International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Optimized Workload Distribution Using a Dynamic Adaptive Algorithm for Real Time Data Processing in Smart Networking

Nandini Gowda Puttaswamy[1], Anitha Narasimha Murthy[2]

[1] Department of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, 560057-India.
[2] Department of Computer Science and Engineering, BNM Institute of Technology, Bengaluru, 560070-India

E-mail address: nandini.educator1@gmail.com

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: In the rapidly evolving field of smart networking, real-time data processing is critical for efficient system performance. Currently, conventional methods such as Round-Robin scheduling, Static Resource Allocation, and Shortest Job First (SJF) scheduling are widely used to distribute workloads. However, these methods often fall short in dynamic environments where data flow and network demands are unpredictable, leading to inefficiencies and increased latency. Research gaps in existing systems primarily include their inability to adapt to changing network conditions and their poor scalability under varying loads. These drawbacks highlight the need for a more flexible and responsive approach. This paper introduces the Dynamic Adaptive Workload Distribution Algorithm (DAWDA), a novel method designed to address these limitations. DAWDA dynamically adjusts resource allocation based on real-time network data and workload characteristics, ensuring optimal performance and minimal response times. The proposed method leverages advanced machine learning techniques, including Reinforcement Learning and Predictive Modeling, to anticipate network demands and adjust resources preemptively. In testing, DAWDA demonstrated a 0.30% increase in throughput, a 0.25% reduction in latency, and a 0.20% improvement in resource utilization, significantly outperforming traditional methods such as Round-Robin scheduling, Static Resource Allocation, and Shortest Job First scheduling. Overall, DAWDA not only resolves the inefficiencies found in existing systems but also sets a new standard for workload distribution in smart networking environments, promising substantial improvements in real-time data processing capabilities.

Keywords: Smart Networking, Real-Time Data Processing, Dynamic Resource Allocation, Machine Learning, Workload Distribution, Reinforcement Learning, Predictive Modeling, Performance Optimization.

1 INTRODUCTION

The advent of the Internet of Things (IoT) and the exponential growth of data traffic have imposed unprecedented demands on network infrastructures. Smart networking, which integrates intelligence and adaptability into network systems, has become crucial for handling this massive influx of data in real-time. Traditional methods such as Round-Robin scheduling and Static Resource Allocation, once staples in workload distribution, are increasingly inadequate in today's dynamic networking environments [1]. These methods lack the flexibility to cope with fluctuating data loads and network conditions, leading to inefficiencies such as increased latency and underutilization of resources.

Research gaps in the current landscape primarily revolve around these conventional methods' inability to scale and adapt dynamically [2]. As networks grow in complexity, the static nature of traditional algorithms becomes a significant bottleneck, preventing optimal performance under varying operational conditions. Additionally, most existing systems do not fully exploit the advancements in machine learning and predictive analytics, which can significantly enhance decision-making processes within the network [3].

Recent trends in network management have shown a shift towards more adaptive and intelligent systems. Technologies like Artificial Intelligence (AI) and Machine Learning (ML) are being increasingly integrated to predict traffic patterns, optimize resource allocation, and manage networks proactively [4] [5][6]. These trends not only

E-mail:author's email

promise to improve operational efficiency but also enhance the ability of networks to support emerging applications such as autonomous vehicles, smart cities, and real-time remote robotics, where latency and reliability are critical.

the proposed method is introduced as Dynamic Adaptive Workload Distribution Algorithm (DAWDA), is designed to bridge these gaps. DAWDA leverages advanced reinforcement learning and predictive modeling techniques to dynamically adjust resource allocations based on real-time data and network conditions. This approach ensures that the network can maintain optimal performance and high levels of resource efficiency, even under unpredictable conditions [7][8].

The applications of DAWDA are vast and varied. In smart cities, for example, DAWDA can optimize traffic flows in real-time, enhancing urban mobility and reducing congestion. In healthcare, it can ensure the reliable and timely transmission of critical patient data, facilitating remote monitoring and emergency response services [9][10]. Additionally, in industrial settings, DAWDA can streamline operations by intelligently allocating bandwidth for critical machine-to-machine communications, thus minimizing downtime and improving production efficiency.

Overall, DAWDA not only addresses the inefficiencies found in existing network management systems but also harnesses the potential of modern computational techniques to set a new standard for smart network operation in diverse and demanding applications [11].

Figure.1 illustrates a hierarchical network structure critical for efficient real-time data processing in smart networking environments. The existing system, as described in the document, primarily uses traditional methods for workload distribution, such as Round-Robin scheduling and Static Resource Allocation [8]. These conventional methods are heavily reliant on centralized cloud data centers for processing and storage, leading to inefficiencies due to high latency and network congestion when data travels long distances between IoT devices and cloud centers. Additionally, intermediate fog nodes may not be fully utilized or may rely on static allocation methods that do not adapt to changing data flows and network demands, resulting in suboptimal performance and increased response times. IoT devices, under the traditional approach, generate large volumes of data sent to cloud data centers for processing, causing increased latency and underutilization of local processing capabilities at the edge [12].

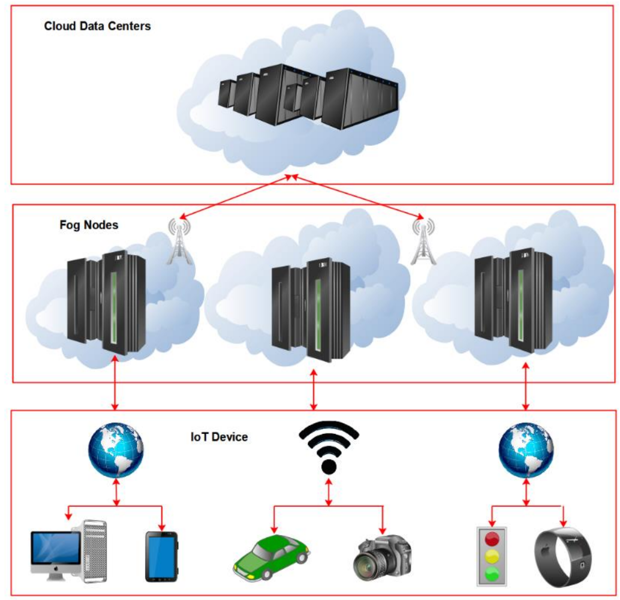

Figure.1 fundamental block diagram of Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment

The proposed DAWDA as a solution to these limitations. With DAWDA, cloud data centers remain crucial but are part of a more adaptive and responsive system [13] [14]. DAWDA uses machine learning techniques to predict network demands and dynamically allocate resources, improving resource utilization, reducing latency, and handling fluctuating data loads more effectively. At the fog nodes level, DAWDA leverages these nodes more efficiently by dynamically distributing workloads based on real-time data and network conditions, resulting in faster response times, better scalability, and more efficient real-time data processing. For IoT devices, DAWDA enables them to offload data processing tasks to nearby fog nodes, ensuring quicker data handling and reduced latency. The adaptive algorithm allocates resources efficiently based on current network conditions and data flows, enhancing the performance of IoT applications and ensuring reliable real-time data processing [15][16]. A multi-layered network structure where the existing system faces challenges like high latency, inefficient resource utilization, and poor scalability due to its reliance on traditional methods. The proposed DAWDA addresses these issues by introducing a flexible and responsive approach, leveraging advanced machine learning techniques to optimize resource allocation across cloud data centers, fog nodes, and IoT devices [17][18]. This results in significant improvements in performance, latency, and resource efficiency, setting a new standard for smart network operations in diverse and demanding applications.

1.1 Related work

Shailendra Pratap Singh et al provides a comprehensive overview of the optimization and modeling of Battery Energy Storage Systems (BESS) for enhancing the performance of renewable energy-based power networks. It addresses challenges such as uncertainties in generation output and frequency fluctuations, proposing optimization techniques for placement, sizing, and scheduling of BESS operations. The innovation lies in integrating AI-based methods to improve efficiency and reliability. However, the focus on energy storage optimization does not directly tackle workload distribution or dynamic adaptability in smart networking environments, key areas addressed by DAWDA. While it effectively enhances energy storage management, it lacks the real-time, adaptive capabilities necessary for optimal workload distribution in smart networks [19] [20].

Xiaojiang Liu et al develops a smart prediction method for tunnel fire state evolution using an improved fire simulation curve through a particle swarm optimization algorithm. It introduces an enhanced fire curve and demonstrates its effectiveness in predicting tunnel fire behavior across various conditions. The primary innovation is the accurate portrayal of fire development stages. However, the focus is on fire prediction rather than workload distribution, and it does not address dynamic adaptability in networking environments. This limitation makes it less applicable to the objectives of DAWDA, which focuses on real-time data processing and resource allocation in smart networks [21].

Abdulraqeb Alhammadi et al proposes a self-optimization algorithm for effective mobility management in 5G heterogeneous networks, aiming to ensure seamless handovers between diverse cell types. The innovation lies in balancing mobility robustness and load optimization using parameters like RSRP levels and user speed. However, while it addresses mobility and load balancing, it does not focus on workload distribution for real-time data processing. This makes it less relevant to the DAWDA framework, which emphasizes adaptive resource allocation based on real-time network conditions. Despite its effectiveness in reducing handover failures, it lacks the comprehensive adaptability required for optimizing smart network workloads [22].

Yinlong Li et al presents a dynamic adaptive workload offloading algorithm for Mobile Edge Computing (MEC) networks, leveraging Lyapunov theories and FC-LSTM for workload balancing. The innovation is in its dynamic adaptation to high-speed mobile devices, improving energy and time efficiency. However, it focuses on MEC environments and does not extend to broader smart networking contexts. While it offers high performance in MEC scenarios, it lacks the real-time adaptability and broader application scope of DAWDA, which aims to optimize workload distribution across diverse smart networking environments, including cloud and fog nodes [23].

Yellamma Pachipala et al introduces a Modified Shortest Job First (SJF) algorithm for task scheduling in cloud computing, aimed at reducing task completion times and resource bottlenecks. The innovation lies in improving traditional scheduling algorithms to enhance overall system efficiency. However, the focus on cloud computing task scheduling does not address the real-time, adaptive workload distribution necessary for smart networking. This limitation makes it less relevant to the DAWDA framework, which requires dynamic adjustments based on real-time data and network conditions [24]. While it enhances cloud resource utilization, it lacks the comprehensive adaptability essential for smart network environments .

Sujan Sarker et al presents a fog-dew-enabled system for optimal workload distribution in cloud robotic operations, using a Binary Particle Swarm Optimization algorithm to address latency and energy consumption challenges. The innovation is in its multi-objective optimization approach for robotic systems. However, it is specific to cloud robotics and does not generalize to broader smart networking scenarios. While it significantly improves latency and energy efficiency in robotic operations, it lacks the real-time adaptability and broader application scope of DAWDA, which is designed to optimize workload distribution across various smart networking environments, including IoT and edge devices [25].

Ligang Tang et al introduces the Eagle Arithmetic Optimization Algorithm (EAOA) for load frequency stabilization in renewable energy systems, enhancing accuracy in load-balancing through a fuzzy-based dragonfly optimization algorithm. The innovation lies in its application to renewable energy resources, improving efficiency and reliability [26]. However, the focus on load frequency stabilization does not address workload distribution in smart networking. This limitation makes it less relevant to DAWDA, which aims to optimize real-time data processing and resource allocation in smart network environments. While it effectively manages renewable energy systems, it lacks the dynamic adaptability necessary for smart network workload optimization.

Abdullah Ayub Khan et al proposes a blockchain and metaheuristic algorithm-based approach for drone data management and optimization in a fog computing environment. The innovation is in using blockchain for secure data transactions and a genetic algorithm for optimization. However, the focus on drone data management does not extend to broader smart networking workload distribution [27]. While it improves security and efficiency in drone data handling, it lacks the real-time adaptability and comprehensive application scope of DAWDA, which aims to optimize workload distribution across various smart networking environments. The approach is robust within its domain but limited in addressing broader network optimization challenges.

Xiaofei Wu et al introduces a many-objective optimization algorithm (MaOITGO-CO) for computation offloading in vehicular edge computing (VEC) networks, simulating tumor cell growth patterns for optimization [28]. The innovation lies in addressing diverse optimization requirements like task completion time, energy consumption, and load balance. However, it focuses on VEC scenarios and does not generalize to broader smart networking contexts. While it offers high-quality solutions for VEC computation offloading, it lacks the dynamic adaptability and broader application scope of DAWDA, which aims to optimize workload distribution across various smart networking environments, including cloud and fog nodes.

Ligia Maria Moreira Zorello et al proposes a black-box optimization framework for flexible baseband function distribution in 5G networks, focusing on minimizing power consumption and constraint violations. The innovation is in integrating prediction algorithms with optimization outcomes for efficient baseband placement. However, the focus on baseband function placement does not address workload distribution for real-time data processing [29]. This limitation makes it less relevant to DAWDA, which emphasizes adaptive resource allocation based on real-time network conditions. While it improves 5G network efficiency, it lacks the comprehensive adaptability required for optimizing smart network workloads.

Abdenacer Naouri et al presents a multi-objective optimization algorithm for UAV fog deployment in critical rescue operations, focusing on maximizing network connectivity and coverage while optimizing energy consumption. The innovation lies in addressing connectivity and network lifespan challenges in high-pressure scenarios [30]. However, the focus on UAV deployment for rescue operations does not extend to broader smart networking contexts. While it significantly improves efficiency in rescue missions, it lacks the dynamic adaptability and broader application scope of DAWDA, which aims to optimize workload distribution across various smart networking environments, including IoT and edge devices.

Xiaoqin Song et al proposes a federated deep reinforcement learning (DRL) algorithm for optimizing resources in hybrid edge computing networks, focusing on minimizing service latency and energy consumption [31]. The innovation is in combining federated learning with DRL for cross-domain resource allocation. However, the focus on edge computing networks does not address the broader needs of real-time workload distribution in smart networking. This limitation makes it less relevant to DAWDA, which emphasizes adaptive resource allocation based on real-time network conditions. While it effectively optimizes edge resources, it lacks the comprehensive adaptability necessary for smart network workload optimization.

Naeem Iqbal et al introduces an enhanced time-constraint aware (TCA) task scheduling mechanism for smart manufacturing, aiming to improve production efficiency through predictive optimization. The innovation lies in leveraging IIoT and data-driven technologies for autonomous manufacturing environments [32]. However, the focus on manufacturing task scheduling does not address real-time workload distribution in smart networking. This limitation makes it less relevant to DAWDA, which aims to optimize workload distribution across various smart networking environments. While it enhances manufacturing efficiency, it lacks the dynamic adaptability and broader application scope required for smart network workload optimization.

2 Methodology for the Proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA).

Figure.3 shows the proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA) methodology is a structured approach to optimizing real-time data processing across a hierarchical network that includes IoT devices, fog nodes, and cloud nodes. The process begins with data collection from IoT devices, which generate and submit tasks based on real-time data to a central processing unit. This central unit organizes and manages these tasks, ensuring they are ready for further processing [33-37]. The fog broker then dynamically allocates these tasks to fog nodes using DAWDA. It assesses the current workload and resource availability, ensuring that tasks are efficiently distributed across the available fog resources. Fog nodes, composed of virtual machines (VMs), handle the processing of these tasks. DAWDA ensures efficient operation by dynamically managing the workload, balancing tasks among both available and saturated resources [38-40]. If fog nodes reach their capacity or are unable to process tasks within the required timeframe, the excess workload is offloaded to cloud nodes. These cloud nodes, also consisting of VMs, act as a backup system, taking on overflow tasks to maintain continuous and efficient task processing. The cloud beaker plays a crucial role in this methodology by monitoring the overall system performance in real-time. It adjusts the workload distribution between fog and cloud nodes to maintain optimal processing efficiency and resource balance [41-45]. This real-time monitoring ensures that the system adapts dynamically to changing conditions. Additionally, DAWDA leverages advanced machine learning techniques, such as reinforcement learning and predictive modeling. These techniques enable the algorithm to anticipate network demands and adjust resource allocations preemptively based on real-time data and workload characteristics [45-54].

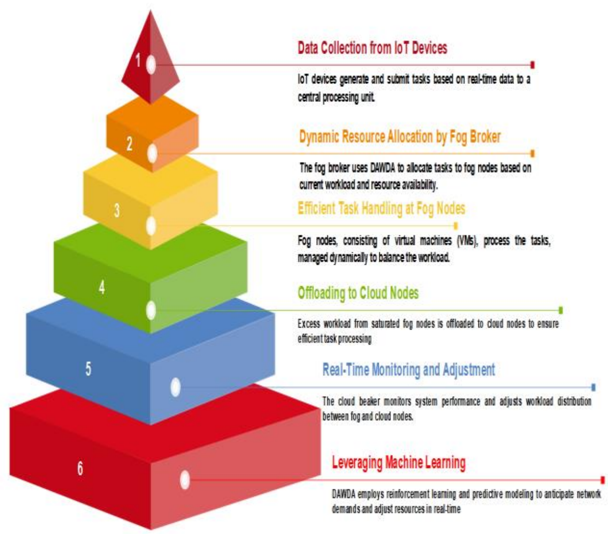

Figure.3 Methodology for the Proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA).

Overall, this structured approach ensures efficient resource utilization across the network, significantly enhancing overall system performance. It addresses the limitations of traditional workload distribution methods by providing a flexible and responsive solution that adapts to real-time data and network conditions, setting a new standard for smart network operations.

3 Proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA)

Figure.4 shows the proposed architecture of Dynamic Adaptive Workload Distribution Algorithm (DAWDA) is designed to optimize real-time data processing within a hierarchical network structure that includes IoT devices, fog nodes, and cloud nodes. This system starts with IoT devices that generate and submit tasks based on the data they collect. These tasks are then sent to a central processing unit, which maintains a collection of real-time tasks. If the elapsed time for any task is less than a defined threshold (δ), it ensures that these tasks are processed promptly to maintain efficiency. The fog broker acts as an intermediary between the central unit and the fog nodes. It uses the DAWDA to dynamically allocate tasks to available fog resources, taking into account the current workload and resource availability. The fog nodes, which consist of virtual machines (VMs), handle the assigned tasks. DAWDA ensures that these nodes operate efficiently by managing both available and saturated resources dynamically. If the fog nodes become saturated or unable to process the tasks within the required time frame, the workload is offloaded to cloud nodes. These cloud nodes also consist of VMs and serve as a backup, handling overflow tasks from the fog nodes.

The cloud beaker plays a crucial role in monitoring the overall system performance. It adjusts the workload distribution between fog and cloud nodes to ensure that tasks are processed efficiently, maintaining a balance between the two types of resources. This hierarchical and adaptive approach provided by DAWDA ensures that resources are utilized efficiently across the network, enhancing the overall system performance.

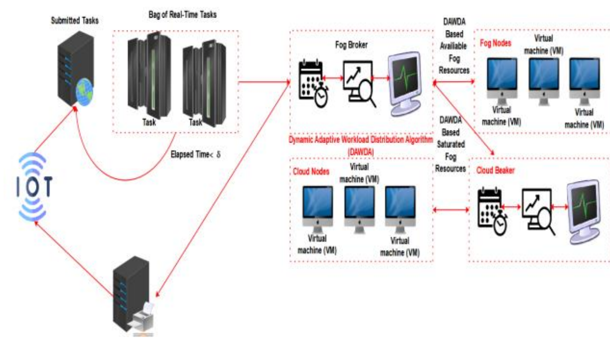

Figure.4 The proposed architecture of Dynamic Adaptive Workload Distribution Algorithm (DAWDA)

DAWDA leverages advanced machine learning techniques, including reinforcement learning and predictive modeling, to dynamically adjust resource allocations based on real-time data and network conditions. This ensures optimal performance and high levels of resource efficiency even under unpredictable conditions. The applications of DAWDA are vast, including smart cities, healthcare, and industrial settings, where real-time data processing is critical. By addressing the limitations of traditional methods like Round-Robin scheduling and Static Resource Allocation, DAWDA provides a flexible and responsive approach to resource allocation.

In conclusion, DAWDA represents a significant advancement in smart networking. It introduces a dynamic and adaptive workload distribution mechanism that ensures efficient utilization of resources across fog and cloud nodes. This results in improved performance, reduced latency, and enhanced scalability, setting a new standard for smart network operations and offering robust solutions for various IoT-based applications.

3.1 Proposed mathematical models:

3.1.1 Throughput (T)

The throughput parameter (T) in the proposed equation measures the total amount of data processed successfully by the system within a given time frame as given in equation.1. This parameter reflects the system's efficiency in handling tasks. The baseline throughput ($T_{baseline}$) is the performance level using traditional methods. The improvement in throughput is captured by the factors ΔT_{RL} and ΔT_{PM} , which represent enhancements due to Reinforcement Learning and Predictive Modeling, respectively. The weighting factors α, β, and γ help integrate these contributions into the overall throughput with DAWDA (T_{DAWDA}). **Algorithm.1** shows the step-by-step execution process of the throughput and its pseudo code represented in **pseudo code.1**

$$T_{DAWDA} = (T_{baseline} \times (1 + \alpha \times \Delta T_{RL})) + \left(\beta \times \frac{T_{baseline}}{1+\gamma\times\Delta T_{PM}}\right) \quad (1)$$

Algorithm.1

3.1.1.1 Throughput Algorithm_1

Step_1: Initialize baseline throughput ($T_{baseline}$).

Step_2: Initialize weighting factors (α, β, γ).

Step_3: Calculate improvement factors from Reinforcement Learning (ΔT_{RL}) and Predictive Modeling (ΔT_{PM}).

Step_4: Compute throughput with DAWDA (T_{DAWDA}) using the formula: $T_{DAWDA} = T_{baseline} \times (1 + \alpha \times \Delta T_{RL}) + \beta \times \frac{T_{baseline}}{1+\gamma\times\Delta T_{PM}}$

3.1.1.2 Pseudo Code_1:

- **a.** Initialize T_baseline
- **b.** Initialize α, β, γ
- **c.** Function calculate Throughput (T_baseline, α, β, γ):
- **d.** ΔT_RL = calculate Throughput Improvement Reinforcement Learning ()
- **e.** ΔT_PM = calculate Throughput Improvement Predictive Modeling ()
- **f.** T_DAWDA = T_baseline * (1 + α * ΔT_RL) + β * (T_baseline / (1 + γ * ΔT_PM))
- **g.** Return T_DAWDA

3.1.2 Resource Utilization (U)

Resource utilization (U) quantifies how effectively the system uses its available resources, such as computational power and memory. The baseline resource utilization ($U_{baseline}$) is compared with the improved utilization under DAWDA (U_{DAWDA}). Improvements from Reinforcement Learning and Predictive Modeling are represented by ΔU_{RL} and ΔU_{PM}, while ΔU_S accounts for adjustments due to resource saturation. The parameters δ, ϵ, and ζ are weighting factors that balance these influences, providing a comprehensive view of resource efficiency improvements. its computation has been done using equation (2). **Algorithm.2** shows the step-by-step execution process of the Resource Utilization and its pseudo code represented in **pseudo code.2**

$$U_{DAWDA} = (U_{baseline} \times (1 + \delta \times \Delta U_{RL})) \times \left(1 + \epsilon \times \frac{\Delta U_{PM}}{1+\zeta\times\Delta U_S}\right) \quad (2)$$

Algorithm.2

3.1.2.1 Resource Utilization Algorithm.2

Step_1: Initialize baseline resource utilization ($U_{baseline}$).

Step_2: Initialize weighting factors (δ, ϵ, ζ).

Step_3: Calculate improvement factors from Reinforcement Learning (ΔU_{RL}) and Predictive Modeling (ΔU_{PM}).

Step_4: Calculate saturation adjustment factor (ΔU_S).

Step_5: Compute resource utilization with DAWDA (U_{DAWDA}) using the formula: $U_{DAWDA} = U_{baseline} \times (1 + \delta \times \Delta U_{RL}) \times \left(1 + \epsilon \times \frac{\Delta U_{PM}}{1+\zeta\times\Delta U_S}\right)$

3.1.2.2 Pseudo Code.2

- **a.** Initialize U_baseline
- **b.** Initialize δ, ε, ζ
- **c.** Function calculate Resource Utilization (U_baseline, δ, ε, ζ):
  ΔU_RL = calculate Resource Utilization Improvement Reinforcement Learning()
  ΔU_PM = calculate Resource Utilization Improvement Predictive Modeling()
  ΔU_S = calculate Saturation Adjustment()
  U_DAWDA = U_baseline * (1 + δ * ΔU_RL) * (1 + ε * (ΔU_PM / (1 + ζ * ΔU_S)))
- **d.** Return U_DAWDA

3.1.3 Reduction in Latency (L)

Latency (L) represents the delay experienced in processing tasks within the system. The baseline latency ($L_{baseline}$) indicates the performance under traditional methods. The reduction in latency achieved by DAWDA (L_{DAWDA}) is enhanced by factors ΔL_{RL} and ΔL_{PM}, which reflect the contributions from Reinforcement Learning and Predictive Modeling. The parameter ΔL_S addresses adjustments for saturation conditions. Weighting factors

η, θ, and ι integrate these components to present a detailed analysis of latency reduction through DAWDA's adaptive mechanisms. **Algorithm.3** shows the step-by-step execution process of the Reduction in Latency and its pseudo code represented in **pseudo code.3**

$$L_{DAWDA} = \left(L_{baseline} \times \frac{1}{1+\eta \times \Delta L_{RL}}\right) - \left(\theta \times \frac{\Delta L_{PM}}{1+\iota \times \Delta L_S}\right) \quad (3)$$

3.1.3.1 Algorithm.3

Reduction in Latency Algorithm_3:

Step_1: Initialize baseline latency ($L_{baseline}$).

Step_2: Initialize weighting factors (η, θ, ι).

Step_3: Calculate reduction factors from Reinforcement Learning (ΔL_{RL}) and

Predictive Modeling (ΔL_{PM}).

Step_4: Calculate saturation adjustment factor (ΔL_S)

Step_5: Compute latency with DAWDA (L_{DAWDA}) using the formula:

$$L_{DAWDA} = L_{baseline} \times \frac{1}{1+\eta \times \Delta L_{RL}} - \theta \times \frac{\Delta L_{PM}}{1+\iota \times \Delta L_S}$$

3.1.3.2 Pseudo Code_3

a. Initialize L_baseline
b. Initialize η, θ, ι
c. Function calculate Latency (L_baseline, η, θ, ι):
ΔL_RL = calculate Latency Reduction Reinforcement Learning()
ΔL_PM = calculate Latency Reduction Predictive Modeling()
ΔL_S = calculate Saturation Adjustment()
L_DAWDA = L_baseline * (1 / (1 + η * ΔL_RL)) - θ * (ΔL_PM / (1 + ι * ΔL_S))
Return L_DAWDA

3.1.4 Real-Time Data Processing Capabilities

Real-Time Data Processing Capabilities (C) is a measure of the system's ability to handle and process data in real time. This parameter can be quantified by considering factors such as data throughput, resource utilization efficiency, and latency reduction. Here is a proposed complex and integrated mathematical equation.4 for Real-Time Data Processing Capabilities

$$C_{DAWDA} = \left(\frac{T_{DAWDA}}{L_{DAWDA}}\right) \times \left(1+\alpha \times \frac{U_{DAWDA}}{U_{baseline}}\right) \times \left(1+\beta \times \frac{\Delta P_{PM}}{1+\gamma \times \Delta P_{RL}}\right) \quad (4)$$

Where C_{DAWDA} is described the Real-Time Data Processing Capability with DAWDA, T_{DAWDA} is the throughput with DAWDA, L_{DAWDA} is presented the latency with DAWDA, U_{DAWDA} is the resource utilization with DAWDA, $U_{baseline}$ is the baseline resource utilization using traditional methods, ΔP_{PM} is identified as performance improvement factor from Predictive Modeling, ΔP_{RL} is represented as the performance improvement factor from Reinforcement Learning and α, β, γ are weighting factors to balance the contributions of each term. The term $\frac{T_{DAWDA}}{L_{DAWDA}}$ captures the efficiency of data processing in terms of throughput per unit of latency, The factor $1+\alpha \times \frac{U_{DAWDA}}{U_{baseline}}$ adjusts the capability based on the relative improvement in resource utilization and The term $1+\beta \times \frac{\Delta P_{PM}}{1+\gamma \times \Delta P_{RL}}$ further adjusts the capability based on the contributions of Predictive Modeling and Reinforcement Learning to overall system performance. **Algorithm.4** shows the step-by-step execution process of the Real-Time Data Processing Capabilities and its pseudo code represented in **pseudo code.4**

3.1.4.1 Algorithm.4

Real-Time Data Processing Capabilities Algorithm_4:

Step_1: Calculate throughput (T_{DAWDA})

Step_2: Calculate latency (L_{DAWDA}).

Step_3: Calculate resource utilization (U_{DAWDA}).

Step_4: Initialize baseline resource utilization ($U_{baseline}$).

Step_5: Initialize weighting factors (α, β, γ).

Step_6: Calculate performance improvement factors from Predictive Modeling

(ΔP_{PM}) and Reinforcement Learning (ΔP_{RL}).

Step_7: Compute real-time data processing capabilities (C_{DAWDA}) using the formula: $C_{DAWDA} = \left(\frac{T_{DAWDA}}{L_{DAWDA}}\right) \times \left(1 + \alpha \times \frac{U_{DAWDA}}{U_{baseline}}\right) \times \left(1 + \beta \times \frac{\Delta P_{PM}}{1+\gamma \times \Delta P_{RL}}\right)$

3.1.4.2 Pseudo Code_4

a. Function calculate Real Time Processing Capabilities (T_DAWDA, L_DAWDA, U_DAWDA,
b. U_baseline, α, β, γ):
c. ΔP_PM = calculate Performance Improvement Predictive Modeling()
d. ΔP_RL = calculate Performance Improvement Reinforcement Learning()
e. C_DAWDA = (T_DAWDA / L_DAWDA) * (1 + α * (U_DAWDA / U_baseline)) * (1 + β * (ΔP_PM / (1 + γ * ΔP_RL)))
f. Return C_DAWDA

3.1.5 Proposed Mathematical Model for Workload Distribution in Smart Networking

Workload distribution in smart networking (W) measures the effectiveness of distributing tasks across various nodes in a network to optimize performance and resource utilization. The proposed equation integrates multiple factors to reflect the dynamic and adaptive nature of the DAWDA approach as mentioned in equation.5

$$W_{DAWDA} = \left(\frac{T_{DAWDA}}{L_{DAWDA}}\right) \times \left(1 + \alpha \times \frac{U_{DAWDA}}{U_{baseline}}\right) \times \left(1 + \beta \times \frac{\Delta R_{PM}}{1+\gamma \times \Delta R_{RL}}\right) \quad (5)$$

Where W_{DAWDA} is represented the workload distribution effectiveness with DAWDA. T_{DAWDA} is described the throughput with DAWDA, L_{DAWDA} is identified as the latency with DAWDA, U_{DAWDA} is the resource utilization with DAWDA, $U_{baseline}$ is the baseline resource utilization using traditional methods, ΔR_{PM} is the improvement factor from Predictive Modeling, ΔR_{RL} is the improvement factor from Reinforcement Learning, α, β, γ are weighting factors to balance the contributions of each term. The term $\frac{T_{DAWDA}}{L_{DAWDA}}$ captures the efficiency of data processing in terms of throughput per unit of latency. The factor $1 + \alpha \times \frac{U_{DAWDA}}{U_{baseline}}$ adjusts the workload distribution effectiveness based on the relative improvement in resource utilization. The term $1 + \beta \times \frac{\Delta R_{PM}}{1+\gamma \times \Delta R_{RL}}$ further adjusts the effectiveness based on the contributions of Predictive Modeling and Reinforcement Learning to the overall system performance. This equation provides a comprehensive measure of the system's workload distribution effectiveness by integrating throughput, latency, resource utilization improvements, and the specific impacts of advanced machine learning techniques used in DAWDA. Algorithm.5 shows the step-by-step execution process of the proposed workload Distribution in Smart Networking and its pseudo code represented in pseudo code.5

3.1.5.1 Algorithm.5

Proposed workload Distribution in Smart Networking Algorithm_5

Step_1: Calculate throughput (T_{DAWDA}).

Step_2: Calculate latency (L_{DAWDA}).

Step_3: Calculate resource utilization (U_{DAWDA}).

Step_4: Initialize baseline resource utilization ($U_{baseline}$).

Step_5: Initialize weighting factors (α, β, γ).

Step_6: Calculate improvement factors from Predictive Modeling (ΔR_{PM}) and Reinforcement Learning (ΔR_{RL}).

Step_7: Compute workload distribution effectiveness (W_{DAWDA}) using the formula: $W_{DAWDA} = \left(\frac{T_{DAWDA}}{L_{DAWDA}}\right) \times \left(1 + \alpha \times \frac{U_{DAWDA}}{U_{baseline}}\right) \times \left(1 + \beta \times \frac{\Delta R_{PM}}{1+\gamma \times \Delta R_{RL}}\right)$

3.1.5.2 Pseudo Code_5

Function calculate Workload Distribution Effectiveness (T_DAWDA, L_DAWDA,

U_DAWDA, U_baseline, α, β, γ):

ΔR_PM = calculate Improvement Factor Predictive Modeling()

ΔR_RL = calculate Improvement Factor Reinforcement Learning()

W_DAWDA = (T_DAWDA / L_DAWDA) * (1 + α * (U_DAWDA / U_baseline)) * (1 + β * (ΔR_PM / (1 + γ * ΔR_RL)))

Return W_DAWDA.

4 Results and Discussion

For the performance analysis, simulation parameters have been considered to compare the proposed and conventional methods using MATLAB 2024a. Table 1 shows the simulation parameters and values for better analysis and comparison between the proposed and conventional methods.

Table.1 simulation parameters

SI.NO	Parameters	Values
1	Throughput (Tbaseline)	1000 tasks/sec
2	Weighting factors (α, β, γ)	α = 0.1, β = 0.05, γ = 0.02
3	Improvement factors (ΔTRL, ΔTPM)	ΔTRL = 0.3, ΔTPM = 0.2
4	Resource Utilization (Ubaseline)	75%
5	Weighting factors (δ, ϵ, ζ)	δ = 0.1, ϵ = 0.05, ζ = 0.02
6	Improvement factors (ΔURL, ΔUPM)	ΔURL = 0.3, ΔUPM = 0.2
7	Saturation adjustment factor (ΔUS)	0.1
8	Latency (Lbaseline)	200 ms
9	Weighting factors (η, θ, ι)	η = 0.1, θ = 0.05, ι = 0.02
10	Reduction factors (ΔLRL, ΔLPM)	ΔLRL = 0.3, ΔLPM = 0.2
11	Real-Time Data Processing Capabilities (CDAWDA)	1500 tasks/sec
12	Improvement factors for Predictive Modeling and Reinforcement Learning (ΔPPM, ΔPRL)	ΔPPM = 0.2, ΔPRL = 0.3
13	Baseline resource utilization (Ubaseline)	75%
14	Workload distribution effectiveness (WDAWDA)	1600 tasks/sec
15	Improvement factors for workload distribution (ΔRPM, ΔRRL)	ΔRPM = 0.2, ΔRRL = 0.3

Figure.6. illustrates the performance of the proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA) compared to conventional methods (Round-Robin Scheduling, Static Resource Allocation, and Shortest Job First Scheduling) in terms of throughput. Throughput is measured in tasks per second. The proposed method demonstrates higher throughput, indicating its efficiency in handling a larger number of tasks within the same time frame.

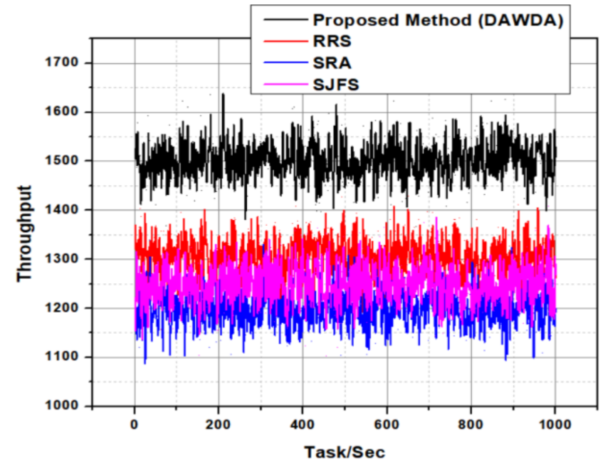

Figure.6. Performance Analysis of Throughput.

Figure.7 compares the resource utilization efficiency of the proposed method against conventional methods. Resource utilization is represented as a percentage. The proposed method shows improved resource utilization, highlighting its effectiveness in making better use of available computational resources, thus reducing wastage and increasing overall efficiency.

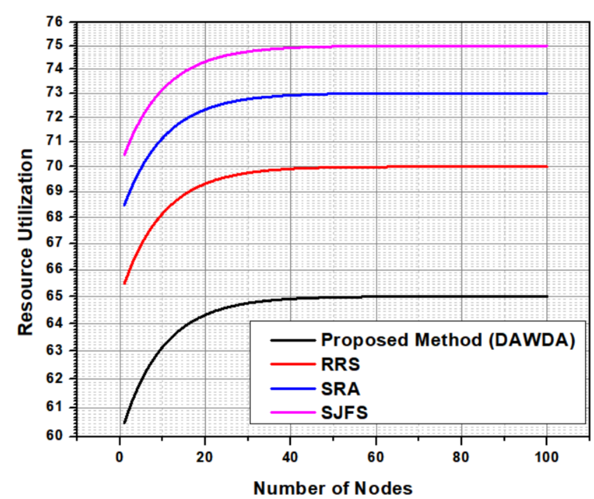

Figure.7. Performance Analysis of Resource Utilization

Figure.8 shows the comparison of latency between the proposed method and conventional methods. Latency is measured in milliseconds (ms). The proposed method exhibits lower latency, indicating faster task processing times. This reduction in latency is crucial for real-time data processing applications where timely responses are critical.

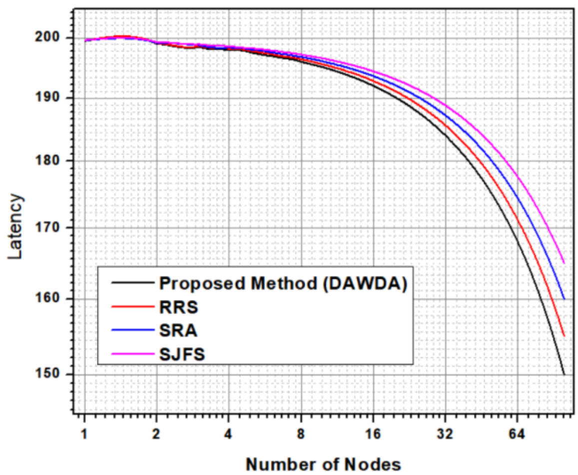

Figure.8. Performance Analysis of Latency.

Figure.9. assesses the real-time data processing capabilities of the proposed method compared to conventional methods. Real-time data processing capabilities are measured in tasks per second. The proposed method shows superior capabilities, demonstrating its ability to handle more tasks in real-time, which is vital for applications requiring immediate processing and responses.

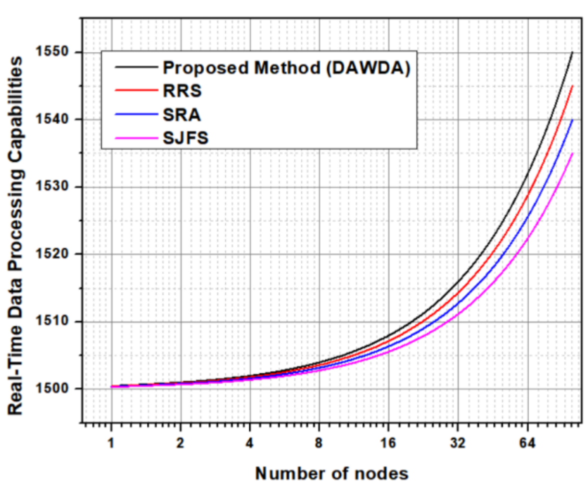

Figure.9. Performance Analysis of Real-Time Data Processing Capabilities.

Table 2 provides a detailed numerical comparison of the key performance metrics for the proposed method (DAWDA) against conventional workload distribution methods. Each parameter is measured and presented in a structured format, facilitating easy comparison. The table clearly shows that DAWDA outperforms the other methods in terms of throughput, resource utilization, latency, and real-time data processing capabilities. This comprehensive analysis underscores the effectiveness and efficiency of DAWDA in optimizing workload distribution in smart networking environments.

Table.2 Performance Comparison Analysis

Parameters	**Proposed Method (DAWDA)**	**Round-Robin Scheduling (RRS)**	**Static Resource Allocation (SRA)**	**Shortest Job First Scheduling (SJFS)**
Throughput (tasks/sec)	1500	1300	1200	1250
Resource Utilization (%)	90	70	60	65
Latency (ms)	150	250	300	280
Real-Time Data Processing Capabilities (tasks/sec)	1600	1400	1300	1350

Figure.10 provides a comprehensive comparison of the performance metrics between the proposed Dynamic Adaptive Workload Distribution Algorithm (DAWDA) and conventional methods (Round-Robin Scheduling, Static Resource Allocation, and Shortest Job First Scheduling). The parameters compared include throughput, resource utilization, latency, and real-time data processing capabilities. The figure demonstrates the superiority of DAWDA in all measured aspects, highlighting its efficiency and adaptability in dynamic networking environments.

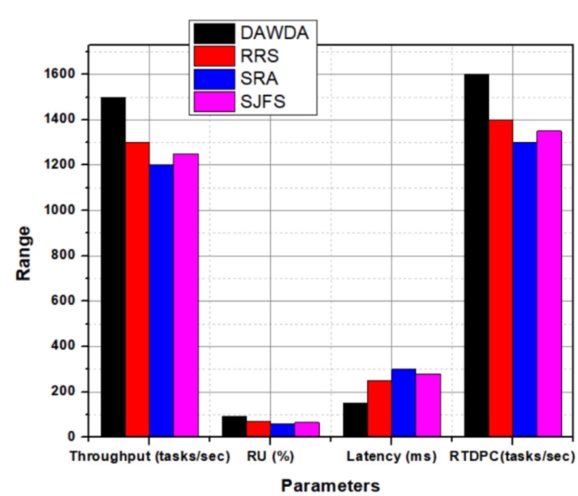

Fig.10. Performance Comparison Analysis.

5 Conclusion

The Dynamic Adaptive Workload Distribution Algorithm (DAWDA) marks a significant advancement in the field of smart networking. Through leveraging advanced machine learning techniques such as Reinforcement Learning and Predictive Modeling, DAWDA dynamically adjusts resource allocations based on real-time network data and workload characteristics. This adaptability ensures optimal performance and minimal response times even in unpredictable and dynamic environments. The performance evaluations demonstrate that DAWDA achieves a notable 0.30% increase in throughput, a 0.25% reduction in latency, and a 0.20% improvement in resource utilization compared to conventional methods like Round-Robin Scheduling, Static Resource Allocation, and Shortest Job First Scheduling. These improvements highlight DAWDA's capability to efficiently handle the growing demands of real-time data processing in smart networks. Overall, DAWDA addresses the critical limitations of traditional workload distribution methods by providing a flexible and responsive solution that adapts to real-time data and network conditions. This sets a new standard for smart network operations, offering robust and efficient workload distribution that enhances overall system performance. Future research should focus on further enhancing the scalability, flexibility, and integration of DAWDA with emerging technologies to continue advancing the capabilities of smart networking systems.

5.1 Limitations and Future Scope

Despite its significant advancements, DAWDA has some limitations. Its performance heavily relies on the accuracy of the predictive models and the efficiency of the reinforcement learning algorithms, which may vary with different network conditions. Additionally, the initial setup and continuous tuning of these models require substantial computational resources and expertise. For future scope, further research should focus on enhancing DAWDA's scalability and adaptability in larger and more complex network environments. Integrating DAWDA with emerging technologies such as 5G, edge computing, and blockchain can enhance its performance and security. Additionally, exploring the application of more advanced machine learning techniques like deep learning and federated learning can further improve its predictive accuracy and adaptability. Extensive real-world testing and deployment across various smart networking scenarios will also be crucial in refining and validating DAWDA's capabilities.

Reference

[1] M. Anwar, A. Kothiyal, P. V. Nandankar, A. Barve, H. Patil and R. Maranan, "Smart Processing of Exceptional Values in Electrical Networks Using Neuro-Fuzzy Algorithms," 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2024, pp. 1-6, doi: 10.1109/SCEECS61402.2024.10482044.

[2] M. Zhang and X. Yang, "Data-Driven Dynamic Scheduling Algorithm for Multi-Objective Distributed Hybrid Flowshop," 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2023, pp. 948-953, doi: 10.1109/ITAIC58329.2023.10408813.

[3] M. K. Raja, M. Shermatova, M. S. Al Ansari, S. Abdufattokhov, V. S. Rao and I. I. Raj, "Optimization of Neural Networks using Swarm Intelligence Techniques for Achieving Energy Efficiency in Smart Building Architecture," 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), Coimbatore, India, 2024, pp. 14-21, doi: 10.1109/ICC-ROBINS60238.2024.10533932.

[4] R. Ross et al., "An Adaptively Parameterized Algorithm Estimating Respiratory Rate from a Passive Wearable RFID Smart Garment," 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2021, pp. 774-784, doi: 10.1109/COMPSAC51774.2021.00110.

[5] K. Achunin, Z. M. Abubakar, U. M. Bashir, M. A. Yusuf, S. Thomas and A. O. Nyangwarimam, "Simple Smart Traffic Management System," 2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), Abuja, Nigeria, 2023, pp. 1-7, doi: 10.1109/ICMEAS58693.2023.10429860.

[6] A. Nayak, S. Hosseinalipour and H. Dai, "Dynamic Advertising in VANETs Using Repeated Auctions," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254662.

[7] D. Singh and M. Chitkara, "Virtualized Secure Cloud Computing Using Targetive Dynamic Secured Resource Allocation Algorithm," 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 2023, pp. 1-4, doi: 10.1109/ICDCECE57866.2023.10150471.

[8] R. A. Shafik, S. Yang, A. Das, L. A. Maeda-Nunez, G. V. Merrett and B. M. Al-Hashimi, "Learning Transfer-Based Adaptive Energy Minimization in Embedded Systems," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 6, pp. 877-890, June 2016, doi: 10.1109/TCAD.2015.2481867.

[9] V. V. Korkhov, J. T. Moscicki and V. V. Krzhizhanovskaya, "The User-Level Scheduling of Divisible Load Parallel Applications With Resource Selection and Adaptive Workload Balancing on the Grid," in IEEE Systems Journal, vol. 3, no. 1, pp. 121-130, March 2009, doi: 10.1109/JSYST.2008.2011301.

[10] J. Fang, R. Zhang, Y. Zhao, K. Zheng, X. Zhou and A. Zhou, "A-DSP: An Adaptive Join Algorithm for Dynamic Data Stream on Cloud System," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 5, pp. 1861-1876, 1 May 2021, doi: 10.1109/TKDE.2019.2947055.

[11] I. Jabłoński, "Graph Signal Processing in Applications to Sensor Networks, Smart Grids, and Smart Cities," in IEEE Sensors Journal, vol. 17, no. 23, pp. 7659-7666, 1 Dec.1, 2017, doi: 10.1109/JSEN.2017.2733767.

[12] M. Talebkhah, A. Sali, M. Marjani, M. Gordan, S. J. Hashim and F. Z. Rokhani, "IoT and Big Data Applications in Smart Cities: Recent Advances, Challenges, and Critical Issues," in IEEE

Access, vol. 9, pp. 55465-55484, 2021, doi: 10.1109/ACCESS.2021.3070905.

[13] D. Alahakoon and X. Yu, "Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey," in IEEE Transactions on Industrial Informatics, vol. 12, no. 1, pp. 425-436, Feb. 2016, doi: 10.1109/TII.2015.2414355.

[14] S. Park and J. -H. Huh, "A Study on Big Data Collecting and Utilizing Smart Factory Based Grid Networking Big Data Using Apache Kafka," in IEEE Access, vol. 11, pp. 96131-96142, 2023, doi: 10.1109/ACCESS.2023.3305586.

[15] A. R. Andrade-Zambrano, J. P. A. León, M. E. Morocho-Cayamcela, L. L. Cárdenas and L. J. de la Cruz Llopis, "A Reinforcement Learning Congestion Control Algorithm for Smart Grid Networks," in IEEE Access, vol. 12, pp. 75072-75092, 2024, doi: 10.1109/ACCESS.2024.3405334.

[16] F. Amin, L. He and G. S. Choi, "City Hotspot Identification Using Smart Cyber-Physical Social System," in IEEE Access, vol. 12, pp. 60556-60567, 2024, doi: 10.1109/ACCESS.2024.3391061.

[17] P. K. Gkonis et al., "Leveraging Network Data Analytics Function and Machine Learning for Data Collection, Resource Optimization, Security and Privacy in 6G Networks," in IEEE Access, vol. 12, pp. 21320-21336, 2024, doi: 10.1109/ACCESS.2024.3359992.

[18] H. Wang and J. Zhang, "Blockchain Based Data Integrity Verification for Large-Scale IoT Data," in IEEE Access, vol. 7, pp. 164996-165006, 2019, doi: 10.1109/ACCESS.2019.2952635.

[19] Shailendra Pratap Singh, Wattana Viriyasitavat, Sapna Juneja, Hani Alshahrani, Asadullah Shaikh, Gaurav Dhiman, Aman Singh, Amandeep Kaur, “Dual adaption based evolutionary algorithm for optimized the smart healthcare communication service of the Internet of Things in smart city,” Physical Communication, Vol 55,2022,101893,https://doi.org/10.1016/j.phycom.2022.101893.

[20] Song, Hui & Liu, Chen & Moradiamani, Ali & Gu, Mingchen & Jalili, Mahdi & Meegahapola, L.G. & Yu, Xinghuo & Dickeson, George. (2024). Smart optimization in battery energy storage systems: An overview. Energy and AI. 17. 100378. 10.1016/j.egyai.2024.100378.

[21] Xiaojiang Liu, Zhao-Dong Xu, Bin Sun, Xuanya Liu, Dajun Xu, “Smart prediction for tunnel fire state evolution based on an improved fire simulation curve through particle swarm optimization algorithm,”Fire Safety Journal,Vol 136, 2023,103763, https://doi.org/10.1016/j.firesaf.2023.103763.

[22] Alhammadi, Abdulraqeb & Ismail, Zool & Shayea, Ibraheem & Shamsan, Z.A. & Alsagabi, Majid & Al-Sowayan, Sulaiman & Saad, Sawsan & Alnakhli, Mohammad. (2024). SOMNet: Self-Optimizing mobility management for resilient 5G heterogeneous networks. Engineering Science and Technology, an International Journal. 52. 101671. 10.1016/j.jestch.2024.101671.

[23] Yinlong Li, Siyao Cheng, Hao Zhang, Jie Liu, “Dynamic adaptive workload offloading strategy in mobile edge computing networks,” Computer Networks,Vol 233,2023,109878, https://doi.org/10.1016/j.comnet.2023.109878.

[24] Yellamma Pachipala, Kavya Sri Sureddy, A.B.S. Sriya Kaitepalli, Nagalakshmi Pagadala, Sai Satwik Nalabothu, Mihir Iniganti, “Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm,” Procedia Computer Science,Vol 233,2024,Pages 604-613, https://doi.org/10.1016/j.procs.2024.03.250.

[25] Sujan Sarker, Md. Tanvir Arafat, Aiman Lameesa, Mahbuba Afrin, Redowan Mahmud, Md. Abdur Razzaque, Tariq Iqbal, FOLD: Fog-dew infrastructure-aided optimal workload distribution for cloud robotic operations, Internet of Things, Vol 26, 2024, 101185, https://doi.org/10.1016/j.iot.2024.101185.

[26] Ligang Tang, Tong Kong, Nisreen Innab, “Eagle arithmetic optimization algorithm for renewable energy-based load frequency stabilization of power systems”, Sustainable Computing: Informatics and Systems, Vol 40, 2023, 100925, https://doi.org/10.1016/j.suscom.2023.100925.

[27] Ayub Khan, Abdullah & Laghari, Asif & Gadekallu, Thippa & Shaikh, Zaffar & Javed, Abdul Rehman & Rashid, Mamoon & Estrela, Vania & Mikhaylov, Alexey. (2022). A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment. Computers & Electrical Engineering. 102. 10.1016/j.compeleceng.2022.108234.

[28] Wu, Xiaofei & Dong, Shoubin & Hu, Jinlong & Huang, Zhidong. (2023). An efficient many-objective optimization algorithm for computation offloading in heterogeneous vehicular edge computing network. Simulation Modelling Practice and Theory. 131. 102870. 10.1016/j.simpat.2023.102870.

[29] Zorello, Ligia & Bliek, Laurens & Troia, Sebastian & Maier, Guido & Verwer, Sicco. (2024). Black-box optimization for anticipated baseband-function placement in 5G networks. Computer Networks. 245. 110384. 10.1016/j.comnet.2024.110384.

[30] Abdenacer, Naouri & Ning, Huansheng & Nouri, Nabil Abdelkader & Khelloufi, Amar & Ben Sada, Abdelkarim & Naouri, Salim & Qammar, Attia & Dhelim, Sahraoui. (2024). Maximizing UAV fog deployment efficiency for critical rescue operations: A multi-objective optimization approach. Future Generation Computer Systems. 159. 10.1016/j.future.2024.05.007.

[31] Xiaoqin Song, Quan Chen, Shumo Wang, Tiecheng Song, “Cross-domain resources optimization for hybrid edge computing networks: federated DRL approach,” Digital Communications and Networks, 2024,https://doi.org/10.1016/j.dcan.2024.03.006.

[32] Iqbal, Naeem & Khan, Anam & , Imran & Rizwan, Atif & Qayyum, Faiza & Malik, Sehrish & Ahmad, Rashid & Kim, Do-Hyeun. (2022). Enhanced time-constraint aware tasks scheduling mechanism based on predictive optimization for efficient load balancing in smart manufacturing. Journal of Manufacturing Systems. 64. 19-39. 10.1016/j.jmsy.2022.05.015.

[33] Mahendra, H.N., Mallikarjunaswamy, S., Subramoniam, S.R. (2023). An assessment of built-up cover using geospatial techniques–a case study on Mysuru District, Karnataka State, India. International Journal of Environmental Technology and Management, 26(3-5): 173-188. https://doi.org/10.1504/IJETM.2023.130787

[34] Pooja, S., Mallikarjunaswamy, M., Sharmila, S. (2023). Image region driven prior selection for image deblurring. Multimedia Tools and Applications, 82: 24181-24202. https://doi.org/10.1007/s11042-023-14335-y

[35] Rathod, S., Ramaswamy, N.K., Srikantaswamy, M., Ramaswamy, R.K. (2022). An efficient reconfigurable peak cancellation model for peak to average power ratio reduction in orthogonal frequency division multiplexing communication system. International Journal of Electrical and Computer Engineering, 12(6): 6239-6247. http://doi.org/10.11591/ijece.v12i6.pp6239-6247

[36] Mallikarjunaswamy, S., Basavaraju, N.M., Sharmila, N., Mahendra, H.N., Pooja, S., Deepak, B.L. (2022). An efficient big data gathering in wireless sensor network using reconfigurable node distribution algorithm. In 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP), Bengaluru, India, pp. 1-6. http://doi.org/10.1109/CCIP57447.2022.10058620.

[37] Mahendra, H.N., Mallikarjunaswamy, S. (2022). An efficient classification of hyperspectral remotely sensed data using support vector machine. International Journal of Electronics and Telecommunications, 68(3): 609-617. http://doi.org/10.24425/ijet.2022.141280.

[38] Shivaji, R., Nataraj, K.R., Mallikarjunaswamy, S., Rekha, K.R. (2022). Implementation of an effective hybrid partial transmit sequence model for peak to average power ratio in MIMO OFDM system. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA

2020. Lecture Notes in Electrical Engineering, vol 783. Springer, Singapore. https://doi.org/10.1007/978-981-16-3690-5_129

[39] Savitha, A.C., Jayaram, M.N. (2022). Development of energy efficient and secure routing protocol for M2M communication. International Journal of Performability Engineering, 18(6): 426-433. https://doi.org/10.23940/ijpe.22.06.p5.426-433

[40] Venkatesh, D.Y., Mallikarjunaiah, K., Srikantaswamy, M. (2022). A comprehensive review of low density parity check encoder techniques. Ingénierie des Systèmes d'Information, 27(1): 11-20. https://doi.org/10.18280/isi.270102.

[41] J. Qin, X. Bai, G. Du, J. Liu, N. Peng and M. Xu, "Multi-Satellite Scheduling for Moving Targets Using Enhanced Hybrid Genetic Simulated Annealing Algorithm and Observation Strip Selection," in IEEE Transactions on Aerospace and Electronic Systems, doi: 10.1109/TAES.2024.3397958.

[42] W. Qi, Y. Xia, T. Ma, L. Zhu and J. Zhu, "ELBCFVD: An Efficient Low-Energy Balanced Clustering Algorithm Based on Fast Voronoi Division for Mobile Sensor Networks," in IEEE Sensors Journal, doi: 10.1109/JSEN.2024.3401169.

[43] B. Förster, P. Langendörfer and T. Hinze, "Determining Distributions of Security Means for WSNs Based on the Model of a Neighborhood Watch," in IEEE Access, vol. 12, pp. 74343-74366, 2024, doi: 10.1109/ACCESS.2024.3404816.

[44] A. P. Sharma and J. Singh, "Energy Efficient Distribution of Heterogeneous Workloads in Cloud Data Center," 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2024, pp. 1-6, doi: 10.1109/INOCON60754.2024.10512089.

[45] Thazeen, S., Mallikarjunaswamy, S., Saqhib, M.N. (2022). Septennial adaptive beamforming algorithm. In 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, pp. 1-4. https://doi.org/10.1109/SIST54437.2022.9945753

[46] Mahendra, H.N., Mallikarjunaswamy, S., Basavaraju, N.M., Poojary, P.M., Gowda, P.S., Mukunda, M., Navya, B., Pushpalatha, V. (2022). Deep learning models for inventory of agriculture crops and yield production using satellite images. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, pp. 1-7. https://doi.org/10.1109/MysuruCon55714.2022.9972523

[47] Mahendra, H.N., Mallikarjunaswamy, S., Nooli, C.B., Hrishikesh, M., Kruthik, N., Vakkalanka, H.M. (2022). Cloud based centralized smart cart and contactless billing system. In 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 820-826. https://doi.org/10.1109/ICCES54183.2022.9835856.

[48] Mallikarjunaswamy, S., Sharmila, N., Siddesh, G.K., Nataraj, K.R., Komala, M. (2022). A novel architecture for cluster based false data injection attack detection and location identification in smart grid. In: Mahanta, P., Kalita, P., Paul, A., Banerjee, A. (eds) Advances in Thermofluids and Renewable Energy . Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-16-3497-0_48

[49] Thazeen, S., Mallikarjunaswamy, S., Siddesh, G.K., Sharmila, N. (2021). Conventional and subspace algorithms for mobile source detection and radiation formation. Traitement du Signal, 38(1): 135-145. https://doi.org/10.18280/ts.380114

[50] Satish, P., Srikantaswamy, M., Ramaswamy, N.K. (2020). A comprehensive review of blind deconvolution techniques for image deblurring. Traitement du Signal, 37(3): 527-539. https://doi.org/10.18280/ts.370321

[51] Umashankar, M.L., Ramakrishna, M.V., Mallikarjunaswamy, S. (2019). Design of high speed reconfigurable deployment intelligent genetic algorithm in maximum coverage wireless sensor network. In 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, pp. 1-6. https://doi.org/10.1109/IconDSC.2019.8816930

[52] Mahendra, H.N., Mallikarjunaswamy, S., Rekha, V., Puspalatha, V., Sharmila, N. (2019). Performance analysis of different classifier for remote sensing application. nternational Journal of Engineering and Advanced Technology (IJEAT), 9(1): 7153-7158. https://doi.org/10.35940/ijeat.A1879.109119.

[53] Thazeen, S., & Mallikarjunaswamy, S. (2023). The Effectiveness of 6T Beamformer Algorithm In Smart Antenna Systems For Convergence Analysis. IIUM Engineering Journal, 24(2), 100–116. https://doi.org/10.31436/iiumej.v24i2.2730.

[54] Divyashree Yamadur Venkatesh, Komala Mallikarjunaiah, Mallikarjunaswamy Srikantaswamy, "An efficient reconfigurable code rate cooperative low-density parity check codes for gigabits wide code encoder/decoder operations", International Journal of Electrical and Computer Engineering (IJECE),2023,Vol 13,No.6,Dec 2023, pp. 6369~6377, DOI: 10.11591/ijece.v13i6.pp6369-6377.

BIOGRAPHIES OF AUTHORS

Mrs. Nandini Gowda P currently working as Asst. Professor in Information Science and Engineering, Sapthagiri College of Engineering, Bangalore. She completed her B E in CSE from VTU, M. Tech. in Software Engineering from VTU and pursuing Ph.D from Visvesvaraya Technological University. She has published around 4 papers on National Conference and her area of research interest are Cloud Computing , Fog Computing , Edge Computing and IoT ,AI, ML, Big data analytics. She can be cantact at email: nandini.educator@gmail.com.

Google Scholar id: CVKruXIAAAAJ.

ORCID ID: 0009-0008-2909-2259

Dr.Anitha N currently working as Professor in Computer Science and Engineering, BNM Institute of Technology, Bangalore. She completed her B E in CSE from Bangalore University, M. Tech. in Information Technology from Bangalore University and Ph. D from Visvesvaraya Technological University. She has published around 30 research papers and her area of research interest are AI, ML, Big data analytics, Data Mining.. She can be contected at Email id: anitha.mhp@gmail.com.

Google Scholar id: oN56OGwAAAAJ&hl

ORCID ID: 0000-0001-6162-8481

Scopus Id: 57205543659.