied, such as when resources are insufficient or the task's predicted processing time exceeds other tasks, the algorithm penalizes the selected task and rewards the remaining tasks, thereby adjusting the probability vector to favor more suitable candidates. Any task that has been successfully scheduled is removed from the action set of $\underset{T}{A}$, reducing the selection pool for subsequent iterations. After each scheduling decision, the automaton resets and reinitiates the selection process, iteratively optimizing task allocation until all tasks are processed or resources are exhausted. By iterating through this adaptive reward-penalty mechanism, the proposed LA algorithm effectively minimizes MK and EC, ensuring that fog node resources are utilized optimally and that tasks are scheduled in alignment with the system's dynamic conditions.

Task scheduling algorithm based of learning automata (TSLA)

```
Input : ξ = {ξ_1, ξ_2, ..., ξ_m}
 Output : Scheduled ξ_i
  If ξ is empty, the TSLA algorithm was terminated.
   Else
    Begin
      For ξ = {ξ_1, ξ_2, ..., ξ_m}
        Ξ_i = One of the tasks is randomly selected based on the probability vector.
        If (Length of Ξ_i < Length Fog) && (The processing time required Ξ_i is the smallest)
          Then
            Begin
              Reward the selected task in probability vector also penalize the other tasks
              Fog resources are assigned to the Ξ_i , Execution
              The selected task is removed from the input list and the probability vector and both
               Lists are updated
            End
        Else
          Begin
            The selected Ξ_i is penalized and other tasks are rewarded then the probability vector
             Updated.
          End
    End
```

V. DATA SET DESCRIPTION

Regarding the implementation of the algorithm, three scenarios are defined. Table 2 shows the two scenarios defined for implementing the proposed algorithm in a homogeneous environment. Also, the third scenario is implemented to examine the performance of the proposed algorithm in a heterogeneous environment based on the CISCO IoT TRACE [1] dataset.

TABLE II. ALGORITHM IMPLEMENTATION SCENARIOS

	Length of tasks	#VM
First scenario	Different sizes = [100,1000]	Fixed number = 50
Second scenario	Fixed-length = 500	Different sizes = [10,100]
Third scenario	Based on Cisco IoT=5000	Weak, Medium and Strong nodes

A. Implementation of the proposed algorithm using the first scenario

This experiment was conducted with a fixed number of virtual machines in order to investigate the effect of increasing the length of tasks on MK parameters and energy consumption in the Fog environment. It is clearly known that the increase in the length of tasks increases the workload in virtual machines, which will lead to delays in the completion of tasks and an increase in MK. The results obtained for MK are shown in Figure 1. Also, we can see that the proposed algorithm has the lowest MK among all algorithms. In addition, the total energy consumed in the fog environment for task processing is shown in Figure 2. This result shows that the proposed algorithm has achieved the lowest energy consumption among other algorithms.

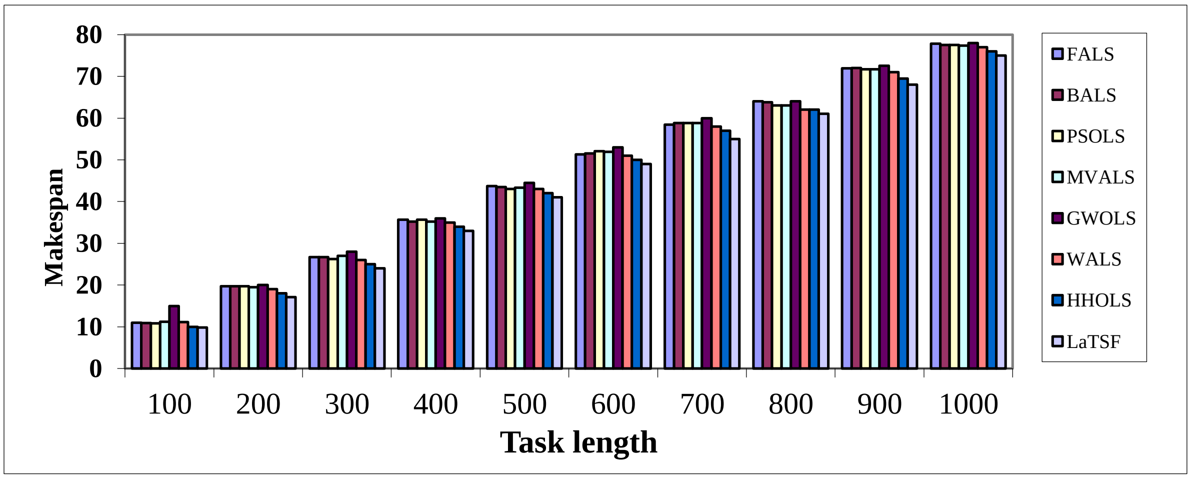

Figure 1. Makespan versus task length in Scenario 1 (homogeneous fog with 50 VMs). Task length increases from 100 to 1000 while the number of VMs stays fixed. We compare our LaTSF scheduler with seven baselines (FALS, BALS, PSOLS, MVALS, GWOLS, WALS, HHOLS). As tasks get longer, the makespan rises for all methods, but LaTSF stays lowest across the range, showing better scalability under heavier workloads.

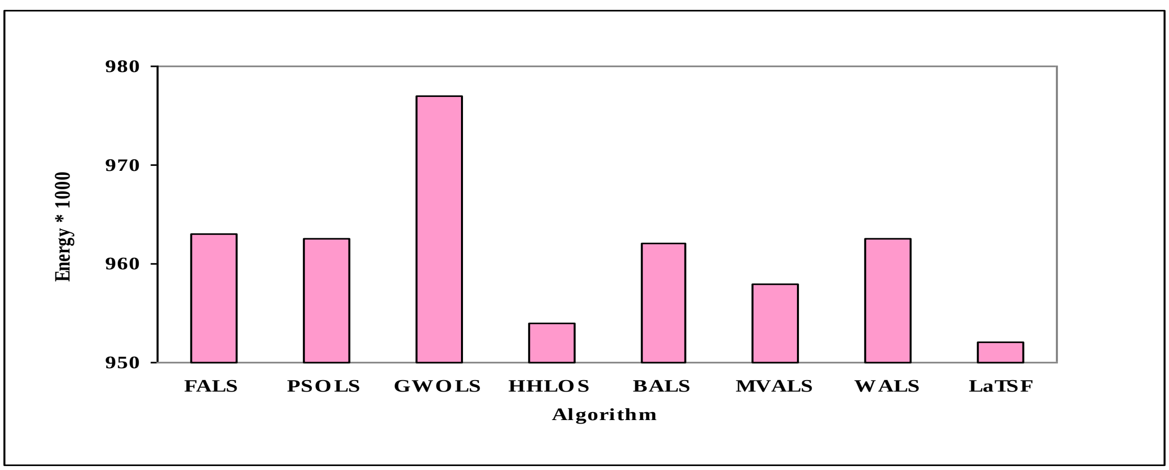

Figure 2. Comparison of energy consumption (in thousands of units) for different scheduling algorithms when running fixed-length tasks on 50 VMs. The LaTSF algorithm consistently achieves the lowest energy consumption among all the compared methods (FALS, PSOLS, GWOLS, HHOLS, BALS, MVALS, WALS). The

results show that GWOLS consumes the highest energy, followed by other algorithms, while LaTSF optimizes energy usage effectively, highlighting its efficiency in fog computing environments.

B. Implementation of the proposed algorithm using the second scenario

In this step, the proposed algorithm has been implemented using tasks with fixed length and VMs with different processing capacities. Also, the obtained results have been compared with seven other algorithms. The results obtained from the proposed algorithm for the fitness parameter and their comparison with other algorithms are shown in Figure 3. In this implementation, it is clear that the TSLA algorithm has obtained the lowest fitness value among these other methods. Also, the energy consumption, MK, and processing power of the VM affect the fitness changes. It should be noted that the activity of virtual machines increases energy consumption. Figure 4 shows that increasing the length of the VMs will decrease the MK. Figure 5 shows that the TSLA algorithm has the lowest energy consumption among other algorithms. On the other hand, Figure 6 shows that running the proposed algorithm on a virtual machine has the lowest cost compared to other algorithms.

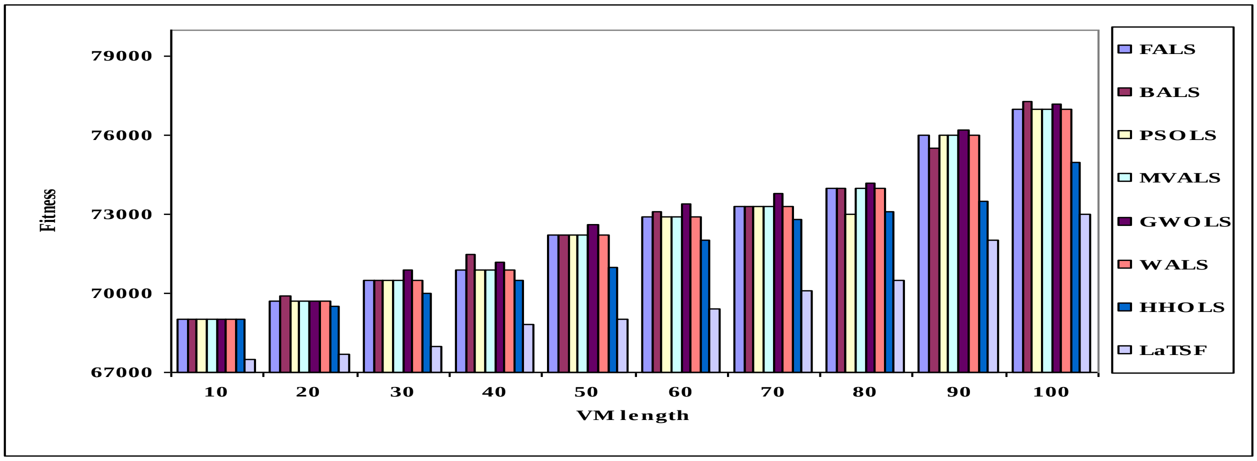

Figure 3. Comparison of fitness values for 500 tasks and VMs of varying lengths (from 10 to 100) using different scheduling algorithms. The results show that LaTSF consistently achieves the lowest fitness value, demonstrating superior performance in optimizing both energy consumption and task completion time. The other methods (FALS, BALS, PSOLS, MVALS, GWOLS, WALS, HHOLS) show higher fitness values, indicating less efficient resource utilization and scheduling compared to LaTSF.

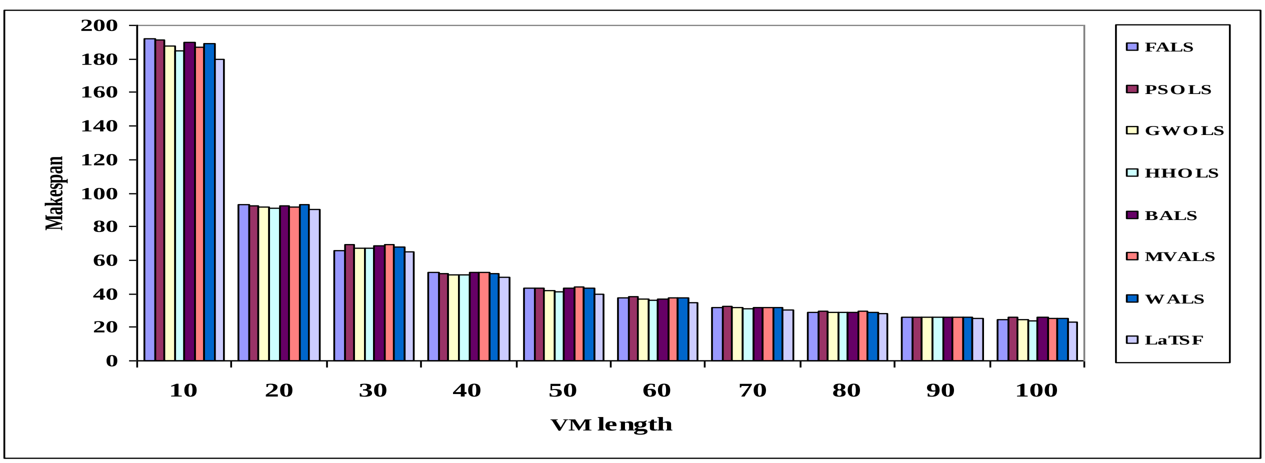

Figure 4. Makespan for 500 fixed-length tasks as VM length increases from 10 to 100 in Scenario 2. Each group compares LaTSF with other methods on the same workload. As VM capacity grows, makespan drops for all methods, but LaTSF stays lowest across the range, showing better use of stronger VMs and faster overall completion.

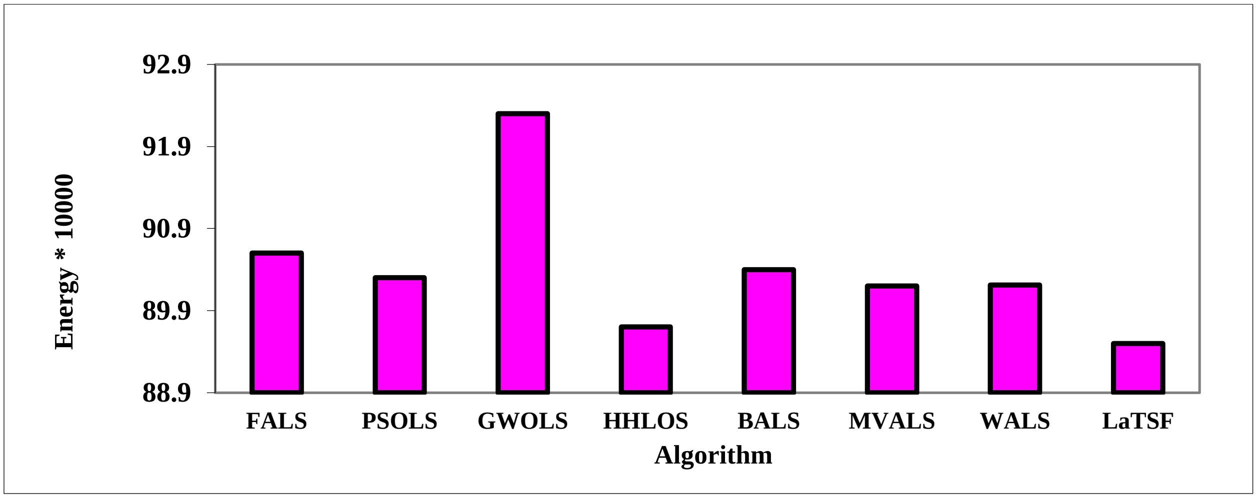

Figure 5. Comparison of energy consumption (in units of 10,000) for the proposed LaTSF algorithm and seven other scheduling methods in Scenario 2 (500 tasks with VMs of varying lengths). LaTSF consistently exhibits the lowest energy consumption, demonstrating its efficiency in resource utilization. In contrast, GWOLS consumes the highest energy, followed by other methods, highlighting the superiority of LaTSF in optimizing energy use.

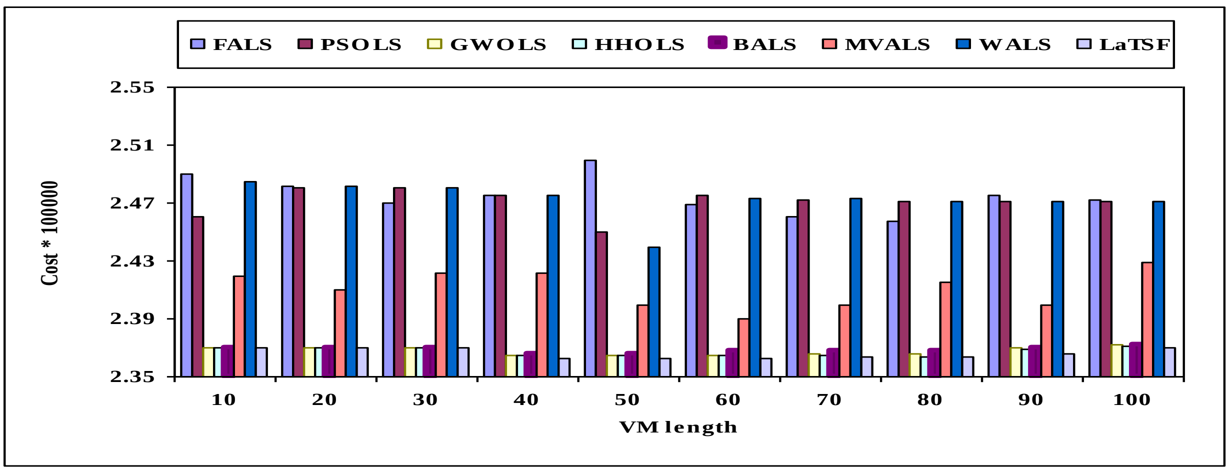

Figure 6. Comparison of operational cost (in units of 10,000) for 500 fixed-length tasks and VMs of varying lengths (10 to 100) using different scheduling algorithms. The LaTSF algorithm consistently achieves the lowest cost, demonstrating its ability to optimize resource usage while minimizing financial overhead.

C. Implementation of the proposed algorithm using the third scenario

In addition to the two scenarios 1 and 2, we implemented the proposed method in a heterogeneous fog environment using a CISCO IoT TRACE dataset consisting of 5000 tasks and 50 fog nodes with three different lengths (20 weak nodes, 20 medium nodes, and 10 strong nodes). This experiment clearly demonstrates the superiority of the proposed method compared to other methods. Figure 7 shows that the proposed LATSF algorithm with the lowest total MK has the best performance among the compared algorithms. The significant difference in the performance of the proposed algorithm compared to other methods is due to the use of an LA in task allocation, which leads to a balanced workload distribution and reduced waiting time in processing queues. It should also be noted that based on the results obtained in the scenario, it is shown that the proposed method has competitive performance in key criteria and can be used in large-scale environmental conditions. Figure 8 shows the energy consumption of seven different scheduling algorithms with the proposed method in a heterogeneous fog environment. As can be seen in the figure, the proposed LATSF algorithm has the lowest energy consumption among the other compared algorithms. In Figure 9, the proposed algorithm with the lowest Fitness value of 0.42 has the best overall performance among the other compared algorithms. This result shows that the proposed algorithm has been able to optimize both energy consumption and execution time metrics simultaneously in a desirable manner. By considering the cost of using each resource and avoiding excessive use of expensive resources (such as powerful nodes) for simple tasks, the LATSF algorithm has been able to significantly reduce operational costs so that it has the lowest cost among others, as can be seen in Figure 10.

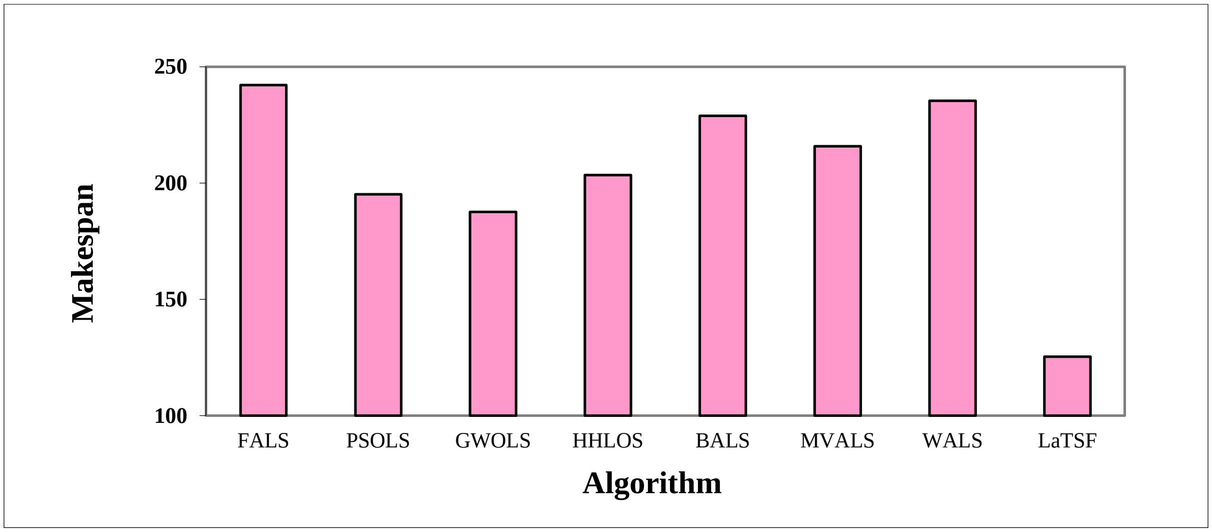

Figure 7. Comparison of Makespan (MK) for the proposed LaTSF algorithm and other scheduling methods in a heterogeneous fog environment. The results show that LaTSF achieves the lowest makespan, demonstrating its efficiency in completing tasks in a shorter time compared to the other algorithms.

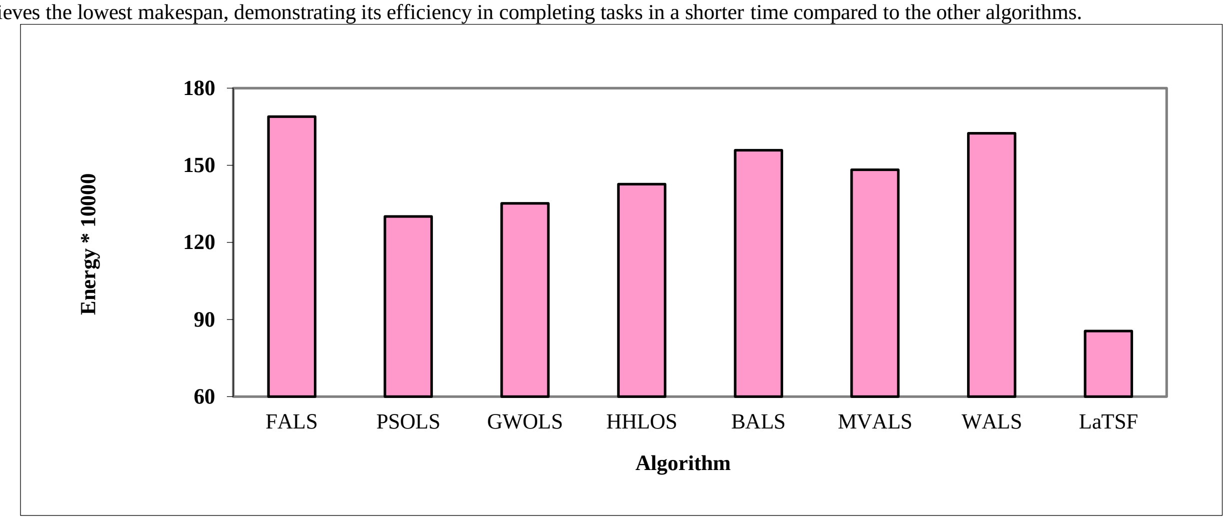

Figure 8. Comparison of energy consumption (in units of 10,000) for the proposed LaTSF algorithm and other scheduling methods in a heterogeneous fog environment. LaTSF demonstrates the lowest energy usage, highlighting its efficiency in minimizing energy consumption compared to the other algorithms, which consume significantly more energy under the same conditions.

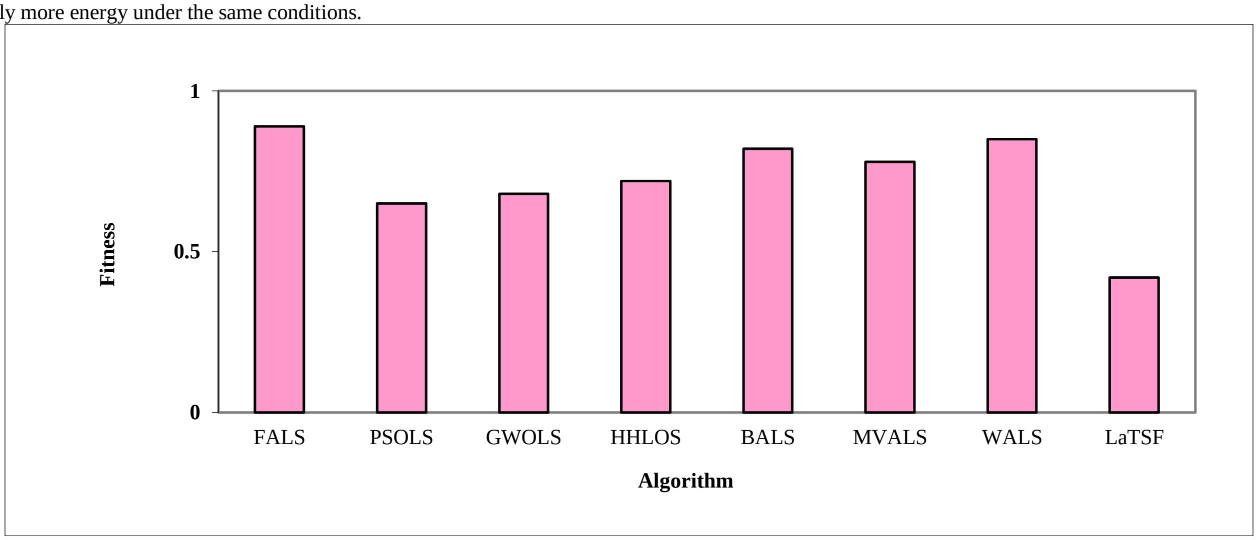

Figure 9. Comparison of fitness values for the proposed LaTSF algorithm and other scheduling methods in a heterogeneous fog environment. Fitness is a combined metric considering both energy consumption and makespan, which LaTSF achieves the lowest fitness value.

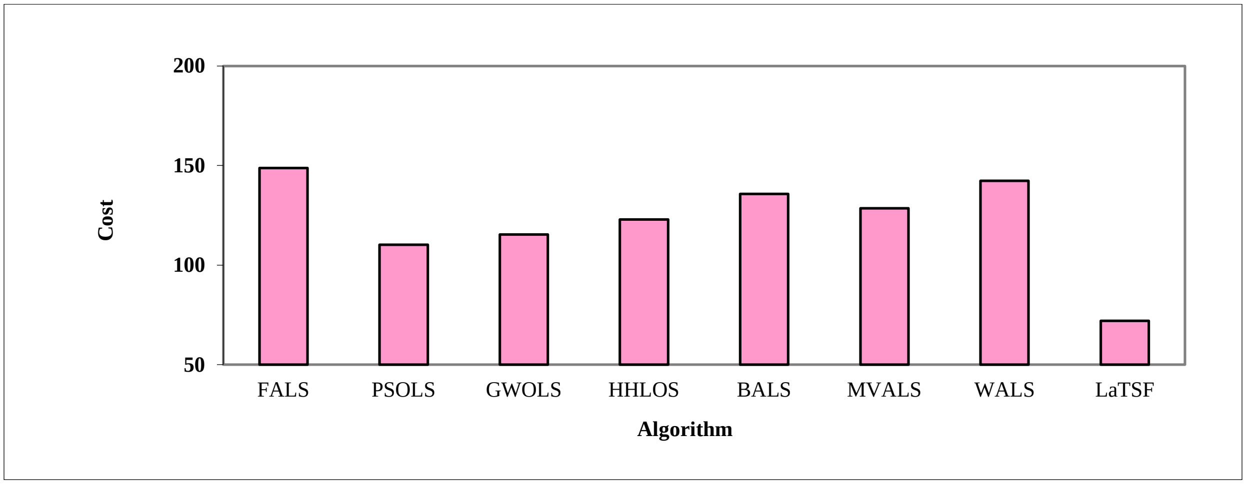

Figure 10. Comparison of the cost for the proposed LaTSF algorithm and other scheduling methods in a heterogeneous fog environment. The cost parameter reflects the operational expenses associated with task execution across different algorithms, and LaTSF achieves the lowest cost.

VI. STATISTICAL VALIDATION OF RESULTS

To statistically substantiate the performance superiority of the proposed LaSTF algorithm, independent *t*-tests were conducted on the results obtained from the first experimental scenario, focusing on three key performance indicators: MK, fitness, and energy consumption. The statistical analysis revealed that LaSTF consistently outperforms competing approaches across all evaluation metrics with a high level of significance:

- MK: LaSTF achieved the lowest average MK of 46.46, demonstrating a statistically significant improvement ($p \leq 0.009$).
- Fitness: The proposed algorithm obtained the lowest mean fitness value of 68,880.0 ± 758.3 (SD), indicating stable and superior optimization performance.
- Energy Consumption: LaSTF recorded the minimum energy usage of 952 units, confirming its energy efficiency. The statistical validity of this result is further supported by the *t*-test outcomes presented in Table 3.

In summary, the *t*-test results confirm that LaSTF's advantages are not limited to numerical improvements but are also statistically significant, highlighting its robustness and effectiveness across multiple performance dimensions.

TABLE III. STATISTICAL COMPARISON OF ALGORITHMS USING INDEPENDENT T-TEST

Algorithm	MK		Fitness		energy	Statistical Comparison	t-value	p-value
	Mean MK±SD	P-value	Mean F±SD	P-value			4.81	0.0015
LaTSF	46.46 ± 27.29	NA	68880.0 ± 758.3	NA	952	963.8	NA	
BaseLine	48.53 ± 26.80	0.0013	71085.0 ± 1954.5	0.0000008	963.8	NA	NA	
FALS	48.48 ± 26.84	0.0012	72448.0 ± 3492.8	0.000002	964	NA	NA	
PSOLS	48.35 ± 26.72	0.0010	72498.0 ± 3492.8	0.000002	963	NA	NA	
GWOLS	48.46 ± 26.79	0.0011	72498.0 ± 3492.8	0.000002	978	NA	NA	
HHOLS	49.90 ± 27.31	0.0002	72498.0 ± 3492.8	0.000002	957	NA	NA	
BALS	48.70 ± 26.79	0.0008	72595.0 ± 3518.9	0.000002	962	NA	NA	
MVALS	47.60 ± 26.65	0.009	72785.0 ± 3565.9	0.000001	961	NA	NA	
WVALS	48.53 ± 26.80	0.0008	68880.0 ± 758.3	0.000002	962.6	NA	NA	

VII. PROPOSED RBF NEURAL NETWORK MODEL

The radial basis function (RBF) network is a type of artificial neural network commonly used for function approximation, classification, and time-series prediction. It consists of three layers: an input layer, a hidden layer with radial basis activation functions, and an output layer. The hidden layer uses a distance metric, often the Euclidean distance, between the input and a set of central points known as "centroids" to determine the activation of each neuron. This network is particularly effective at solving engineering problems, such as signal processing and pattern recognition, due to its ability to approximate complex nonlinear functions. The proposed RBF structure is illustrated in Figure 11. As seen, a hidden layer with 20 neurons is considered for the proposed RBF.

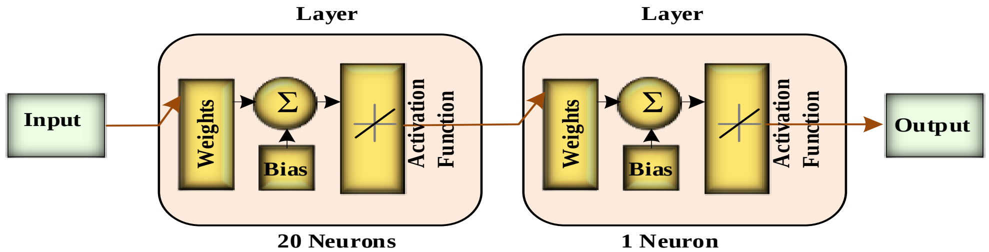

Figure 11. Structure of the propose RBF network.

The performance of the proposed RBF model for the fitness output parameter during the training phase is shown in Figure 12. This plot shows the MSE of the training data versus different epochs, which shows good convergence of the RBF network in the training level.

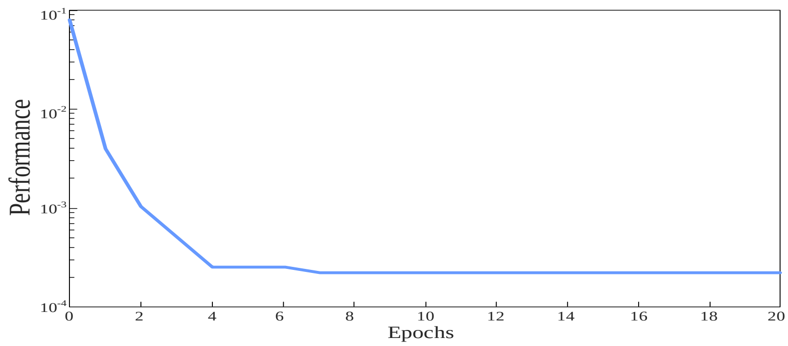

Figure 12. Performance of the training phase of the proposed RBF model for fitness output parameter.

Results of the predicted test and train data using the proposed RBF model for three output parameters of MK, cost, fitness in the train and test phases are plotted in Figure 13. According to the results, cost and fitness parameters are predicted more accurately compared to the MK parameter.

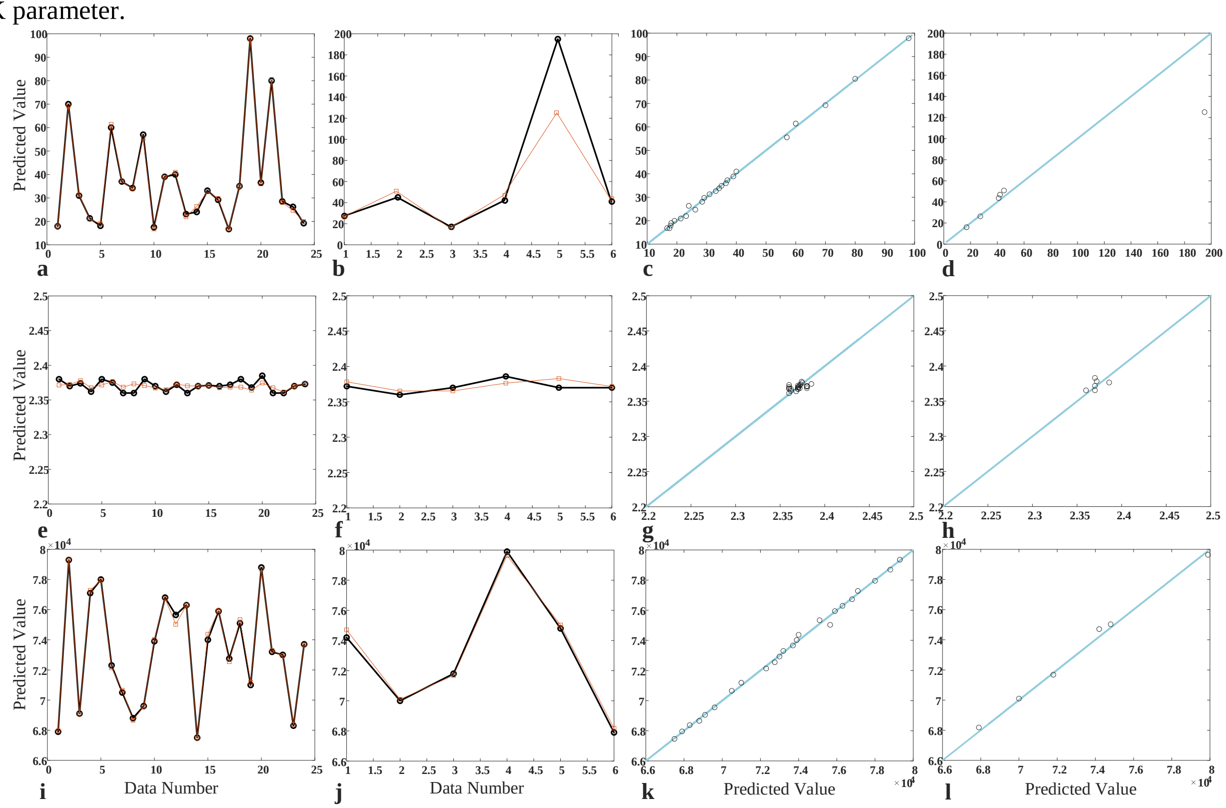

Figure 13. Results of the predicted test and train data using proposed RBF model. Predicted values of MK parameter in (a) train and (b) test phases. Regression plots of the MK parameter in (c) train and (d) test phases. Predicted values of cost parameter in (e) train and (f) test phases. Regression plots of cost parameter in (g) train and (h) test phases. Predicted values of fitness parameter in (i) train and (j) test phases. Regression plots of the fitness parameter in (k) train and (l) test phases.

The calculated MRE error results of the predicted output parameters are listed in Table 4. For the proposed RBF network with the considered outputs, the obtained low MRE suggests that the network is performing well in approximating the target functions across all output dimensions.

TABLE IV. CALCULATED MRE ERROR RESULTS OF THE PROPOSED RBF NETWORK

Output Parameter	MRE Train	MRE Test
MK	0.022976	0.12648
Cost	0.002109	0.0028419
Fitness	0.0017878	0.0033881

VIII. PRACTICAL DEPLOYMENT AND FUTURE DIRECTIONS

A. Practical deployment outlook

The proposed LaTSF algorithm is well-suited for real-world IoT and fog computing applications, such as healthcare, transportation, and smart cities. The algorithm effectively addresses the challenges of resource heterogeneity by categorizing fog nodes into different tiers and assigning tasks based on available node capabilities. This ensures that more resource-intensive tasks are handled by high-capacity nodes while simpler tasks are assigned to lower-capacity ones. To minimize system overhead, LaTSF utilizes local task scheduling decisions and reduces communication requirements. In terms of security, it is recommended to integrate encryption, secure communication protocols, and access control mechanisms to ensure data privacy and the integrity of the system. These practical considerations make the LaTSF approach a feasible and efficient solution for deployment in resource-constrained, dynamic environments.

B. Limitations and future work

According to the findings, the proposed hybrid model based on learning automata and RBF neural networks has achieved better performance in reducing execution time, energy consumption, and costs in heterogeneous fog environments compared to other methods. However, there are limitations such as increased computational complexity when running in large environments, scalability challenges in large-scale networks and the need for greater adaptability for environments with dynamic workloads. In future work, we will seek to address these challenges. We plan to develop an adaptive scheduler for dynamic workloads using advanced reinforcement learning techniques. Integration with 5G/6G networks will also be explored to exploit capabilities such as network slicing to improve quality of service (QoS) and scalability. Finally, we aim to develop a more comprehensive framework by integrating lightweight security protocols to improve the model's practical applicability in the real world.

IX. CONCLUSIONS

This paper presents a novel approach to task scheduling in fog computing by combining a Learning Automata (LA)-based scheduling algorithm with RBF neural network models. The proposed method addresses key challenges in fog environments, including energy consumption, MK, and operational cost, while optimizing task allocation across distributed virtual machines (VMs). To investigate the comprehensive performance of the proposed algorithm, we conducted extensive experiments in three distinct scenarios, in large-scale homogeneous and heterogeneous fog environments using the Cisco IoT Trace dataset. The integration of the RBF model enhances the system's predictive capabilities, allowing for accurate estimation of MK, cost, and fitness in response to changes in VM length. The RBF network, specifically structured with 20 neurons in the hidden layer, demonstrated strong performance in approximating these output parameters. Results indicate that cost and fitness were predicted with high accuracy, as evidenced by low MRE values in both training and test phases. Although MK predictions exhibited slightly higher MRE, the model's overall effectiveness in function approximation was validated by low error rates across all outputs, showcasing the RBF's capacity for precise modeling of complex, nonlinear relationships within the fog network. Additionally, the RBF network showed good convergence during training, with MSE decreasing consistently over epochs, as illustrated in the regression plots for the fitness parameter. Through simulations, the proposed LA-RBF approach achieved notable reductions in MK and energy consumption compared to conventional scheduling methods, underscoring its potential to support energy-aware fog computing. The performance of the proposed method and the accuracy of the obtained results were also accurately confirmed through independent t-tests (p-value ≤ 0.009).

REFERENCES

[1] Cisco-Systems. "Cisco IoT Trace Dataset." https://developer.cisco.com/iot/ (accessed 15 Oct. 2023.
[2] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, "Edge computing and cloud computing for internet of things: A review," in *Informatics*, 2024, vol. 11, no. 4: MDPI, p. 71.
[3] M. Trigka and E. Dritsas, "Edge and cloud computing in smart cities," *Future Internet*, vol. 17, no. 3, p. 118, 2025.

[4] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *International Journal of Communication Systems*, vol. 33, no. 8, p. e4340, 2020.
[5] S. Pallewatta, V. Kostakos, and R. Buyya, "QoS-aware placement of microservices-based IoT applications in Fog computing environments," *Future Generation Computer Systems*, vol. 131, pp. 121-136, 2022.
[6] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE international conference on edge computing (EDGE)*, 2017: IEEE, pp. 17-24.
[7] D. Zhao, Q. Zou, and M. Boshkani Zadeh, "A QoS-aware IoT service placement mechanism in fog computing based on open-source development model," *Journal of Grid Computing*, vol. 20, no. 2, p. 12, 2022.
[8] R. Rateb *et al.*, "An optimal workflow scheduling in IoT-fog-cloud system for minimizing time and energy," *Scientific Reports*, vol. 15, no. 1, p. 3607, 2025.
[9] R. Ebrahim Pourian, M. Fartash, and J. Akbari Torkestani, "A deep learning model for energy-aware task scheduling algorithm based on learning automata for fog computing," *The Computer Journal*, vol. 67, no. 2, pp. 508-518, 2024.
[10] R. Ebrahim Pourian, M. Fartash, and J. Akbari Torkestani, "A new approach to the resource allocation problem in fog computing based on learning automata," *Cybernetics and Systems*, vol. 55, no. 7, pp. 1594-1613, 2024.
[11] E. Khezri, R. O. Yahya, H. Hassanzadeh, M. Mohaidat, S. Ahmadi, and M. Trik, "DLJSF: data-locality aware job scheduling IoT tasks in fog-cloud computing environments," *Results in Engineering*, vol. 21, p. 101780, 2024.
[12] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective Task Scheduling in cloud-fog computing using goal programming approach," *Cluster Computing*, vol. 25, no. 1, pp. 141-165, 2022.
[13] N. Mansouri and R. Ghafari, "Cost-Efficient Task Scheduling Algorithm for Reducing Energy Consumption and Makespan of Cloud Computing," *Computer and Knowledge Engineering*, vol. 5, no. 1, pp. 1-12, 2022.
[14] S. Ijaz, E. U. Munir, S. G. Ahmad, M. M. Rafique, and O. F. Rana, "Energy-makespan optimization of workflow scheduling in fog–cloud computing," *Computing*, vol. 103, pp. 2033-2059, 2021.
[15] A. Sathya Sofia and P. GaneshKumar, "Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II," *Journal of Network and Systems Management*, vol. 26, pp. 463-485, 2018.
[16] H. Topcuoglu, S. Hariri, and M.-Y. 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.
[17] P. Karami *et al.*, "Design of a photonic crystal exclusive-OR gate using recurrent neural networks," *Symmetry*, vol. 16, no. 7, p. 820, 2024.
[18] S. Roshani *et al.*, "Mutual coupling reduction in antenna arrays using artificial intelligence approach and inverse neural network surrogates," *Sensors*, vol. 23, no. 16, p. 7089, 2023.
[19] S. Roshani *et al.*, "Design of a Microwave Quadrature Hybrid Coupler with Harmonic Suppression Using Artificial Neural Networks," *Active and Passive Electronic Components*, vol. 2024, no. 1, p. 8722642, 2024.
[20] P. Karami *et al.*, "Design and Modeling of a Photonic Crystal Multiplexer Using Artificial Intelligence," *Advanced Electromagnetics*, vol. 14, no. 1, pp. 59-64, 2025.
[21] S. Roshani, S. I. Yahya, B. Najafi, A. Jadidian, M. Karimi, and S. Roshani, "Optimizing a Compact Ring Coupler with Neural Network Modeling for Enhanced Performance in Radio Frequency Applications," *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY*, vol. 13, no. 1, pp. 122-130, 2025.
[22] M. Jamshidi, S. I. Yahya, S. Roshani, M. A. Chaudhary, Y. Y. Ghadi, and S. Roshani, "A fast surrogate model-based algorithm using multilayer perceptron neural networks for microwave circuit design," *Algorithms*, vol. 16, no. 7, p. 324, 2023.
[23] S. I. Yahya, S. Roshani, M. Ami, Y. Y. Ghadi, M. A. Chaudhary, and S. Roshani, "A compact rat-race coupler with harmonic suppression for GSM applications: design and implementation using artificial neural network," *Micromachines*, vol. 14, no. 7, p. 1294, 2023.
[24] P. Choppara and S. Mangalampalli, "An efficient deep reinforcement learning based task scheduler in cloud-fog environment," *Cluster Computing*, vol. 28, no. 1, p. 67, 2025.
[25] F. Ramezani, J. Lu, and F. Hussain, "Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization," in *Service-Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings 11*, 2013: Springer, pp. 237-251.
[26] A. Awad, N. El-Hefnawy, and H. Abdel_kader, "Enhanced particle swarm optimization for task scheduling in cloud computing environments," *Procedia Computer Science*, vol. 65, pp. 920-929, 2015.
[27] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba, and M. Zivkovic, "Task scheduling in cloud computing environment by grey wolf optimizer," in *2019 27th telecommunications forum (TELFOR)*, 2019: IEEE, pp. 1-4.
[28] A. Alzaqebah, R. Al-Sayyed, and R. Masadeh, "Task scheduling based on modified grey wolf optimizer in cloud computing environment," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, 2019: IEEE, pp. 1-6.
[29] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-12, 2020.
[30] H. Liu, "Research on cloud computing adaptive task scheduling based on ant colony algorithm," *Optik*, vol. 258, p. 168677, 2022.
[31] M. U. Khan, "Optimization of Project Scheduling Problem Using Multi-Objective Bat-inspired Algorithm and comparison with other Nature Inspired Algorithms," 2019.

[32] T. Bezdan, M. Zivkovic, N. Bacanin, I. Strumberger, E. Tuba, and M. Tuba, "Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 1, pp. 411-423, 2022.
[33] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 976-986, 2018.
[34] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12638-12649, 2020.