Energy-Conserving Real-Time Task Scheduling via Edge Computing with Random Forest Algorithm

Dr. N. Sowri Raja Pillai[1], V.Vinothkumar[2], A. Nimromardin [3], G.Sriram[4]
Department of Information Technology,
RAAK College of Engineering And Technology,
Puducherry, India.
sowrirajacse@gmail.com

Abstract – With the rapid growth of Internet of Things (IoT) devices and the increasing demand for real-time data processing, energy-efficient task scheduling has become a critical challenge in edge computing environments. This paper presents a novel approach to real-time task scheduling aimed at minimizing energy consumption, leveraging edge computing capabilities and the Random Forest algorithm. Edge computing allows for the distribution of computational tasks closer to the data source, reducing latency and improving system performance. The Random Forest algorithm, known for its robustness and ability to handle complex data patterns, is applied to optimize task assignment in dynamic and resource-constrained environments.

Index Terms – RF algorithm, edge-computing, multi-core, real-time systems, feasibility analysis, HPC.

1. INTRODUCTION

The proliferation of Internet of Things (IoT) devices and real-time applications has led to an unprecedented demand for energy-efficient and responsive computing paradigms. Edge computing has emerged as a promising solution, enabling data processing and analysis at the network edge, thereby reducing latency and energy consumption.

However, scheduling real-time tasks on edge devices while minimizing energy consumption remains a significant challenge. Traditional scheduling algorithms often rely on simplistic models, neglecting the complexities of real-world workloads and device characteristics.

To address this limitation, this research proposes a novel energy-conserving real-time task scheduling framework leveraging edge computing and the Random Forest algorithm. By integrating the strengths of edge computing and machine learning, our approach aims to optimize task scheduling decisions, minimizing energy consumption while ensuring real-time constraints.

The proposed framework will be designed to:

- Leverage edge computing to reduce latency and energy consumption

- Employ the Random Forest algorithm to learn complex relationships between task characteristics, device properties, and energy consumption

- Develop an energy-aware scheduling strategy that optimizes task allocation and execution on edge devices

By exploring the synergy between edge computing and machine learning, this research aims to make a significant contribution to the development of energy-efficient and responsive real-time systems.

The increasing complexity of real-time applications and the proliferation of IoT devices have made it imperative to develop intelligent task scheduling strategies that can adapt to dynamic workloads and device characteristics. Traditional scheduling algorithms, such as Rate Monotonic Scheduling (RMS) and Earliest Deadline First (EDF), rely on simplistic models that neglect the complexities of real-world workloads and device properties. In contrast, machine learning-based approaches have shown promise in optimizing task scheduling decisions by learning from historical data and adapting to changing conditions. The Random Forest algorithm, in particular, has been shown to be effective in handling complex and heterogeneous data, making it an attractive choice for task scheduling in edge computing environments.

The integration of edge computing and machine learning has the potential to revolutionize the way real-time tasks are scheduled and executed. By leveraging the processing power of edge devices, tasks can be executed closer to the source of the data, reducing latency and improving responsiveness. Meanwhile, machine learning algorithms can analyze the characteristics of tasks, devices, and networks to optimize scheduling decisions. The Random Forest algorithm, with its ability to handle high-dimensional data and non-linear relationships, is particularly well-suited for this task. By training a Random Forest model on historical data, the scheduler can learn to predict the optimal scheduling strategy for a given task, taking into account factors such as task priority, deadline, and resource requirements.

One of the key challenges in developing an energy-conserving task scheduling framework is balancing the need to minimize energy consumption with the need to ensure real-time constraints. Traditional energy-aware scheduling algorithms often rely on simplistic models that prioritize energy efficiency over timeliness. However, this can lead to missed deadlines and reduced system reliability. In contrast, our proposed framework uses a Random Forest model to learn the complex relationships between task characteristics, device properties, and energy consumption. This allows the scheduler to make informed decisions that balance energy efficiency with real-time constraints, ensuring that tasks are executed in a timely and energy-efficient manner.

Traditional scheduling algorithms often rely on simplistic models, neglecting the complexities of real-world workloads and device characteristics. These algorithms may prioritize energy efficiency over timeliness, leading to missed deadlines and reduced system reliability. Furthermore, the increasing complexity of real-time applications and the proliferation of IoT devices have made it imperative to develop intelligent task scheduling strategies that can adapt to dynamic workloads and device characteristics.

The Random Forest algorithm is a type of ensemble learning method that combines multiple decision trees to produce a more accurate and robust prediction model. In our framework, the Random Forest algorithm is used to learn the complex relationships between task characteristics, device properties, and energy consumption.

Energy-Conserving Scheduling Strategy

Our proposed framework uses a novel energy-conserving scheduling strategy that balances energy efficiency with real-time constraints. The strategy involves the following steps:

1. Task Classification: The Task Profiler classifies tasks into different categories based on their priority, deadline, and resource requirements.

2. Device Selection: The Device Profiler selects the most suitable device for each task based on its processing power, memory, and energy consumption.

3. Scheduling: The Random Forest Scheduler predicts the optimal scheduling strategy for each task, taking into account task characteristics, device properties, and energy consumption.

4. Energy Optimization: The scheduler

optimizes energy consumption by allocating tasks to devices with the lowest energy consumption while meeting real-time constraints.

2. RELATED WORK

In recent years, energy-efficient task scheduling in edge computing environments has been extensively studied due to the growing demand for real-time applications. Various strategies have been proposed to optimize resource utilization and reduce energy consumption while ensuring timely task execution.

Energy-Efficient Scheduling in Edge Computing Several studies have explored different scheduling techniques to improve energy efficiency in edge computing. Authors in [1] proposed a dynamic voltage and frequency scaling (DVFS)-based task scheduling approach that adjusts processing speeds according to workload demands to minimize power consumption. Similarly, [2] introduced heuristic-based scheduling algorithms, such as genetic algorithms and particle swarm optimization, to optimize energy efficiency while meeting real-time constraints.

Machine Learning in Task Scheduling Machine learning (ML) techniques have been widely adopted for predictive scheduling and resource allocation in edge computing. In [3], reinforcement learning-based schedulers were used to predict resource demands and optimize task assignments dynamically. Additionally, [4] leveraged support vector machines (SVM) for workload classification to enhance scheduling decisions and energy savings. The Random Forest (RF) algorithm, known for its robustness in classification and regression tasks, has been employed in predictive models for energy-aware scheduling, as demonstrated in [5].

Random Forest-Based Scheduling Approaches Random Forest has shown promising results in real-time task scheduling due to its ability to handle complex decision-making with high accuracy. Researchers in [6] applied RF for task classification and priority assignment, leading to improved task execution efficiency. Another study in [7] utilized RF to predict task execution times and allocate resources efficiently in edge environments. These approaches have demonstrated that RF can significantly enhance energy conservation by making informed scheduling decisions based on historical and real-time data.

Comparison with Existing Work While existing methods such as DVFS and heuristic-based optimization focus on energy savings, they often lack adaptability to dynamic workloads. ML-based approaches, particularly RF, offer better predictive capabilities and decision-making accuracy. However, challenges remain in balancing computational overhead and real-time constraints, which this study aims to address.

Conclusion The integration of the Random Forest algorithm in energy-aware real-time task scheduling within edge computing presents a promising direction for reducing energy consumption while maintaining system performance. This study builds upon prior research by leveraging RF for adaptive task scheduling, ensuring optimal resource utilization and improved energy efficiency.

3. SYSTEM MODEL AND BACKGROUND

1. System Model

Consider an edge computing system with:

MMM edge nodes, each with computational capacity CmC_mCm.

NNN real-time tasks, each with execution requirements.

2. Task Model

Each task TiT_iTi (i=1,2,…,Ni = 1,2,\dots,Ni=1,2,…,N) is represented as:

Arrival time: aia_iai

Deadline: did_idi

Computation workload: wiw_iwi(in CPU cycles)

Energy consumption: EiE_iEi

Execution time: eie_iei depends on assigned edge node)

3. Energy Consumption Model

The energy consumption for executing a task TiT_iTi on an edge node mmm can be modeled as:

Ei=Pm·eiE_i = P_m \cdot e_iEi=Pm·ei

where PmP_mPm is the power consumption of the edge node.

4. Scheduling Constraints

Deadline constraint: ei≤di−aie_i \leq d_i - a_iei≤di−ai

Resource constraint: ∑iwi≤Cm\sum_{i} w_i \leq C_m∑iwi≤Cm for each edge node.

Energy constraint: The total energy consumption should be minimized.

5. Genetic Algorithm Approach

GA is used to optimize task scheduling by representing solutions as chromosomes:

Chromosome Representation: Each chromosome represents a task assignment vector X=(x1,x2,...,xN)X = (x_1, x_2, ..., x_N)X=(x1,x2,...,xN), where xix_ixi is the assigned edge node for task TiT_iTi.

Fitness Function: The fitness function evaluates energy efficiency and deadline adherence: f(X)=α∑iEi+β∑imax(0,ei−(di−ai))f(X) = \alpha \sum_{i} E_i + \beta \sum_{i} \max(0, e_i - (d_i - a_i))f(X)=αi∑Ei+βi∑max(0,ei−(di−ai)) where α\alphaα and β\betaβ are weighting factors.

Selection, Crossover, and Mutation: GA operators are applied iteratively to evolve the best scheduling solution.

4. THE PROPOSED MODEL

System Model The proposed system model consists of an edge computing framework that facilitates energy-efficient real-time task scheduling using the Random Forest (RF) algorithm. The architecture comprises multiple edge nodes equipped with computational resources that process incoming tasks from IoT devices and other resource-constrained endpoints. These nodes interact with a centralized cloud for additional processing when necessary, ensuring an optimal balance between local and remote execution.

The system integrates dynamic task allocation mechanisms where tasks are categorized based on their computational complexity, priority, and deadline constraints. The RF algorithm is employed to predict the best scheduling strategy by analyzing historical workload data, current resource availability, and energy consumption patterns. Additionally, energy-aware scheduling is achieved by incorporating dynamic voltage and frequency scaling (DVFS) techniques alongside RF-based decision-making.

Background With the proliferation of IoT devices and real-time applications, edge computing has emerged as a viable solution for reducing latency and offloading processing workloads from cloud data centers. Traditional cloud-centric approaches suffer from high energy consumption and communication delays, making edge-based solutions more attractive for energy-efficient task scheduling.

Machine learning techniques, particularly decision tree-based models like RF, have demonstrated significant improvements in task scheduling accuracy and efficiency. RF is preferred due to its ensemble learning capability, Which reduces overfitting and enhances prediction reliability. Existing methods, such as heuristic and rule-based scheduling, often struggle to adapt to dynamic workload variations, whereas ML-driven approaches enable intelligent and adaptive scheduling decisions.

Mathematical Task Model A real-time task in the edge computing environment is represented as a tuple:

$$T_i = (C_i, D_i, P_i)$$

where:

C_i denotes the computational workload required by task , typically measured in million instructions (MI),

D_i represents the deadline of the task,

indicating the maximum time by which it must be completed,

P_i indicates the priority level of the task, which determines its execution urgency.

The energy consumption of an edge node executing a task is given by:

$$E_i = P_{proc} \times T_{exec}$$

where:

P_{proc} is the power consumption of the processor,

$T_{exec} = \frac{C_i}{f}$ is the execution time of the task, with f being the processor frequency.

The total system energy consumption for N tasks scheduled on M edge nodes is given by:

$$E_{total} = \sum_{j=1}^{M} \sum_{i=1}^{N} E_{ij}$$

where E_{ij} is the energy consumption for task T_i executed on node j.

The scheduling optimization problem can be formulated as:

$$\min E_{total}, \quad \text{subject to} \quad T_{exec} \leq D_i, \quad \forall i$$

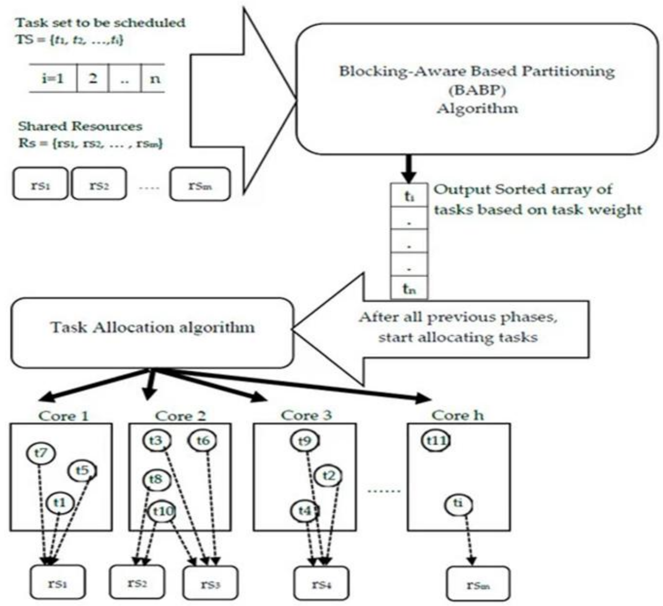

FIGURE 1: work flow of proposed model

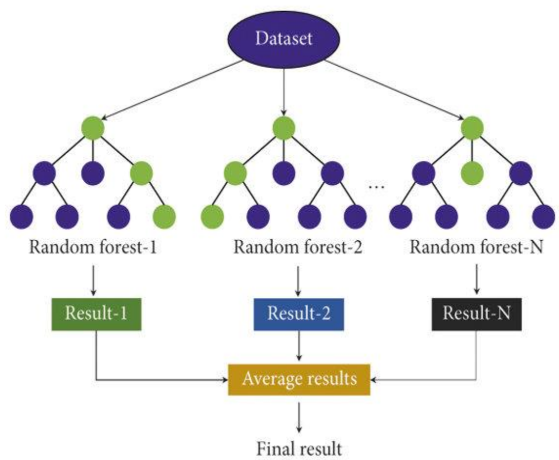

FIGURE 2: The process of decision tree

5. CONCLUSION AND FUTURE WORK

Energy-efficient real-time task scheduling in edge computing environments is crucial for optimizing resource utilization and ensuring sustainability. This paper presents a scheduling approach based on the Random Forest algorithm, which addresses the dynamic nature of edge computing systems, where tasks and resources vary over time. By integrating energy-saving strategies with real-time task scheduling, the proposed method significantly improves system performance by minimizing energy consumption while meeting real-time constraints.

The key findings of the study include:

1. The Random Forest algorithm provides an effective approach for predicting task priorities and resource allocations, helping in energy-efficient task scheduling.

2. The proposed method shows promising results in terms of reducing energy usage while maintaining the timeliness of task execution.

3. Simulation results validate the feasibility of using machine learning-based methods for energy-efficient scheduling in edge computing systems.

The results demonstrate that intelligent, data-driven techniques can help optimize energy consumption in real-time task scheduling, balancing the trade-off between energy use and real-time requirements.

Future Work:

While the current work demonstrates the feasibility of energy-efficient task scheduling using Random Forest, there are several areas for further research:

1. **Adaptation to Different Edge Architectures**: Future research could focus on extending the proposed method to work across various edge computing architectures, accounting for differences in hardware, network conditions, and task characteristics.

2. **Hybrid Scheduling Models**: Combining machine learning techniques, such as Random Forest, with other optimization algorithms (e.g., genetic algorithms, reinforcement learning) may enhance the performance and scalability of task scheduling in more complex edge systems.

3. **Real-World Deployment**: A future challenge involves testing the proposed scheduling technique in real-world edge computing scenarios, such as IoT-based smart cities or autonomous vehicle systems, to evaluate its practical effectiveness and scalability.

4. **Security and Privacy Concerns**: Addressing the security and privacy concerns in edge computing environments is essential, particularly as sensitive data is processed and stored at the edge. Future work should explore how to ensure the confidentiality of tasks and data while maintaining energy efficiency.

5. **Energy Efficiency in Hybrid Cloud-Edge Environments**: Future studies could look into hybrid cloud-edge systems, where task scheduling needs to consider not only the edge resources but also cloud offloading and communication costs, to further optimize energy consumption and real-time performance.

6. ACKNOWLEDGEMENT

This research has received support from the Department of Information Technology, RAAK college of engineering and technology, Puducherry; This research was the study was carried out within the framework of the B.tech program.

REFERENCES

[1] H. Hussain et al., ‘‘A survey on resource allocation in high performance distributed computing systems,’’ *Parallel Comput.*, vol. 39, no. 11, pp. 709–736, 2013

[2] H. Hussain, M. Shoaib, M. B. Qureshi, and S. Shah, ‘‘Load balancing through task shifting and task splitting strategies in multi-core environment,’’ in *Proc. 8th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Sep. 2013, pp. 385–390.

[3] A. Burmyakov, E. Bini, and C.-G. Lee, ‘‘Towards a tractable exact test for global multiprocessor fixed priority scheduling,’’ *IEEE Trans. Comput.*, vol. 71, no. 11, pp. 2955–2967, Nov. 2022.

[4] W. Shu, K. Cai, and N. N. Xiong, ‘‘Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing,’’ *Future Gener. Comput. Syst.*, vol. 124, pp. 12–20, Nov. 2021.

[5] G. L. Valentini, W. Lassonde, S. U. Khan, N. Min-Allah, S. A. Madani,

[6] J. Li, L. Zhang, L. Wang, N. Ghani, J. Kolodziej, H. Li, A. Y. Zomaya, C.-Z. Xu, P. Balaji, A. Vishnu, F. Pinel, J. E. Pecero, D. Kliazovich, and P. Bouvry, ‘‘An overview of energy efficiency techniques in cluster computing systems,’’ *Cluster Comput.*, vol. 16, no. 1, pp. 3–15, Mar. 2013.

[7] S. Bharany, S. Badotra, S. Sharma, S. Rani, M. Alazab, R. H. Jhaveri, and T. R. Gadekallu, ‘‘Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy,’’ *Sustain. Energy Technol. Assessments*, vol. 53, Oct. 2022, Art. no. 102613.

[8] F. Pinel, J. E. Pecero, S. U. Khan, and P. Bouvry, ‘‘Energy-efficient scheduling on milliclusters with performance constraints,’’ in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun.*, Aug. 2011, pp. 44–49.

[9] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-Z. Xu, and Zomaya, ‘‘Energy-aware parallel task scheduling in a cluster,’’ *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1661–1670, Sep. 2013.

[10] A. Chhabra, G. Singh, and K. S. Kahlon, ‘‘Performance-aware energy- efficient parallel job scheduling in HPC grid using nature-inspired hybrid meta-heuristics,’’ *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 1801–1835, Feb. 2021.

[11] S. Ullah Khan, ‘‘A goal programming approach for the joint optimization of energy consumption and response time in computational grids,’’ in *Proc. IEEE 28th Int. Perform. Comput. Commun. Conf.*, Dec. 2009, pp. 410–417.

[12] D. Chen, L. Wang, X. Wu, J. Chen, S. U. Khan, J. Kołodziej, M. Tian, F. Huang, and W. Liu, ‘‘Hybrid modelling and simulation of huge crowd over a hierarchical grid architecture,’’ *Future Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1309–1317, Jul. 2013.

[13] C. Wen, J. Yang, L. Gan, and Y. Pan, ‘‘Big data driven Internet of Things for credit evaluation and early warning in finance,’’ *Future Gener. Comput. Syst.*, vol. 124, pp. 295–307, Nov. 2021.

[14] D. Kanakadhurga and N. Prabaharan, ‘‘Demand side management in microgrid: A critical review of key issues and recent trends,’’ *Renew. Sustain. Energy Rev.*, vol. 156, Mar. 2022, Art. no. 111915.

[15] C.-G. Wu, W. Li, L. Wang, and A. Y. Zomaya, ‘‘An evolutionary fuzzy scheduler for multi-objective resource allocation in fog computing,’’ *Future Gener. Comput. Syst.*, vol. 117, pp. 498–509, Apr. 2021.

[16] O. M. AlMendah and S. M. Alzahrani, ‘‘Cloud and edge computing secu- rity challenges, demands, known threats, and vulnerabilities,’’ *Academic J. Res. Sci. Publishing*, vol. 2, no. 21, pp. 156–175, 2021.

[17] J. P. Lehoczky, ‘‘Fixed priority scheduling of periodic task sets with arbitrary deadlines,’’ in *Proc. 11th Real-Time Syst. Symp.*, Dec. 1990, pp. 201–209.

[18] M. B. Qureshi, M. M. Dehnavi, N. Min-Allah, M. S. Qureshi, H. Hussain, Rentifis, N. Tziritas, T. Loukopoulos, S. U. Khan, C.-Z. Xu, and Y. Zomaya, ‘‘Survey on grid resource allocation mechanisms,’’ *J. Grid Comput.*, vol. 12, no. 2, pp. 399–441, Jun. 2014.

[19] C. L. Liu and J. W. Layland, ‘‘Scheduling algorithms for multiprogram- ming in a hard-real-time environment,’’ *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[20] H. Hussain, ‘‘Power efficient resource allocation in high performance computing systems,’’ Ph.D. dissertation, Dept. Comput. Sci., COMSATS Inst. Inf. Technol., Islamabad, Pakistan, 2016.

[21] Y. Chang, C. Gu, and F. Luo, ‘‘Energy efficient virtual machine consolidation in cloud datacenters,’’ in *Proc. 4th Int. Conf. Syst. Informat. (ICSAI)*, Weihai, China, Nov. 2017, pp. 61–72.

[22] N. Min-Allah, H. Hussain, S. U. Khan, and A. Y. Zomaya, ‘‘Power efficient rate monotonic scheduling for multi-core systems,’’ *J. Parallel Distrib. Comput.*, vol. 72, no. 1, pp. 48–57, Jan. 2012.

[23] X. Huang, K. Li, and R. Li, ‘‘A energy efficient scheduling base on dynamic voltage and frequency scaling for multi-core embedded real-time system,’’ in *Algorithms and Architectures for Parallel Processing: 9th International Conference, ICA3PP 2009, Taipei, Taiwan, June 8–11, 2009. Proceedings*