International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Smartflux: A Dual-Phase Resource Orchestration Model for IOT-Fog-Cloud Ecosystems

M. Kumar
(Assistant Professor)
Computer Science Engineering
(Internet of Things)
Guru Nanak Institutions Technical Campus
Hyderabad, Telangana, India
maddekumar.iotgnitc@gniindia.org

Amogh Pitti
Computer Science Engineering
(Internet of Things)
Guru Nanak Institutions Technical Campus
Hyderabad, Telangana, India
amoghpitti@gmail.com

Mohammed Yaseen
Computer Science Engineering
(Internet of Things)
Guru Nanak Institutions Technical Campus
Hyderabad, Telangana, India
mohammedseen02@gmail.com

Peri Sri Nitya Annapurna
Computer Science Engineering
(Internet of Things)
Guru Nanak Institutions Technical Campus
Hyderabad, Telangana, India
nityaperi09@gmail.com

Abstract *- The rapid expansion of Internet of Things (IoT) ecosystems has triggered an immense surge in data generation, demanding computational models that can efficiently manage and process this influx. Although cloud computing provides scalable resources for such tasks, its inherent latency and lack of contextual responsiveness limit its effectiveness for time-sensitive IoT applications. Fog computing, introduced to bridge this gap by enabling localized processing closer to data sources, offers reduced latency but is constrained by limited computational capacity. To overcome these limitations, this research introduces a hybrid IoT–fog–cloud framework that strategically balances real-time responsiveness with computational scalability. A two-phase resource allocation mechanism is proposed: initially, tasks are distributed based on a task guarantee ratio to either fog or cloud layers; subsequently, a Bayesian classifier refines this allocation using historical data for adaptive scheduling. To further enhance performance, the Crayfish Optimization Algorithm (COA), a novel bio-inspired metaheuristic, is employed to minimize execution delays and system latency. Simulations conducted via the iFogSim toolkit confirm the effectiveness of the proposed model, showcasing superior task handling and reduced latency compared to existing approaches.*

Keywords:
IoT, fog computing, cloud computing, resource allocation, task classification, Bayes classifier, COA.

I. INTRODUCTION

The proliferation of IoT devices has led to massive data generation and increased demand for real-time processing. While cloud computing offers scalable processing and storage, it suffers from high latency and lacks location awareness, making it unsuitable for delay-sensitive applications. Fog computing, positioned closer to data sources, addresses latency by enabling local processing. However, its limited computational capacity restricts its ability to manage large-scale or resource-intensive tasks.

To overcome these limitations, a collaborative fog–cloud model is essential. Fog nodes handle real-time, latency-critical tasks, while the cloud manages data-heavy computations. Effective resource allocation in such a hybrid environment remains challenging due to task diversity and QoS constraints.

This paper introduces a two-stage resource allocation framework that classifies and schedules tasks across fog and cloud layers. Initially, tasks are grouped based on a guaranteed ratio, followed by Bayesian classification using historical allocation data. The Crayfish Optimization Algorithm (COA) is employed for optimal resource mapping, minimizing delay and execution time. This approach enhances overall system efficiency by leveraging the strengths of both fog and cloud infrastructures.

II. LITERATURE SURVEY

Several resource allocation strategies have been proposed for IoT–fog–cloud environments, focusing on latency reduction and efficient task scheduling. Early models primarily optimized fog layer performance but lacked cloud integration, limiting scalability.

Enhanced metaheuristic algorithms such as modified Whale and Particle Swarm Optimization have shown promise in reducing execution time and energy consumption. However, these approaches often overlook hybrid scheduling, leading to fog resource overload. Other studies used linear programming or heuristic methods to classify tasks by service type, but relied on static task categories, reducing adaptability.

Decentralized schedulers have improved latency for real-time tasks but ignored cloud offloading. Priority-based models, especially in healthcare scenarios, prioritized urgent tasks effectively but assumed fixed task classifications. Game-

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

theoretic, queuing theory, and fuzzy clustering methods further advanced fog resource optimization but did not consider the fog–cloud interplay.

Overall, existing approaches either focus solely on fog or assume predefined task classes, lacking dynamic classification and hybrid resource coordination. This gap underscores the need for an adaptive framework that intelligently distributes tasks across fog and cloud layers.

III. METHODOLOGY

To address the limitations of both cloud and fog paradigms in handling diverse IoT workloads, we propose a two-stage task classification and resource allocation framework. The architecture is designed to balance delay-sensitive and data-intensive task execution between fog and cloud layers, leveraging their respective strengths.

A. System Architecture: The framework comprises several components: task-generating IoT devices, a dual-stage classifier, a resource allocator, and distributed resources at the fog and cloud layers. Tasks from end devices are first placed in an arrival queue and then processed by the classification module, which determines their execution suitability based on task characteristics.

B. Stage 1: Task Guarantee Ratio Classification: Initially, tasks are grouped by processing requirements. A guaranteed ratio is computed for each group on both fog and cloud layers, considering available CPU, memory, and bandwidth resources. Tasks with lower resource demands and latency constraints are assigned to the fog, while more intensive tasks are routed to the cloud. This stage ensures a baseline classification using current system capacity.

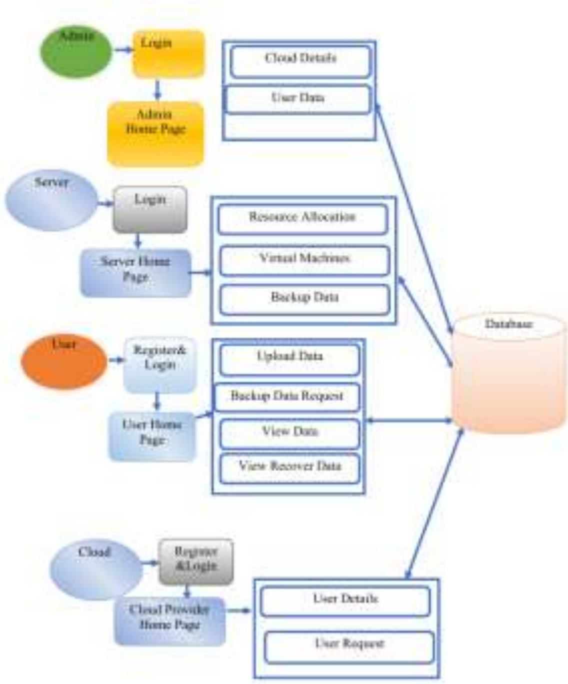
C. Stage 2: Bayesian Classification: To improve classification accuracy, a Bayesian model is applied using historical task execution data. Each new task is analyzed for its processing demand, memory usage, and deadline. Based on probability distributions learned from past allocations, the classifier predicts the most suitable layer for execution. This step enhances system adaptability under dynamic workloads.

D. Resource Allocation Using Crayfish Optimization: Once tasks are classified, resources within the assigned layer are allocated using the Crayfish Optimization Algorithm (COA). Inspired by crayfish foraging behavior, COA combines exploration and exploitation strategies to identify optimal resource-task mappings. Each solution is evaluated using a fitness function that minimizes execution time and total delay, including transmission, queuing, and processing components.

The algorithm iteratively adjusts task placement based on environmental conditions such as simulated temperature and feeding behavior, allowing convergence toward optimal allocation with minimal overhead.

E. Objective Function: The optimization aims to minimize a cost function composed of task execution time and end-to-end delay. These factors are modeled based on task parameters and resource capabilities. The overall objective is defined as a weighted sum of delay and processing time, with equal emphasis on both to balance responsiveness and efficiency.

IV. SYSTEM ARCHITECTURE



V. RESULT & ANALYSIS

To validate the effectiveness of the proposed IoT–fog–cloud task allocation framework, simulation experiments were conducted using the iFogSim toolkit, which supports modeling and evaluation of resource scheduling in fog and cloud environments. The performance of the proposed approach was benchmarked against three existing strategies: fog-only execution, cloud-only execution, and a random allocation method.

A. Simulation Setup: Tasks were generated uniformly across four categories, each with varying processing and memory requirements and deadlines. A total of

400 tasks were submitted in batches. The fog layer was configured with 25 nodes, while 10 cloud resources were available. The COA algorithm was employed for resource mapping, and classification was carried out in two phases: task guarantee ratio and Bayesian analysis.

B. Execution Time Analysis: Results demonstrated that the proposed framework consistently achieved lower execution times compared to benchmark methods. For a workload of 200 tasks, it outperformed fog-only and random strategies, with fog-only suffering due to resource saturation. As task volume increased to 400, the framework maintained stable performance, especially for high-demand tasks, by effectively distributing loads between fog and cloud layers.

C. Latency Evaluation: Latency was measured as the time from task submission to response delivery, factoring in queuing, transmission, and execution delays. The fog-only method showed the least latency due to proximity, while cloud-only faced higher delays. The proposed system achieved balanced latency by assigning delay-sensitive tasks to fog and computationally intensive tasks to cloud, resulting in significantly improved average latency under increased load.

D. Task Failure Rate: The system's robustness was further evaluated by analyzing task failure rates, defined by tasks missing their deadlines. The proposed method recorded the lowest failure rate across all task categories. In contrast, the cloud-only approach failed more time-sensitive tasks due to high transmission delays, and fog-only struggled with high-complexity tasks due to resource limitations. The proposed hybrid model minimized such failures by assigning tasks according to their profile and system state.

VI. FUTURE SCOPE

To further optimize the system's efficiency, future iterations could incorporate predictive analytics to anticipate the optimal layer and resource allocation for incoming tasks. This foresight would be grounded in historical execution patterns, thereby significantly reducing the overhead associated with real-time resource discovery. The current implementation, developed using the iFogSim simulation environment, was benchmarked against several deployment strategies, including randomized allocation, fog-exclusive processing, and cloud-only execution models.

Empirical results from these simulations consistently demonstrated the superiority of the proposed hybrid approach, particularly in minimizing execution delays, reducing latency, and lowering task failure rates. These findings underscore a critical insight: neither fog computing nor cloud computing, in isolation, is sufficient to handle the full spectrum of computational demands. Instead, a synergistic integration of both paradigms is essential to achieve scalable, resilient, and performance-optimized task execution in dynamic environments.

VII. CONCLUSION

The inherent limitations of both fog and cloud computing—namely, the high transmission latency for delay-sensitive tasks in cloud environments and the extended execution times for computation- and data-intensive tasks in fog layers—highlight the necessity for a more intelligent and adaptive resource allocation strategy. This study introduces a dual-stage resource allocation framework designed to optimize task distribution across fog and cloud infrastructures.

In the first stage, tasks entering the system are categorized based on their guaranteed ratios for successful execution on either the fog or cloud layer. These tasks are then assigned to the most suitable resources using the Cuckoo Optimization Algorithm (COA), ensuring efficient initial placement. In the second stage, a Bayesian classifier leverages historical allocation data to predict the optimal resource-layer pairing for newly arriving tasks. This predictive mechanism enhances the system's ability to manage large volumes of tasks with reduced allocation errors.

By integrating classification and optimization techniques, the proposed framework not only improves resource utilization but also ensures timely and reliable task execution, making it a robust solution for dynamic, heterogeneous computing environments.

VIII. REFERENCES

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014, doi: 10.1109/SURV.2013.042313.00197.

[2] S. Sefati, M. Mousavinasab, and R. Z. Farkhady, "Load balancing in cloud computing environment using the grey wolf optimization algorithm based on the reliability: Performance evaluation," Journal of Supercomputing, vol. 78, no. 1, pp. 18–42, May 2021, doi: 10.1007/s11227-021-03810-8.

[3] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multiobjective task scheduling in cloud-fog computing using goal programming approach," Cluster Computing, vol. 25, no. 1, pp. 141–165, Aug. 2021, doi: 10.1007/s10586-021-03371-8.

[4] H. Jin, S. Lv, Z. Yang, and Y. Liu, "Eagle strategy using uniform mutation and modified whale optimization algorithm for QoS-aware cloud service composition," Applied Soft

Computing, vol. 114, Jan. 2022, Art. no. 108053, doi: 10.1016/j.asoc.2021.108053.

[5] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," IEEE Wireless Communications, vol. 26, no. 1, pp. 87–93, Feb. 2019, doi: 10.1109/MWC.2019.1700441.

[6] H. Atlam, R. Walters, and G. Wills, "Fog computing and the Internet of Things: A review," Big Data and Cognitive Computing, vol. 2, no. 2, p. 10, Apr. 2018, doi: 10.3390/bdcc2020010.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in Proc. MCC Workshop on Mobile Cloud Computing, Aug. 2012, pp. 13–15, doi: 10.1145/2342509.2342513.

[8] R. K. Naha, S. Garg, A. Chan, and S. K. Battula, "Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment," Future Generation Computer Systems, vol. 104, pp. 131–141, Mar. 2020, doi: 10.1016/j.future.2019.10.018.

[9] Z. Movahedi, B. Defude, and A. M. Hosseininia, "An efficient population-based multi-objective task scheduling approach in fog computing systems," Journal of Cloud Computing, vol. 10, no. 1, pp. 1–31, Oct. 2021, doi: 10.1186/s13677-021-00264-4.

[10] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, Jun. 2017, doi: 10.1002/spe.2509.