IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 May 2024, accepted 9 June 2024, date of publication 17 June 2024, date of current version 24 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3415418

RESEARCH ARTICLE

Opti-FOG: Optimizing Resource Allocation With Anti-Inspired FOG Node Selection

P. KARTHIKEYAN AND K. BRINDHA
School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632014, India
Corresponding author: K. Brindha (brindha.k@vit.ac.in)

ABSTRACT Fog computing is a decentralized environment capable of data collection, storage, and analysis, in addition to providing cloud computing services to edge devices. In a fog computing environment, several noteworthy challenges have been faced with data storage technologies. Based on the literature, to enhance fog node data storage when traffic in a specific location exceeds capacity, the capacity of the fog node is enhanced by adding another fog node to the existing node to accommodate the demand. However, selecting a neighbor node in networking brings several challenges, such as node capacity, security, and availability. This article proposes a framework Optimizing Resource Allocation with Anti-Inspired Fog Node Selection (Opti-FOG), for resolving the issues of node capacity and availability for selecting the optimal node to allow a storage system to grow or shrink without impairing the performance of surrounding nodes. The proposed framework, Opti-FOG, addresses the challenges of fog node selection in decentralized environments, enhancing both latency and availability in fog computing systems. By leveraging the Ant Colony Optimization (ACO) algorithm, Opti-FOG optimizes resource allocation while considering proximity and weighted node values, thereby improving the efficiency and responsiveness of fog node selection processes. The simulation results underscore the significance of Opti-FOG in accurately assessing fog node selection performance, demonstrating its potential for practical deployment scenarios where latency and availability are crucial factors.

INDEX TERMS Ant colony optimization, fog computing, latency, security, scalability.

I. INTRODUCTION

In the contemporary world, sensor nodes occupy a wide range of sectors; such as smart monitoring of the environment, smart homes, smart farming, and smart healthcare. A sensor node is a tiny independent device that can be used to gather information from the environment. In wireless sensor networks (WSNs), which are sizable networks of sensor nodes set up to monitor a specific area, they are frequently employed. A recent statistical study predicts that the global market for sensor nodes will grow from $10.5 billion in 2020 to $22.2 billion in 2025. Wireless sensor nodes become vital for the Internet of Things (IoT) because they can collect data from their surroundings and communicate it to a higher layer over the Internet. The number of linked devices is projected to reach 29 billion by the year 2023, indicating that the IoT is expanding at a rapid rate.

The IoT must create real-time methods to lower costs and better use resources while working within the constraints of energy management systems [1]. IoT introduced a Markov model and optimization algorithm modeled after nature to manage resource restrictions, ensure service quality, and avoid energy loss [2]. The sensors, gateways, and power meters that make up an IoT infrastructure can collect and analyze sensor data in real-time, allowing for the scheduling of a home's electrical appliances based on the user's presence and preferences [3]. The IoT's success depends on the creation of an effective communication architecture that enables data transmission across many kinds of devices. IoT-based WSNs adopt a multi-criteria approach for node selection, taking energy efficiency and anticipated network longevity into account. Each parameter in the node selection

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

process is computed using the widely utilized multi-criteria Analytical Network Process (ANP) [4]. The network must continue to operate regularly despite a substantial increase in the number of end users; scalability becomes one of the most significant challenges in an IoT network. Two alternative approaches, both based on fuzzy logic, are offered for selecting nodes in the IoT [5].

Sensors are the backbone of IoT devices and have the ability to gather a humongous trove of data from their surroundings. Fog computing is a decentralized system built on top of existing sensor networks, which can quickly and efficiently gather, transmit, and analyze large amounts of data [6]. Offloading and migrating tasks are essential in a fog-computing environment. However, these fog devices present unique challenges because of their limited resources. It recommends L3Fog, a system for low-latency fog selection and offloading based on learning and location awareness, for the Internet of Things [7]. Fog computing not only executes applications near IoT devices, but it also provides short-term storage for the data generated by those devices. As such, there is a reduction in the frequency of data transmissions to the cloud. Problems that arise in fog computing-based IoT architecture include determining the best route between two points and the best choice of nodes. To address the problem efficiently, the suggested greedy-heuristic approach considers delay, energy consumption, multi-hop paths, and dynamic networks as parameters [8]. Due to the variety of IoT requirements and the availability of many different types of fog nodes, making a decision on which one to use for a given activity presents several challenges. There has been no research done on the best way to choose fog nodes while also satisfying user preferences, user requirements, and optimal resource utilization. For workload offloading from IoT devices, it proposed using a best fit-power weighted difference (BF-PWD) method to rank and choose fog nodes [9].

The ACO algorithm is a strategy used in a variety of fields, including combinatorial optimization, communication networks, and robotics, that is inspired by how natural ant colonies choose the most efficient route between food sources and their nests [10]. Fog computing's low latency is better than the cloud's high latency for applications that can't handle delays. For tasks to respond faster, fog nodes need to spread IoT apps out in the right way. It was suggested that the ant colony optimization technique would offer a low-cost, fast-responding, efficient load-balancing solution for fog computing [11]. The nodes in the fog layer will cooperate to reduce response time if the task schedule is balanced, otherwise they will delay by delegating certain responsibilities to another fog node close by. An ACO algorithm, an optimization algorithm, can be used to decide which fog node to offload work from to achieve load balancing across the network [12].

In this research article, the proposed Optimizing Resource Allocation with Anti-Inspired FOG Node Selection (Opti-FOG) algorithm for selecting the best optimal node in the fog computing environment achieves the following contributions

- Introduce a scalable framework characterized by a three-layer design that asserts the ability to systematically identify the optimal fog node from neighboring nodes.
- Employ Quality of Service (QoS) metrics as a basis for calculation, enabling the precise evaluation of energy consumption, CPU utilization, and memory storage for each node.
- Compile a comprehensive dataset by aggregating the total energy consumption, CPU utilization, and memory storage metrics from all adjacent nodes.
- Establish weighted values for energy consumption, CPU utilization, and memory storage capacity, underscoring their inherent significance in the fog node selection process.
- Evaluate and scrutinize sorting algorithms, with a specific focus on Quick Sort, to confidently declare the most effective approach for arranging fog nodes based on their weighted scores.
- Demonstrate the application of the Ant Colony Optimization technique, to determine the shortest distance and combining with a high score substantiating its effectiveness in selecting the optimal fog node.

The literature review and relevant works, discussed in Section II, provide insights into scalability, energy consumption, and neighbor node selection using the Ant Colony Optimization algorithm. Section III, introduces the proposed approach, outlining its methodology and framework for addressing the critical aspects of fog computing. Section IV, presents the results of experimental evaluations conducted to validate the effectiveness and efficiency of the proposed work. Section V discusses future research opportunities for advancing the proposed work. Finally, Section VI concludes the paper.

II. LITERATURE AND RELEVANT WORKS

This section explores critical frontiers in the fog and another dynamic field of recent literature that includes energy consumption approaches, bio-inspired algorithms for shortest distance finding, and scalability in node selection. The literature delves into the intricacies of choosing scalable nodes, examines energy consumption techniques, and investigates the use of bio-inspired algorithms for shortest distance finding. Together, the section converges to form a rich tapestry of exploration and discovery, charting the course for future advancements at the intersection of literature and technology.

A. SCALABILITY OF NODE SELECTION

Fog node selection employs a decentralized technique in which a local fog node is selected to serve as an orchestrator. This technique takes into account the current fog status to assess whether it is necessary to place the needed fog

nodes. The fog orchestration node selection methodology allows even the most basic IoT devices to choose a fog orchestration node [13]. Due to its scalability and quick, cost-free transactions, IOTA's Distributed Ledger for IoT has gained attention. However, there is a chance of failure because it relies on a single coordinator. In response to this, the study introduces the Multiple Coordinator Selection (MCS) algorithm. It increases system robustness by examining parameters like trust level and node activity [14]. Content delivery networks (CDNs) have difficulties when it comes to efficiently distributing content, particularly in locations with high populations. Latency in the network and bandwidth limitations plague traditional cloud-based designs. It proposed the Joint Optimization Algorithm for Fog-Node location and Content Distribution (JFnP-CDA) to solved the problem by optimizing the cost of the CDN overall as well as the location of fog nodes and content [15]. Deploying intelligent fog nodes is necessary to manage data from edge to cloud efficiently and maintain network operability. Cost and computing complexity make it difficult, but balancing coverage and connectivity is essential. Its suggested fog dispersal approach has a low complexity and quick convergence by utilizing the Marine Predator approach (MPA) [16]. Wearable sensor-based IoT-driven medical solutions are revolutionizing healthcare. It is critical to use SDN-oriented fog frameworks, especially for the early identification of communicable diseases. FogCom uses the SPNE method from game theory to resolve fog node selection problems and a three-layered CNN approach for disease diagnosis [17]. In order to effectively allocate resources and delegate tasks, fog computing systems must carefully select their fog nodes. The Fog Node Selection Engine (FNSE) is a tool that we use to help consumers make decisions by predicting fog node trust values using AI-driven models. There are three models developed and executed independently: fuzzy logic (FL), logistic regression (LR), and deep neural network (DNN) [18].

The fog computing technique for the Industrial Internet of Things (IIoT) has reduced communication delays in complex activities. However it's not possible to store all resources on a single fog node, so selecting the fog node is the biggest challenge in a fog computing environment. It recommended a multi-level hierarchical fog node deployment model for the industrial environment, where the selection relies on energy, storage, location, network properties, and available computing resources [19]. In an IoT context supported by fog, Horizontal Distributed Fog (HDFog) presented a two-step distributed horizontal architecture with a greedy selection criterion for computation offloading, whereby sensor nodes in user devices transfer their responsibilities to a nearby fog node. When executing demanding real-time applications, HDFog minimizes energy consumption and delays [20]. Fog-to-fog interactions in decentralized computing infrastructures allow for optimal resource usage, low-latency data processing, and collaborative decision-making. By leveraging the processing power of neighboring nodes, fog computing systems can distribute tasks effectively, leading to improved performance. The geographical dispersion of fog nodes enables smooth data sharing and collaboration, while the dynamic nature of the environment supports adaptive resource allocation. Collaborative data processing and backup capabilities among fog nodes further enhance system reliability and scalability, making fog-to-fog interactions essential for optimizing the overall efficiency of fog computing environments [21].

The hybrid scheduling-clustering approach tackles the challenge of scheduling applications with time constraints in Cloud-Fog scenarios, which is exacerbated by the participation of IoT nodes and unstable surroundings. To establish stable clusters of IoT nodes, the proposed method leverages a fuzzy inference-based lightweight decentralized clustering algorithm. By utilizing cloud resources and mobile and inactive Fog node clusters, it provides effective scheduling while promoting scalability and flexibility [22]. To optimize fog computing system deployment, the study offers a unique fog device deployment algorithm that employs a recently developed metaheuristic technique known as search economics. The algorithm's goals include improved performance measures such as reduced latency and resource utilization [23]. Fog nodes are essential in 6G networks, as they drive edge intelligence for better resource allocation, routing, and security. Local data processing speeds up decision-making and reduces latency for important applications. Fog nodes ensure effective resource use and increased network performance by dynamically allocating resources and routing traffic. They also improve security and enable low-latency service delivery, resulting in increased network efficiency and user experience. Furthermore, the discussion emphasizes the challenges faced by 6G networks, the importance of machine learning for intelligent network management, and the potential benefits of 6G's complex structure, highlighting AI's critical role in improving network performance, reliability, and latency [24].

One of the hardest things about fog networking is selecting the best node for performance and utilization of resources. A new distributed agent-based orchestrator model takes into account the CPU, memory, battery life, and security levels of fog nodes, this model can provide flexible service in a dynamic fog computing environment [25]. Scalability is capable of expanding the capacity of fog network environments; fog nodes select the nearest neighbor node to position to overcome traffic issues. There are two methods available: fog discovery and fog selection. If research on fog node selection is extremely limited, various methodologies are adopted, such as machine learning, deep learning, hybrid algorithms and objective approaches, and multi-criteria decision-making [26]. An effective method of selection is created for an ideal fog node selection problem, which is formulated as a multi-objective optimization problem, using enhanced particle swarm optimization (PSO) and a modified

firefly algorithm (FA). This study takes into account the two essential objectives to be improved upon in the work of trust and sojourn rate [27].

B. RECKONING OF ENERGY CONSUMPTION WITH DIFFERENT METHODOLOGY

Numerous sorting algorithms have been studied to address energy efficiency while taking time and space complexity into consideration. It investigates the amount of energy consumed by various sorting algorithms, with a particular emphasis on the relationship between energy consumption and computational performance. It provides the fundamental information needed to select an appropriate sorting algorithm implementation to reduce overall energy consumption [28]. Each computing node has a finite amount of energy, so selecting the node with the least energy consumption is one of the primary challenges in a computing network. The author described a technique for measuring energy based on the energy consumption of the node [29]. The network's consumption of energy and balance both show continued indications of being out of balance and excessive energy usage. For this purpose, a novel node selection technique is developed for optimizing transmission energy balance based on selection probability and energy balancing of node routes. It takes into consideration the optimal path node choice when determining the amount of energy possessed by each node in the network [30]. IoT technology generates enormous amounts of data, moving it to cloud data centers is expensive in terms of latency and security. Although fog node capabilities are limited by static configurations, fog computing provides effective local processing. In order to anticipate requests, this study uses the Decision Tree (DT) model to predict fog service location and dynamic reconfiguration. Analysis shows reduced energy use, enhanced throughput, and dynamic fog node switching [31]. Fog computing improves computational resources for IoT users while increasing energy consumption and reducing system reliability due to node mobility. To address this, it created the Minimal Schedule Time with Energy Constraint (MSTEC) method for low-delay fog computing operations, which optimizes energy utilization [32]. Fog computing brings services closer to end devices, minimizing delay and network burden when compared to cloud architecture. It offers a heuristic-based approach to efficiently use fog resources based on edge node cluster load, resulting in lower network utilization and delays [33]. It analyzes the difficulties of federated edge learning (FEEL) and over-the-air computation (AirComp) in wireless networks, and presents a solution: a dynamic device scheduling algorithm. This approach picks edge devices based on data significance, channel condition, and energy consumption, hence enhancing model training efficiency and convergence in federated learning [34]. It emphasizes how important dependable fog nodes are to optimizing energy consumption, enhancing network efficiency, and guarding against insider threats [35].

C. SHORTEST DISTANCE FINDING BASED ON BIO-INSPIRED ALGORITHM

Optimization algorithms for bio-inspired computing are a relatively new technology that draws inspiration and ideas from the biological evolution of nature to create robust and innovative competing tactics. Numerous disciplines, including biology, computer science, and mathematics, can make use of it. The following are examples of contemporary bio-inspired optimization algorithms: artificial algae algorithm (AAA), elephant search algorithm (ESA), moth flame optimization (MFO), chicken swarm optimization algorithm (CSOA), fish swarm algorithm (FSA), ant colony optimization (ACO), genetic bee colony (GBC) algorithm, whale optimization algorithm (WOA), and particle swarm optimization (PSO) [36]. Selection of fog nodes optimally is necessary for fog computing, which is essential for applications requiring low latency. In order to provide an effective strategy utilizing modified Firefly Algorithm (FA) and enhanced particle swarm optimization (PSO), this paper formulates it as a multi-objective optimization problem. With BWM, objectives are combined and weights are established based on trust, rate of sojourn, node capacity, and energy usage [27]. Fog node task selection is heavily influenced by the cost of offloading jobs; tasks that are less expensive are chosen over those that are more expensive. It suggested a Reference Incentive Mechanism (RIM) that adds a frame of reference to fog computing offloading in order to encourage fog nodes to participate in expensive jobs without raising platform prices. High-cost jobs are made more appealing to fog nodes and are therefore more likely to be selected by RIM through the use of reference objects and the establishment of a suitable reference task [37].

The ACO Algorithm is the best optimization algorithm out of all algorithms for finding the best optimal node. Finding the node with the shortest distance to its neighbors is one of the most challenging aspects of network architecture, Finding the shortest neighbor node saves time and cost. "Ant algorithms" refers to the study of algorithms that are motivated by the activities of ant colonies. The ACO algorithm models how ant colonies pick the fastest route to and from their nests [10]. The distribution issue in an industrial environment was successfully resolved with the help of the greedy method. However, one of its limitations is that it does not take into account the potential future effects of its decisions. This is the primary motivation for introducing the idea of ant colony optimization techniques. The organized behavior of ants served as an inspiration for the development of the ant colony optimization algorithm, which also determines the shortest route in distribution environments [38].

In Table 1 shows recently published articles in the research domain of fog computing. A review of the relevant literature reveals that various methodologies and approaches exist for estimating energy consumption using various metrics; however, all research articles focus solely on reducing energy efficiency rather than selecting the best optimal node for

TABLE 1. Survey of the existing literature.

Author	Category	Conspectus	Limitations
Alam et al.[7]	Fog Node Selection	Internet of Things position-aware low latency fog selection and offloading system that uses machine learning-based methods to forecast the location of the mobile nodes	Scalability and accuracy dependency challenges in fog node selection exist
Hussein et al.[11]	Ant Colony Optimization and Particle swarm optimization	Effectively distribute IoT tasks across fog nodes in consideration of communication cost and response time.	Lack of detailed coverage on fog node resource limitations
Qayyum et al.[19]	Fog Node Selection	For even low-powered IoT devices to be able to choose a FON, using a lightweight FON Selection model (FONS).	Limited scalability testing and energy consumption analysis
Deb et al.[20]	Fog Node Selection	The deployment model of fog nodes in an industrial environment is multi-level and hierarchical	Lack of advanced optimization approaches in fog node selection
Liutkevicius et al.[25]	Fog Node Discovery	A dynamic fog computing environment that is compatible with a distributed agent-based orchestrator paradigm that can provide flexible service provisioning	Lack of Scalability concerns and complexity impact on latency in fog nodes
Ogundoyin et al.[27]	Fog Node Selection	The PSO-FA method is used to solve the fog node selection problem effectively and reliably.	Scalability challenges exist in fog node selection using the proposed method
Khan et al. [30]	Fog Node Selection	Energy-adjustment factor to optimize node-selection probability for dynamic network node selection.	Inadequate focus on energy optimization and dynamic techniques for efficiency
Cheng et al. [39]	General Node selection	To effectively extend the network lifetime, a minimum number of nodes must be selected to offer necessary data services within the error threshold.	Challenges exist in adaptive node selection algorithms and security integration
Salimian et al. [40]	Particle swarm optimization	A fog-cloud control middleware-based conceptual computing framework is suggested to best position IoT services and maximize fog resource utilization and QoS.	Inadequate examination of scalability implications and evaluation measures
Alotaibi et al. [12]	Ant Colony Optimization	Fog master schedules the task to a fog layer node using the Ant Colony Optimization algorithm.	Lack of security and comprehensive performance metric analysis
Kishor et al. [41]	Smart Ant Colony Optimization	Reduce the amount of time spent offloading tasks in an IoT-Fog environment by taking into account the latency of computation and communication.	Limited consideration for task offloading delays and power consumption
Yin et al.[42]	Hybrid Monarch butterfly optimization and Ant Colony optimization algorithm (HMA)	Provide a cloud-fog computing architecture for intelligent production lines and a priority-based task rescheduling method to ensure that time-sensitive positions obtain priority services	Scalability issues and evaluation gaps in priority-based rescheduling technique

storage to prevent network traffic and save resources, time, and cost. The objective of this paper is to reduce traffic on independent fog nodes using a fog computing network by selecting the optimal fog node for data storage allocation.

III. SYSTEM DESIGN AND PROPOSED FRAMEWORK

In fog networking, each fog node can store data, perform computation, and communicate with other nodes. A variety of aspects in a fog computing environment are required for scalability at an individual node, some of the most typical indicators of scalability requirements are as follows: 1. Resource Utilization 2. Latency 3. Throughput and 4. User demand. As a consequence of the fog node's implementation on scalability, additional fog nodes can be deployed to handle increased traffic loads without impacting the performance of the underlying network. In response to the escalating challenges posed by burgeoning data traffic on independent fog nodes within fog computing networks, this paper presents a strategic solution Optimizing Resource Allocation with Anti-Inspired Fog Node Selection (Opti-FOG), aimed at mitigating congestion and enhancing overall system efficiency. By introducing a meticulous approach to the selection of optimal fog nodes for data storage allocation, the proposed methodology seeks to alleviate the burden on individual nodes, thereby optimizing the network's performance.

The System design of the proposed work Opti-FOG in a fog environment is shown in Figure 1. This proposed algorithm unfolds through a systematic process, commencing with the data generation by IoT devices. The data generated are been transferred to the fog node through the edge devices. In fog computing environments, scalability and latency management are two intertwined challenges crucial for optimizing data transfer mechanisms. As the number of edge

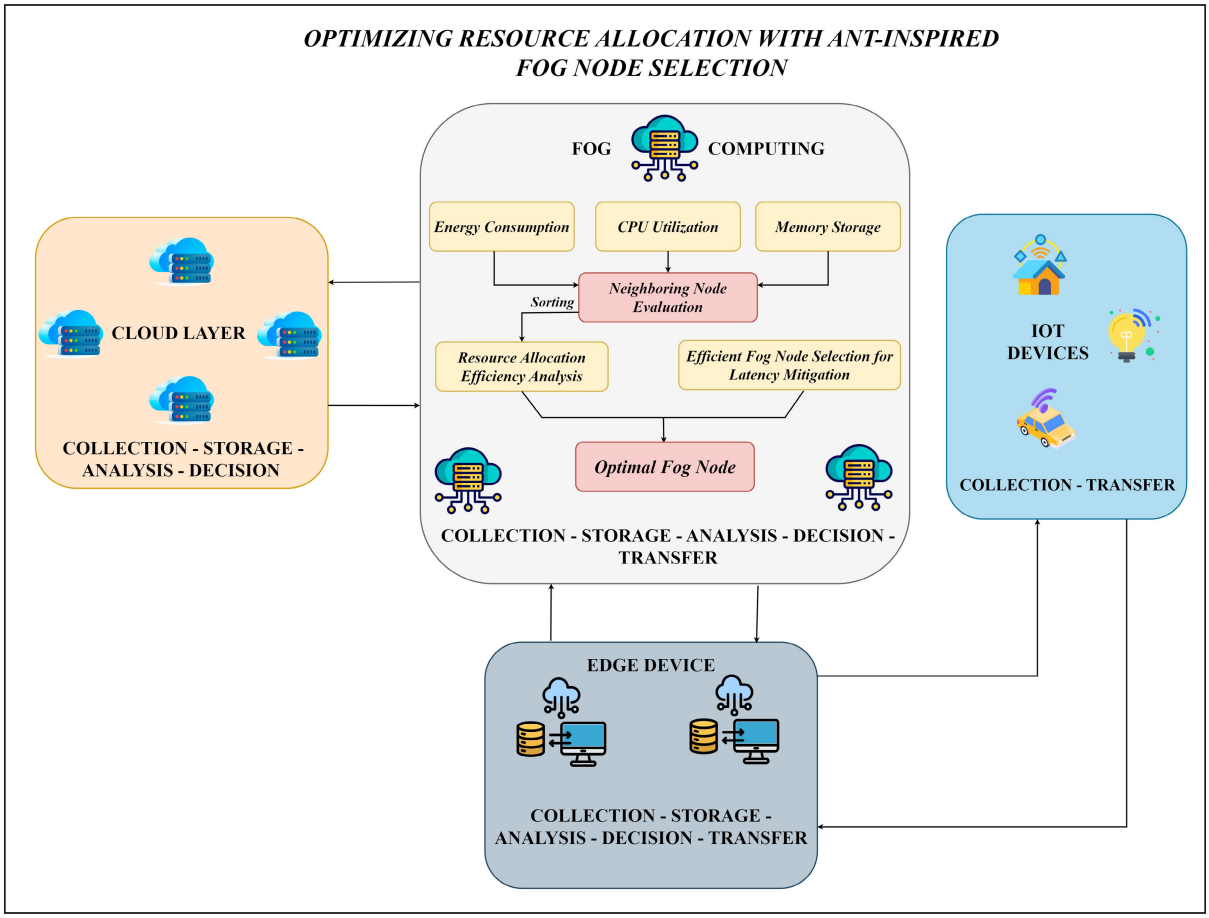


FIGURE 1. System Design of Opti-FOG: optimization resource allocation with anti-inspired fog node selection.

devices proliferates within fog networks, accommodating the escalating volume of generated data demands scalable solutions. However, achieving scalability must also consider the imperative to mitigate latency, a primary goal of fog computing. By processing data closer to the edge, fog computing minimizes latency and enhances real-time responsiveness. Nonetheless, the act of transferring data to remote fog nodes can introduce additional latency, counteracting the benefits of edge processing. Balancing scalability and latency entails deploying distributed architectures and efficient routing protocols that dynamically allocate resources and prioritize data traffic based on proximity and urgency. Leveraging edge caching and pre-processing techniques can further mitigate latency by reducing the need for continuous data transmission to distant fog nodes. Additionally, advancements in edge computing hardware and network infrastructure, such as edge computing gateways and optimized communication protocols, play pivotal roles in addressing scalability and latency concerns simultaneously. Ultimately, integrating scalable data transfer mechanisms with latency-aware processing strategies forms the cornerstone of efficient and responsive fog computing ecosystems.

To achieve this each fog node delves into neighboring node estimation, a pivotal step that provides a comprehensive understanding of the dynamic network environment. Exactly gather the necessary environmental properties as the foundation for the subsequent analyses on a fog node by another fog node. This leads to efficient resource allocation efficiency analysis, after meticulously examining the intricate details of energy consumption, CPU utilization, and memory storage across all nodes. Armed with these insights, adapt ant colony optimization techniques, strategically exploring the shortest paths within the dataset to enhance decision-making. The integration and analysis allow the harmonious amalgamation of the diverse datasets and subject them to a comprehensive evaluation, ensuring a holistic understanding of the network's intricacies. This approach reveals the optimal node selection as a solution for data storage in the fog environment to ensure that scalability requirements are met while minimizing latency in data transmission. The output represents the synthesis of the efforts, blending input data, accurate estimation, resource efficiency analysis, optimization techniques, and a thorough validation process.

Through this comprehensive journey, the research contributes not only to the selection of the optimal node but also to the broader discourse on enhancing the efficiency and intelligence of fog computing networks. To fortify the reliability

and applicability of the approach, a rigorous validation and evaluation process is undertaken. This meticulous scrutiny aims to validate the robustness of the methodology and verify its efficacy in real-world scenarios. Figure 2 shows the Opti-FOG framework designed for selecting the optimized fog node for storage.

The Opti-FOG system introduces significant algorithmic contributions by tailoring the Ant Colony Optimization (ACO) algorithm specifically for the challenges inherent in fog computing environments. Traditional ACO algorithms, while effective for problems like the Traveling Salesman Problem, require modifications to address the unique demands of fog node selection, which involve dynamic resource allocation, proximity considerations, and real-time responsiveness. In the Opti-FOG framework, the ACO algorithm is adapted to consider multiple criteria crucial for fog computing. These criteria include node capacity, security parameters, and network latency, which are not typically part of conventional ACO implementations. By incorporating these factors, the modified ACO algorithm can dynamically adjust to varying network conditions and workloads, ensuring optimal performance and reliability.

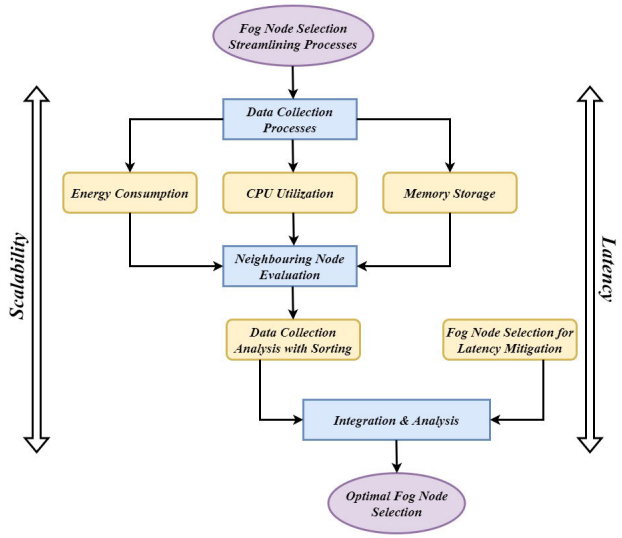


FIGURE 2. Framework of Opti-FOG: optimization resource allocation with anti-inspired fog node selection.

One key modification is in the pheromone update rules. In traditional ACO, pheromones primarily guide ants towards the shortest path. However, in the context of fog computing, the shortest path may not always be the most suitable due to potential security risks or capacity limitations. Therefore, Opti-FOG enhances the pheromone update mechanism to prioritize not only the shortest paths but also those that offer the most reliable and secure nodes. This ensures that the selected fog nodes are capable of handling data traffic efficiently while maintaining robust security standards. Furthermore, Opti-FOG introduces real-time monitoring and dynamic reconfiguration capabilities within the ACO framework. This allows Opti-FOG to continuously assess the state of the network and make necessary adjustments to node assignments, thereby enhancing the system's adaptability and resilience against node failures and network disruptions. These algorithmic innovations are crucial for enabling the efficient and effective deployment of fog nodes, ensuring scalability, resilience, and security. By detailing these specific modifications, the technical depth and novelty of the Opti-FOG system are demonstrated, highlighting its suitability and superiority for real-world applications in fog computing environments.

A. ESTIMATING NEIGHBORING FOG NODE PERFORMANCE

Estimating neighboring fog node performance is vital for optimized node selection in fog computing. It ensures efficient resource allocation and minimizes latency by choosing nodes with suitable capabilities and proximity. Load balancing across nodes prevents bottlenecks and supports scalability as networks expand. Additionally, selecting energy-efficient nodes enhances system sustainability and fault tolerance, ensuring reliable operations. To achieve better resource utilization, responsiveness, and reliability in fog computing environments performance estimation of the nodes is done based on energy consumption, CPU utilization, and memory storage.

1) FOG NODES ENERGY CONSUMPTION AND COMPUTATIONAL MODELING

The term ''energy consumption'' refers to the amount of energy required for data communication and computation at the network nodes. In a fog-networking environment, several metrics are utilized to determine the amount of energy consumed and resources available in the fog node, such as packet sending, receiving, idle, and sleep.

Figure 3 shows how to determine each fog node's energy consumption and sort the dataset using a sorting algorithm.

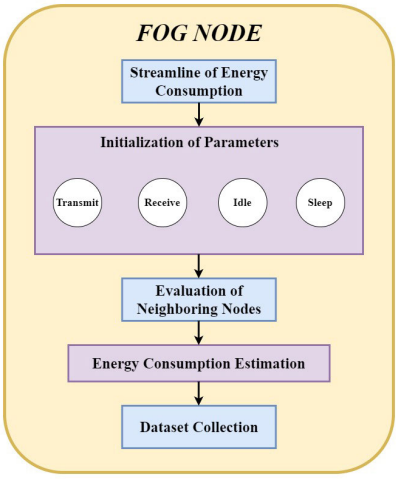


FIGURE 3. Estimating energy consumption.

To proceed with the mathematical modeling approaches for determining how much energy is consumed by each fog node, Where Transmit (Tt), receive (Tr), idle (Ti), and sleep (Ts) are the amounts of time spent in the fog node, and Pt, Pr, Pi, and Ps are the amounts of power used in each state of the fog node. The amount of time spent and power used by the fog node is taken into consideration to calculate TEC, which represents the total resource of energy consumption.

$$TEC = \sum_{i=1}^{4} (Time)_i * (Power\_Rate)_i \quad (1)$$

where:

- Each component is represented by i: idle, sleep, transmit, and receive
- $Time_i$ indicates the total time spent in each state
- $Power\_Rate_i$ denotes the corresponding power rate for each state

The mathematical notations that are used to find each node's resources are shown in Table 2.

TABLE 2. Mathematical notations.

S. No.	Notations	Descriptions
1	TEC	Total Energy Consumption
2	TAT	Total Active Time
3	MSC	Memory Storage Capacity
4	WES	Weighted Energy Score
5	WCPUS	Weighted CPU Score
6	WMS	Weighted Memory Score
7	TWS	Total Weight Score

The process for determining each Fog Node's energy usage based on the packet sending, receiving, idle, and sleep states is shown in Algorithm 1. Algorithm 2 aggregates energy consumption values for transmission, reception, idle, and sleep states to calculate the overall energy consumption of a fog node. It also monitors transitions between these states and schedules tasks using a timer.

2) FOG NODES CPU UTILIZATION

The CPU load of the system is continuously monitored by Algorithm 3, which then converts it into a percentage to calculate the CPU usage of a fog node. Through this procedure, the Fog Node can accurately measure how much of its computational resources are being used, which is important for optimizing task allocation and system performance as a whole.

$$TAT = Simu\_Dura + \sum_{i=1}^{4} (Time)_i \quad (2)$$

$$CPUUtilization = (TAT \ / \ Simu\_Dura) * 100 \quad (3)$$

where:

- Each component is represented by i: idle, sleep, transmit, and receive.
- $Time_i$ indicates the total time spent in each state.
- TotalActiveTime (TAT) indicates the duration of CPU activity during the simulation.
- $Simu_Dura$ indicates the simulation's overall duration.

3) FOG NODES MEMORY STORAGE

When determining a fog node's accessible memory storage, Algorithm 4 converts the storage capacity from kilobytes to megabytes as needed.

$$MSC = MSC - PacketSize \quad (4)$$

where:

- Memory Storage Capacity (MSC) represents the total available memory storage capacity before storing the packet.
- packetSize represents the size of the data packet or message that needs to be stored in memory.

B. WEIGHTED SCORE CALCULATION

The weighted score of a Fog Node is determined by evaluating its energy consumption, CPU utilization, and memory storage. Each metric is assigned predefined weight values, reflecting their respective significance in assessing the node's performance shown in algorithm 5.

$$TWS = \sum_{i=1}^{3} WeightedScore_i \quad (5)$$

where:

- $WeightedScore_i$ represents each npde weighted score (energy, CPU, memory)
- i represents each component (energy, CPU, memory)

C. SORTING TECHNIQUES FOR THE ENERGY MODELLING DATASET

Sorting is a process that can arrange a set of data in a certain order, either ascending or descending. It makes organizing large datasets easier and takes less time and space. In general, only a few criteria, like running time, memory usage, stability, and parallel processing, are taken into account when choosing the best sorting algorithm.

In this study, the quick sorting algorithm is taken into consideration based on those findings. In Algorithm 6 it shows the sorting algorithm adapted to sort the fog nodes according to their overall weighted score. QuickSort uses a selected "pivot" value to split a large array into smaller arrays. One array contains numbers smaller than the pivot, and another contains values larger than the pivot. Until there is just one element in each sub-array, this partitioning operation is repeated. After sorting, the array is suitably rearranged, and the sorted Fog Nodes are shown in accordance with the algorithm's specifications.

D. FINDING SHORTEST NEIGHBOR

In a fog network, nodes act as equals, sharing resources through a decentralized architecture. The shortest path

Algorithm 1 Counting of Packet Sending, Receiving, Idle, and Sleep of Each Fog Node

Require $Ti, R, STS, STI, RN, SC, IC, STT, SPC, SLPC, IPC, STR, RPC, TRT, TTT, TST, TIT$
Ensure Calculation of total Packet Sending, Receiving, Idle, and Sleep **function** `Simulation()`:
  $Ti \leftarrow$ `new Timer();` $R \leftarrow$ `new Random();` $STS \leftarrow 0;$ $STI \leftarrow 0;$. **while** *true* **do**
    $RN \leftarrow R.nextInt(100)$;
    if $RN < 20$ **then**
      if *not isSleeping* **then**
        $STS \leftarrow$ System.currentTimeMillis() ;
        isSleeping $\leftarrow$ *true* ;
        $SC++;$ $SLPC++;$
      end
    end
    else if $RN < 50$ **then**
      if *not isIdle* **then**
        $STI \leftarrow$ System.currentTimeMillis() ;
        isIdle $\leftarrow$ *true* ;
        $IC++;$ $IPC++;$
      end
    end
    else
      $STT \leftarrow$ System.currentTimeMillis() ;
      $SPC++;$
    end
    for *neighborinstanceof neighbors* **do**
      $STR \leftarrow$ System.currentTimeMillis() ;
      $RPC++;$ $TRT+=(System.currentTimeMillis() - STR);$
    end
    $TTT+=(System.currentTimeMillis() - STT)$;
    if $isSleepingAND(System.currentTimeMillis()-STS > 5)$ **then**
      $TST+=(System.currentTimeMillis() - STS)$;
    end
    if $isIdleAND(System.currentTimeMillis()-STI > 5)$ **then**
      $TIT+=(System.currentTimeMillis() - STI)$;
    end
  end
end function

Algorithm 2 FogNodes Total Energy Consumption Calculation

Require: $TEC, REC, IEC, SEC, ToEC, TPR, RPR, IPR, SPR, TRT, TTT, TST, TIT$
Ensure: Total Energy Consumption of the Fog Node
  function TotalEnergyConsumption
    // Initialize the value of Power Rate of Transmit or Sending, Receive, Idle and Sleep
    $TPR \leftarrow 5.0$
    $RPR \leftarrow 3.0$
    $IPR \leftarrow 1.0$
    $SPR \leftarrow 0.1$
    // Calculate the Energy Consumption of different States in Fog Node.
    $TEC \leftarrow TTT \times TPR$
    $REC \leftarrow TRT \times RPR$
    $IEC \leftarrow TIT \times IPR$
    $SEC \leftarrow TST \times SPR$
    $ToEC \leftarrow TEC + REC + IEC + SEC$
  end function

Algorithm 3 Calculation for CPU Utilization of Fog Nodes

Require: *OB*, *SOB*, *Load*
Ensure: CPU Utilization of the Fog Node
function CPUUtilization
OB ← ManagementFactory.get OperatingSystemMXBean()
// CPU load monitoring
if *OB instance of OperatingSystemMXBean* **then**
SOB ← *OB* as *OperatingSystemMXBean*
// Convert CPU Load to Percentage
Load ← *SOB.getSystemCpuLoad*() × 100
// Upload the CPU Load
setCpuUtilization(Load)
end
else
Print "CPU utilization monitoring not supported"
end
end function

Algorithm 4 Calculation for Memory Storage of Fog Nodes

Require: MSC, PS, MSMB, Unit.
Ensure: Available Memory Storage of the Fog Node
function MemorySize
MSC ← *MSC* − *PS*
MSMB ← *MSC*/1024.0
Unit ← "KB"
if *MSMB* ≥ 1.0 **then**
Unit ← "MB"
end
Print *MSMB* + *Unit*
end function

Algorithm 5 Calculation of Weight Score in Fog Node

1 **Require:** EW, CPUW, MW, ME, MAE, MCPU, MACPU, MMS, MAMS, NE, NCPUU, NMS, WES, WCPUS, WMS, TWS
2 **Ensure:** Weight Score of Fog Node
function TotalWeightedScore
// Initialize the value
EW ← 0.4
CPUW ← 0.3
MW ← 0.3
// Calculate the Normalized values
NE ← (*TEC* − *ME*)/(*MAE* − *ME*)
NCPUU ← (*CPUUtilization* − *MCPU*)/(*MACPU* − *MCPU*)
NMS ← (*MSC* − *MMS*)/(*MAMS* − *MMS*)
// Calculate the Weighted Score
WES ← *EW* × *NE*
WCPUS ← *CPUW* × *NCPUU*
WMS ← *MW* × *NMS*
// Calculate the Total Weighted Score
TWS ← *WES* + *WCPUS* + *WMS*
end function

Algorithm 6 Sorting and Printing the Sorted Fog Nodes, With QuickSort

1 **Require:** Ti, low, high, Pi
2 **Ensure:** Sorting in Weight Score
function Sortprint
// Sorting Start
Ti.start()
QuickSort(0, allFogNodes.size() - 1)
// Print sorted fog nodes
for *each FogNode fogNode in allFogNodes* **do**
Print fogNode.TotalWeightedScore()
end
end function

function QuickSort((low, high))
if *low* < *high* **then**
Pi ← partition(low, high)
QuickSort(low, Pi - 1)
QuickSort(Pi + 1, high)
end
end function

between nodes is determined autonomously by each node. Ant colony optimization (ACO) algorithms play an important role in discovering these paths by emulating ant behavior as they look for food. Initially, ants wander around aimlessly looking for food sources. This haphazard exploration exposes several paths between the nest and food. As ants find food, they bring pieces back to the nest, leaving pheromone trails along the quickest way. Pheromones, or chemical signals generated by ants, direct other ants in the colony to follow established courses.

In Opti-FOG ACO unfolds through distinct stages within a fog network environment. Initially, ants reside within their nest amidst an absence of pheromones. As they venture out, each path is explored with a 0.5 probability, gradually intensifying the pheromone concentration as more ants opt for shorter paths. Consequently, the pheromone concentration escalates upon the ants' return from the shortest path. In the context of a Fog network, the ACO algorithm proves instrumental in discerning the shortest path between a source and destination node. This algorithm evaluates all potential paths from a given node to its nearest neighbor, assigning each path a pheromone value. Upon surpassing the maximum pheromone threshold, the algorithm dynamically updates the max_pheromone value, iteratively progressing through feasible paths until completion.

E. OPTIMIZED NODE SELECTION

Fog computing reduces latency and bandwidth by bringing cloud-based services to the edge of the network, where end users and their devices are located. While the number of devices and users that rely on fog computing services continues to grow, there will be a need for more energy and storage space. Using the methodology of new neighbor node selection, the problem can be resolved.

Optimization algorithms play a pivotal role has accurate value calculated. As researchers strive to enhance the efficiency and effectiveness of these algorithms, a comprehensive understanding of the parameters involved becomes essential. This literature survey, examination into the diverse landscape of optimization algorithms, examining the parameters that govern their behavior and influence their performance.

Table 3 lists the survey that aims to provide insights into the intricate interplay between algorithmic parameters and their impact on optimization outcomes.

TABLE 3. Literature survey on optimization algorithms.

Author	Algorithms	Parameter
Hussein, M. K et al.[11]	Ant Colony Optimization and Particle Swarm Optimization	Quality of Service
Salimian, M et al.[28]	Particle Swarm Optimization	Quality of Service
Alotaibi, B. K et al.[29]	Ant Colony Optimization approach to implement load balancing technique	Quality of Service
Kishor, A. et al.[30]	Smart Ant Colony Optimization, Comparison with Particle Swarm Optimization algorithm and Bee Life algorithm	Quality of Service
Yin, Z et al.[31]	Hybrid Monarch Butterfly Optimization and Ant Colony Optimization algorithm (HMA)	Quality of Service

This Algorithm 7 seeks to determine the most ideal neighboring fog node by taking into account both distance and weighted score. During the selection procedure, nodes that surpass a predetermined weighted score threshold are filtered out.

The procedure FindShortestNeighbor (fogNode) aims to determine the best neighbor selection based on certain criteria. It retrieves a list of neighboring FogNodes from the fogNode parameter and proceeds to evaluate each neighbor. If the list is empty, it returns null. Otherwise, it iterates through the neighbors, calculating the distance between the fogNode and each neighbor while also considering the weighted score of the neighbor. If the weighted score is greater than 0.5 and the calculated distance is less than the shortest distance encountered so far, it updates the shortest distance and selects the neighbor with the highest score. Finally, it prints the best-selected neighbor (BS) which is the optimized node for data storage.

Algorithm 7 Selection of the Best Neighbor Node

```
Require: Ti, low, high, Pi, BS, SD, MIN_VALUE,
  MAX_VALUE, DIS, SCO
Ensure: Best Neighbor Selection
  function FindShortestNeighbor(fognode)
      neighbors ← fogNode.getNeighbors();
  if neighbors is empty then
      return null;
      end
      BS ← null
      SD ← Double.MAX_VALUE
  for each FogNodes neighbor in neighbors do
      DIS ←
  fogNode.calculateDistance(neighbor.getX(),
  neighbor.getY())
      SCO ← neighbor.calculateWeightedScore()
  if SCO > 0.5 then
      if DIS < SDandSCO > BS then
              SD ← DIS
              BS ← SCO
              BS ← neighbor
          end
       end
      end
      Print BS
  end function
```

IV. RESULTS AND DISCUSSIONS

This section presents a critical stage of the proposed work, in which a stable simulation environment is created to serve as the test bed for the research. Using careful application and testing, empirical findings are revealed that shed light on the effectiveness and complexities of the approaches. Then, a thorough performance assessment offers important information about the advantages, disadvantages, and possible areas for improvement of the suggested framework. These elements work together to help us traverse the complexity of our research landscape as we go through the iterative process of analysis and interpretation.

A. SIMULATION ENVIRONMENT

In Java simulation environment, a dynamic network of different node sizes was established, where nodes autonomously share data received from edge devices with neighboring nodes whenever scalability for data storage is required. Each node's activities, including packet sending, receiving, idle, and sleep states were meticulously monitored. These parameters were crucial for calculating the total energy consumption, CPU utilization, and memory storage of neighboring fog nodes. Subsequently, a weighted score is computed for each neighboring fog node based on these metrics, guiding the selection of the most optimal nodes. The best nodes calculated are sorted, and a quicksort algorithm was implemented, ensuring a systematic arrangement based on their weighted scores. Moreover, to streamline node selection

for enhanced performance, the Ant Colony Optimization algorithm is adopted. Through this intricate process, the system dynamically identified and prioritized the best-suited fog nodes for data transmission, contributing to the overall efficiency and reliability of the fog-computing environment. The dynamic nature present in fog computing systems is comprehensively shown in Figure 4.

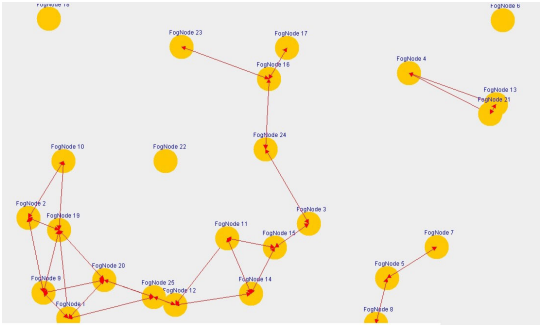


FIGURE 4. Dynamic fog-computing environment.

The data collected over the required metrics that are essential for the efficient optimization of the fog computing environment is shown in Table 4.

TABLE 4. Description of FogNode activities.

FogNode	Packet Receiving	Packet Sending	Sleeping	Idle
1	23	4	1	2
2	15	2	0	4
3	43	4	1	2
4	15	3	3	1
5	18	3	1	3
6	8	2	2	2
7	14	3	1	2
8	21	5	2	0
9	4	1	3	3
10	15	3	0	3
11	16	3	1	2
12	31	5	1	1
13	9	1	3	2
14	13	3	1	2
15	12	4	0	2
16	6	2	2	1
17	4	4	1	2
18	6	5	1	1
19	15	2	3	1
20	20	4	0	2
21	18	4	1	2
22	0	5	1	0
23	14	3	1	2
24	34	3	2	1
25	17	2	2	2

B. IMPLEMENTATIONS RESULTS

The implementation, results, and discussions focus on evaluating the effectiveness of the fog computing system in terms of its performance, efficiency, and reliability. This includes analyzing various metrics such as latency, throughput, resource utilization, and scalability. Results detail the observed performance of the fog computing system under different conditions and workloads. This include measurements of data processing times, response times, and network latency during data transfer. The improvements and challenges encountered during the implementation are highlighted.

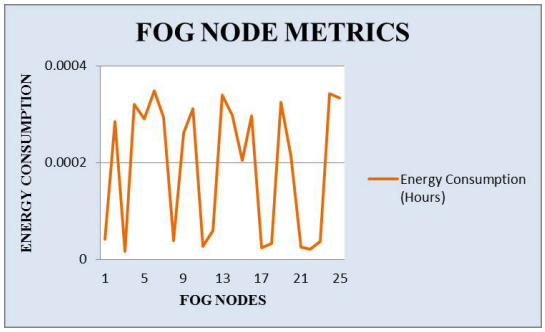


FIGURE 5. Fog node metrics - Energy Consumption.

Figure 5 illustrates the energy consumption of fog nodes, with the y-axis representing energy consumption and the x-axis depicting individual fog nodes. The graph provides a comparative view of energy usage across different nodes, offering insights into resource utilization and efficiency within the fog computing environment.

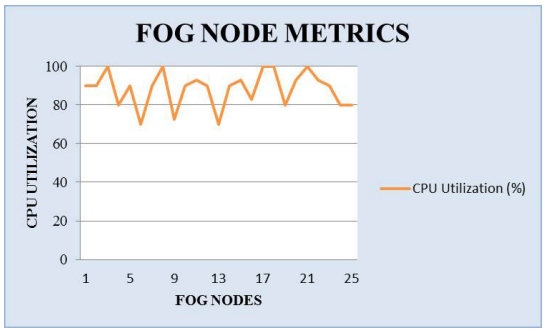


FIGURE 6. Fog node metrics - CPU Utilization.

Figure 6, CPU utilization is plotted on the y-axis against individual fog nodes on the x-axis. This graph showcases the distribution of CPU utilization among fog nodes, highlighting variations in computational workload and performance across the network.

Figure 7, displays memory capacity on the y-axis and fog nodes on the x-axis. It visualizes the memory storage capabilities of fog nodes, revealing disparities in memory capacity and potential constraints on data processing and storage within the fog computing infrastructure.

These graphs provide a comprehensive overview of key performance metrics for fog nodes that include energy consumption, CPU utilization, and memory capacity. Analyzing these metrics enables stakeholders to optimize resource allocation, identify potential bottlenecks, and enhance the

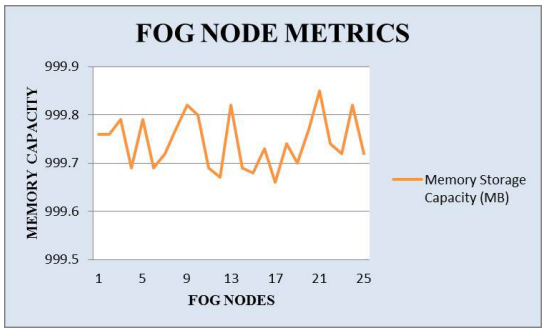


FIGURE 7. Fog node metrics - Memory Storage Capacity.

overall efficiency and reliability of the fog computing environment.

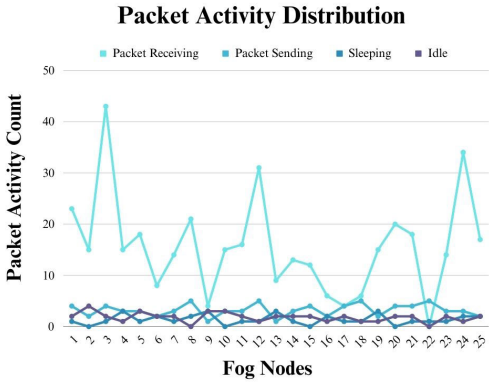


FIGURE 8. Packet activity distribution.

In Figure 8, displays the packet activity distribution across fog nodes, with the x-axis representing individual fog nodes and the y-axis indicating the count of packet activities. This graph provides a visual representation of the volume and distribution of packet activities among different nodes within the fog computing network. By analyzing this distribution, stakeholders can gain insights into network traffic patterns, identify nodes with high packet activity, and optimize data transmission and processing workflows to improve overall network efficiency and performance.

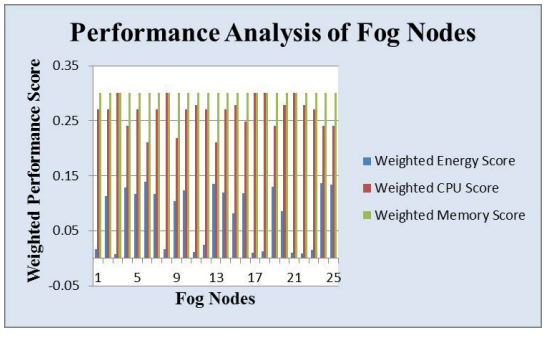


FIGURE 9. Performance analysis of fog node.

The graph in Figure 9 depicting fog node performance analysis illustrates the relationship between the total weighted score of fog nodes (on the y-axis) and the individual fog nodes (on the x-axis). The total weighted score serves as a comprehensive metric that encapsulates various performance parameters such as energy consumption, CPU utilization, and memory storage. When traversing along the y-axis, representing the total weighted score, it is observed how fog nodes are distributed based on their performance. Nodes with higher total weighted scores exhibit superior performance across the evaluated metrics, indicating optimal resource utilization and efficiency in data processing and transmission. Meanwhile, the x-axis represents the individual fog nodes participating in the analysis. Each node is positioned according to its corresponding total weighted score, allowing for easy comparison and identification of high-performing nodes. Through this graph, stakeholders can visually assess the performance distribution among fog nodes. Nodes positioned towards the right end of the graph demonstrate superior performance, while those towards the left may indicate areas for improvement or optimization.

C. PERFORMANCE EVALUATION

The graph in Figure 10 facilitates decision-making processes, enabling stakeholders to identify top-performing fog nodes for critical tasks or to pinpoint nodes requiring optimization efforts. It serves as a valuable tool for optimizing resource allocation, enhancing system reliability, and improving overall performance in fog computing environments.

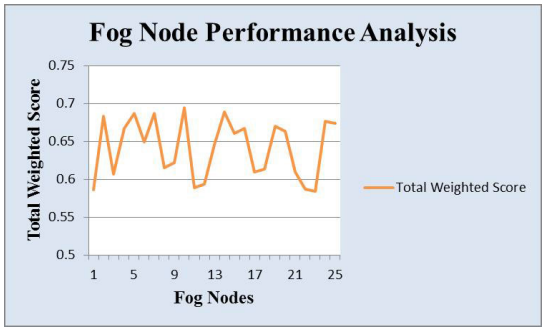


FIGURE 10. Fog node performance analysis.

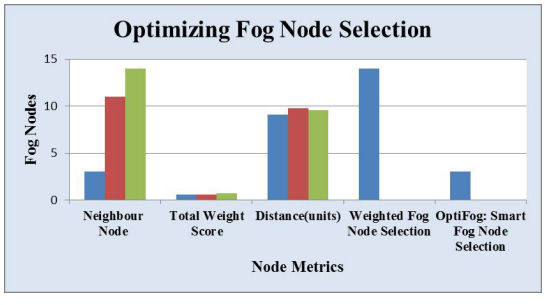


FIGURE 11. Comparison analysis.

In Figure 11 shows the comparison of the evaluation process done in the simulation environment shown in

Figure 4. In the dynamic fog computing environment, the selection of the best-optimized fog node for data storage by Fog Node 15 involved two distinct considerations. Initially, based on the weighted fog node value derived from the total weight score, Fog Node 14 emerged as the preferred choice due to its superior performance metrics and overall efficiency. Fog Node 14 demonstrated commendable energy consumption, CPU utilization, and memory capacity, positioning it as the top contender among Fog Node 15's neighboring nodes. However, a nuanced evaluation by the Opti-FOG algorithm introduced an additional criterion—shortest distance—into the decision-making process. Despite Fog Node 14's impressive performance metrics, the algorithm prioritized proximity and selected Fog Node 3 as the best-optimized node for Fog Node 15's data storage needs. Fog Node 3, while marginally further away than Fog Node 14, offered a more direct and efficient data transmission path, minimizing latency and optimizing network responsiveness.

This approach reveals the optimal node selection as a solution for data storage in the fog environment to ensure that scalability requirements are met while minimizing latency in data transmission. By integrating performance metrics and proximity considerations, fog computing systems can achieve optimal resource utilization and enhance overall system performance in dynamic and evolving environments. Ultimately, this dual evaluation process highlights the sophistication of fog computing algorithms in navigating trade-offs and selecting the most suitable nodes for diverse operational scenarios. By optimizing performance metrics and proximity, fog computing systems can effectively balance scalability and latency concerns, ensuring efficient and responsive data storage and transmission in fog environments.

To estimate the performance of the proposed work Opti-Fog, the comparison between Khan and Singh [30], Ogundoyin and Kamil [27], and Opti-Fog regarding their consideration of parameters for energy consumption is analyzed and shown in the graph in Figure 12. Khan et al. focus solely on transmit and receive operations as parameters for energy consumption. While these parameters are undoubtedly crucial components of energy consumption, they offer a limited view of the node's activity and resource utilization. Ogundoyin et al. extend this perspective by including idle time in addition to transmit and receive operations. While this expansion provides a more holistic view than Khan et al., it still overlooks the energy consumption during sleep mode, which can be significant, especially in scenarios where nodes spend a considerable amount of time in low-power states.

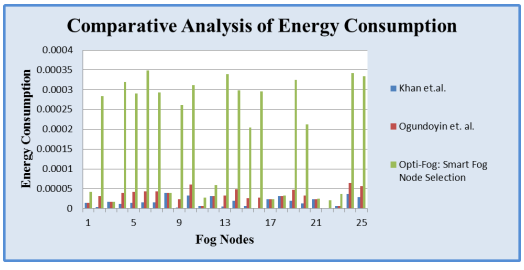


FIGURE 12. Comparative Analysis of Energy Consumption.

In contrast, Opti-Fog's consideration of transmit, receive, idle, and sleep parameters offers a more comprehensive assessment of energy consumption across different operational states of the nodes. By accounting for energy consumption during sleep mode, Opti-Fog captures the energy expenditure even when nodes are not actively transmitting or receiving data, providing a more accurate reflection of the node's overall energy utilization pattern. This comprehensive approach of Opti-Fog ensures that energy consumption is evaluated across all relevant operational states, enabling a more precise understanding of the node's energy behavior. Consequently, Opti-Fog is better equipped to make informed decisions regarding resource allocation and node selection, leading to more accurate and efficient optimization outcomes compared to its counterparts.

The graph in Figure 13 visually depicts the energy consumption distribution for each method, with bars representing the central tendency (mean or median) and spread (standard deviation or range) of the data. Observing the graph, it's evident that Opti-Fog exhibits significantly higher energy consumption compared to the other two methods, as indicated by its much larger mean and median values. Additionally, Opti-Fog demonstrates greater variability in energy consumption, as reflected by its larger standard deviation and range compared to the other methods. Conversely, Khan et al. show the lowest energy consumption values among the three methods, with Ogundoyin et al., falling in between the other two methods in terms of energy consumption as given in the Table 5. The graph provides a clear visual representation of the differences in energy consumption between the methods, highlighting their respective tendencies and variability.

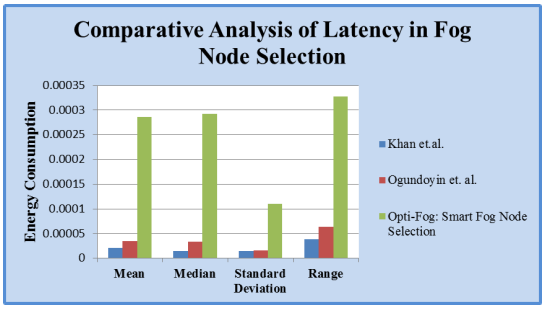


FIGURE 13. Energy consumption distribution.

The statistical analysis of the dataset comparing "Khan et al.", "Ogundoyin et al.", and "Opti-Fog: Smart Fog Node Selection" shows Figure 14 that "Opti-Fog" is the superior method for node selection in terms of energy consumption. The ANOVA results demonstrate significant variations in mean energy consumption figures between the three approaches, with an F-statistic of 36.60 and a p-value of 1.08e-11, indicating very strong statistical significance. This implies that the disparities in energy

TABLE 5. Comparative analysis of energy consumption distribution.

Statistics	Khan et. al.	Ogundoyin et. al.	Opti-Fog: Smart Fog Node Selection
Mean	0.0000207	0.0000351	0.0002864
Median	0.000015	0.000033	0.000293
Standard Deviation	0.0000147	0.0000162	0.0001107
Range	0.000038	0.000064	0.000328

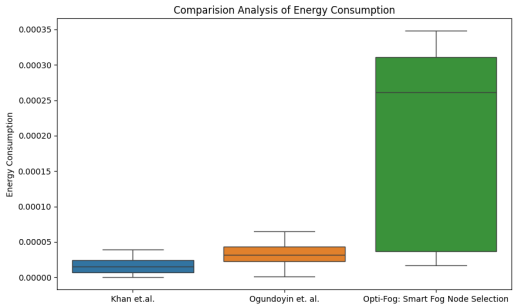


FIGURE 14. Comparative analysis of energy consumption.

use are not attributable to random variation. To determine which techniques differ significantly, a Tukey's HSD post hoc test was used. The post-hoc test findings revealed that ''Opti-Fog'' is significantly different and superior to the other two approaches. Specifically, ''Opti-Fog'' consumes more energy consistently than ''Khan et al.'' and ''Ogundoyin et al.''

In the Fog Computing environment, the Opti-Fog method stands out for its approach to resource allocation and node selection, which incorporates an intriguing perspective on energy consumption. Contrary to conventional insight, where lower energy consumption is often associated with efficiency and reliability, Opti-Fog challenges this notion by suggesting that nodes exhibiting higher energy consumption can be deemed more trustworthy. The rationale behind this assertion lies in the premise that nodes consuming more energy are likely handling heavier workloads or executing critical tasks, thus demonstrating a higher level of activity and engagement within the network. This heightened level of involvement suggests that these nodes are actively contributing to the network's operations and are potentially more reliable. Furthermore, the increased energy consumption could signify a willingness to invest resources in sustaining network performance and fulfilling obligations, further bolstering their perceived trustworthiness. Therefore, giving priority to nodes with higher energy consumption could be viewed as a deliberate move to improve network performance and reliability in the context of Opti-Fog, where nodes are chosen based on their patterns of energy use. Opti-Fog provides a fresh viewpoint on resource allocation and node selection by adopting this paradigm change in node trustworthiness evaluation, opening the door for more durable and resilient Fog Computing infrastructures.

D. CASE STUDY: OPTIMIZING FOG NODE SELECTION FOR HEALTHCARE MANAGEMENT SYSTEM USING THE OPTI-FOG FRAMEWORK

In a healthcare management system, real-time monitoring of patient vital indicators, such as blood pressure, is essential for timely medical action. However, the sheer volume of data collected by these sensors can overload fog nodes, causing delayed response and inadequate system performance. To address this issue, the Opti-FOG framework is presented, which aims to optimize fog node selection and resource allocation. By deliberately selecting fog nodes based on proximity, capacity, and availability, Opti-FOG provides effective data processing while improving system responsiveness. Opti-FOG reduces latency and maximizes resource usage by dynamically allocating computing resources. Finally, implementing Opti-FOG in healthcare management systems has the potential to improve patient care by allowing for more rapid data processing and improving overall system performance.

1) SCENARIO

Blood pressure monitoring is a standard technique in hospitals, particularly for patients admitted with various medical disorders or undergoing therapy. Patients may require frequent blood pressure checks based on their medical needs, which can contribute to the overall data collected by blood pressure sensors. This increased data production puts a tremendous pressure on the current fog nodes that process and analyze this data. As a result, the hospital has difficulty managing and digesting the influx of data, which is crucial for rapid medical treatments and patient care. Recognizing the significance of tackling these difficulties, the hospital recognizes the need to strengthen its fog computing infrastructure by incorporating another fog node to it. This proactive strategy is required to ensure the continuous flow of data processing, allowing healthcare professionals to quickly access critical patient information and make educated clinical decisions.

2) IMPLEMENTATION OF OPTI-FOG FRAMEWORK

- Scaling Horizons: Expanding the Realm of Fog Computing:

  In response to growing data traffic, the hospital launches a strategic push to strengthen its fog computing infrastructure. This entails adding more fog nodes to the current network to accommodate the increasing data load. However, this growth endeavor is not without difficulties. The selection of appropriate neighbor nodes presents significant challenges, involving careful consideration of aspects like as node capacity and availability. Despite these hurdles, the hospital understands the importance of developing its fog computing network in order to maintain seamless data processing and uninterrupted healthcare services.

- Revolutionizing Resource Allocation with Opti-FOG:
  The Opti-FOG framework emerges as a viable method to overcome the challenges of fog node selection and optimization. This new architecture is intended to handle the multifarious difficulties that arise in fog computing environments, notably in healthcare contexts. Opti-FOG uses the advanced Ant Colony Optimization (ACO) algorithm to carefully analyze a variety of crucial criteria, including proximity, node capacity, and availability. By methodically assessing these criteria, Opti-FOG determines the best fog node configuration for effectively managing growing data traffic, improving both data storage and processing capacities inside the healthcare infrastructure.
- Resource Allocation:
  The dynamic resource allocation method is a key component of the Opti-FOG framework, as it plays an important role in optimizing system performance. Opti-FOG intelligently allocates resources among fog nodes using thorough workload distribution analysis and node capacity assessment. This dynamic allocation technique ensures that fog nodes with available capacity are carefully discovered and chosen to handle extra data processing workloads. Opti-FOG supports effective resource consumption by carefully utilizing existing resources, ultimately improving overall system performance and responsiveness.
- Unveiling Opti-FOG's Real-World Journey in Healthcare:
  In a significant step toward modernization, the Opti-FOG framework takes center stage in the hospital's fog computing environment, encouraging collaboration between IT workers and healthcare practitioners alike. With thorough planning and execution, Opti-FOG integrates smoothly into the hospital's existing infrastructure, poised to transform data management procedures. Simulation findings demonstrate Opti-FOG's superior capacity to precisely assess fog node selection performance while dramatically improving system latency and availability. As Opti-FOG moves into real-world deployment scenarios, the transformative influence on healthcare management systems becomes clearer. By putting latency and availability first, Opti-FOG emerges as a shining example of innovation, ushering in a new era of efficiency and efficacy in healthcare data management.

3) OUTCOME OF OPTI-FOG FRAMEWORK WITH OPTIMIZED FOG NODE SELECTION IN HEALTHCARE MANAGEMENT SYSTEM

For the healthcare facility, the deployment of Opti-FOG produces a multitude of revolutionary outcomes that pave the way for previously unheard-of improvements in data management and system performance.

- Enhance System Performance:
  Efficient processing of patient data is ensured by Opti-FOG's advanced algorithms, which significantly enhance system performance by managing latency problems. Better clinical outcomes and patient care are ultimately the result of this improved performance, which also leads to more effective workflows and faster reaction times.
- Scalability:
  With Opti-FOG, the fog computing infrastructure of the hospital gains a new degree of scalability that allows it to adapt to variations in data traffic with ease. Opti-FOG makes sure the system is responsive and flexible even during times of high activity by dynamically altering resources in response to demand. This improved scalability encourages more resilience and flexibility, making it easier for the healthcare facility to adjust to shifting patient needs and operational specifications.
- Optimized Resource Utilization:
  Opti-FOG's strategic approach to resource allocation revolutionizes the utilization of computer resources in the healthcare environment. By leveraging variables like availability, capacity, and proximity to improve data processing and storage effectiveness, Opti-FOG maximizes performance. Because resources are being used optimally, the healthcare organization may allocate them more sensibly and sustainably, which also boosts operational effectiveness and reduces costs.
- Improved System Performance, Scalability, and Reliability with Opti-FOG:
  In conclusion, the implementation of the Opti-FOG framework within a healthcare management system exemplifies a transformative approach to fog computing. By dynamically optimizing fog node selection and resource allocation, Opti-FOG significantly enhances system performance, scalability, and reliability. Addressing the challenges of node failures and network disruptions through robust fault tolerance mechanisms, dynamic reconfiguration, and real-time monitoring ensures continuous, high-quality service delivery. This proactive strategy not only improves patient care through timely data processing but also reduces operational costs and enhances the overall user experience. The Opti-FOG framework stands as a testament to innovative solutions in fog computing, capable of adapting to the ever-evolving demands of modern healthcare environments.

In Figure 15 depicts the dynamics of patients' systolic and diastolic blood pressure readings over time. Systolic pressure changes may be indicative of variable degrees of stress, activity, or other blood pressure-related conditions. Meanwhile, the relatively stable diastolic pressure indicates that the baseline pressure in the arteries is consistent. This information is critical for healthcare practitioners to

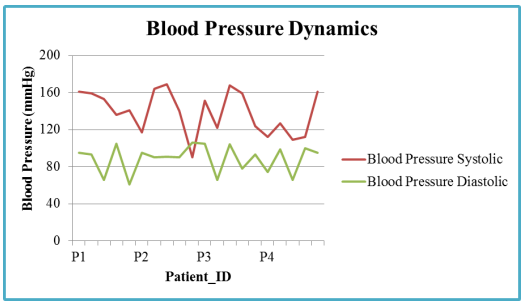


FIGURE 15. Blood pressure dynamics.

successfully monitor and manage their patients' cardiovascular health levels.

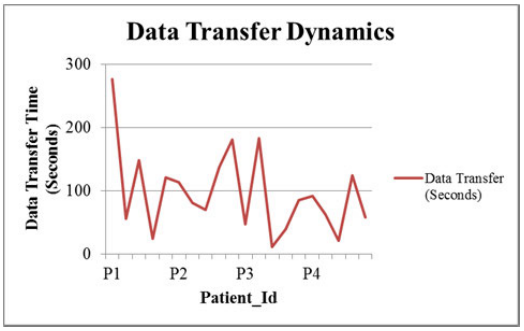


FIGURE 16. Data transfer dynamics.

In Figure 16 depicts that data transfer times differ significantly amongst patients, with some experiencing more consistent and efficient transfers and others seeing more variability and larger beginning delays. This information is critical for identifying potential bottlenecks in the data transfer process and improving it for higher performance. Analyzing these dynamics can assist healthcare providers in ensuring timely and trustworthy data transmission, which is required for successful patient monitoring and care.

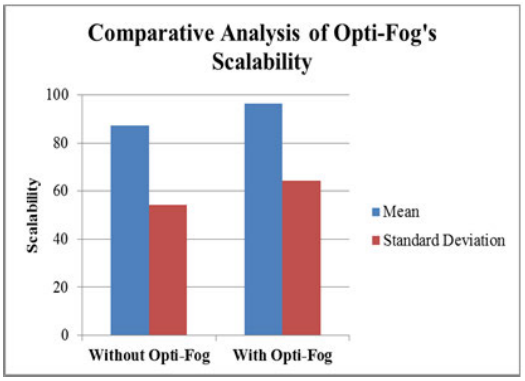


FIGURE 17. Comparative analysis of Opti-FOG's scalability.

In Figure 17 shows the ''Comparative Analysis of Opti-Fog's Scalability'' graph, which depicts the mean scalability and standard deviation of two scenarios: one without Opti-Fog and one with. Scalability is represented on the y-axis, which ranges from 0 to 100. The blue bars reflect mean scalability, and the red bars represent standard deviation. The mean scalability without Opti-Fog is around 87.42, with a standard deviation of around 54.16. This shows that the average scalability is relatively low, with significant variability. In contrast, the Opti-Fog scenario has a higher mean scalability of around 96.28 and a standard deviation of around 64.34. This suggests that Opti-Fog enhances the average scalability and also variability, leading to performance that is more consistent. Additionally, Opti-Fog's architecture improves data storage efficiency, particularly when the nodes are busy. This ensures that data storage remains robust and efficient, even under heavy load conditions, further underscoring its superiority. From this analysis, it can be concluded that the Opti-Fog system is more efficient. It improves overall scalability while also ensuring more stable and predictable performance, making data storage more efficient when nodes are busy, compared to scenarios without Opti-Fog.

E. ENHANCING SECURITY IN OPTI-FOG: OVERCOMING KEY CHALLENGES

By addressing the critical security challenges through the implementation of encryption, multi-factor authentication, secure communication protocols, access control policies, and intrusion detection systems, Opti-FOG ensures a robust and secure fog computing environment. This comprehensive approach not only protects data and resources but also enhances the overall reliability and trustworthiness of the fog computing infrastructure.

- Data Security:
  The Opti-FOG framework ensures robust data security by implementing advanced encryption techniques for data both at rest and in transit. By using strong encryption algorithms such as AES-256 for stored data and TLS/SSL for data transmission, Opti-FOG protects sensitive information from unauthorized access and breaches. Additionally, the framework includes data integrity checks using cryptographic hash functions (e.g., SHA-256) to ensure that data remains unaltered during storage and transmission. This ensures that even if an attacker gains access to the data, they cannot read or alter it without detection.
- Authentication and Authorization:
  Opti-FOG enhances authentication and authorization through multi-factor authentication (MFA) mechanisms and fine-grained access control policies. Users and devices must verify their identity using multiple authentication methods, such as passwords, biometrics, or cryptographic tokens, before gaining access to the system. Furthermore, Opti-FOG employs role-based access control (RBAC) and attribute-based access control (ABAC) to ensure that only authorized users and devices can access specific resources and perform

certain actions. These measures prevent unauthorized access and ensure that users only have the permissions necessary for their roles.

- Network Security:

  Opti-FOG secures network communications between fog nodes and edge devices by using secure communication protocols such as TLS/SSL. This prevents eavesdropping and man-in-the-middle attacks by ensuring that data is encrypted during transmission. The framework also incorporates network segmentation and virtual private networks (VPNs) to create isolated and secure communication channels. Firewalls and intrusion prevention systems (IPS) are deployed to monitor and filter incoming and outgoing traffic, blocking malicious activities and unauthorized access attempts. This comprehensive approach to network security ensures that communication remains secure and reliable.
- Access Control:

  Opti-FOG implements stringent access control mechanisms to regulate who can access what within the system. Role-Based Access Control (RBAC) assigns permissions based on user roles, ensuring that individuals can only access information and perform actions pertinent to their responsibilities. Attribute-Based Access Control (ABAC) provides even finer control by considering various attributes (such as time of day, location, and device type) in access decisions. These access control measures are enforced through policies that are regularly reviewed and updated to adapt to new security requirements and threats. This ensures that access to sensitive data and critical system functions is tightly regulated and monitored.
- Intrusion Detection and Prevention:

  The Opti-FOG framework includes advanced intrusion detection and prevention systems (IDPS) to monitor and analyze network traffic for signs of malicious activities or policy violations. These systems use machine learning and anomaly detection techniques to identify unusual patterns that may indicate a security threat. Once a potential threat is detected, Opti-FOG can automatically respond by isolating the affected nodes, blocking malicious traffic, and alerting administrators. Real-time monitoring and automated response capabilities ensure that threats are promptly addressed, minimizing the impact of any security incidents. Additionally, regular security audits and penetration testing help to identify and mitigate vulnerabilities before they can be exploited.

V. FUTURE RESEARCH OPPORTUNITIES IN ADVANCING OPTI-FOG

The incorporation of modern facilities AI/ML-based approaches into resource allocation processes with node selection appears to be a crucial approach for addressing the changing needs of fog computing environments. A proactive response to the problems caused by dynamic workloads, erratic demands, and variable network conditions is provided by this combination. Fog nodes are able to go beyond traditional models and improve node selection tactics while dynamically optimizing resource allocation through the application of artificial intelligence and machine learning. By utilizing AI/ML algorithms, one may make intelligent decisions and adjust in real-time, assuring optimal resource use and preventing bottlenecks. Reinforcement learning, deep learning, genetic algorithms, federated learning, and swarm intelligence are among the leading techniques in this game-changing field. DL models enable predictive analytics to precisely estimate future resource requirements, while RL algorithms provide fog nodes the capacity to learn and modify resource distribution policies on their own. GAs provide an evolutionary method for allocating resources optimally, whereas Swarm Intelligence encourages cooperative decision-making among dispersed nodes. Federated Learning also guarantees privacy-preserving optimization via dispersed fog networks. By incorporating these AI/ML-based methods into resource allocation and node selection, fog computing systems become more efficient and scalable. Additionally, this provides the foundation for intelligent, self-adaptive networks that can handle the wide range of demands of contemporary applications.

A. ENHANCING RESILIENCE IN OPTI-FOG FOR ROBUST SYSTEM ADAPTABILITY WITH DYNAMIC FOG NODE RECONFIGURATION

The Opti-FOG framework can incorporate fault tolerance techniques, dynamic node reconfiguration, load balancing algorithms, adaptive routing strategies, and real-time monitoring and response systems to strengthen its resilience against node failures or network outages. By reducing downtime and guaranteeing continuous service delivery, these steps improve the resilience and dependability of the system. Through adeptly navigating these challenges, Opti-FOG enhances user experience through continuous performance and lowers expenses related to manual intervention and emergency repairs. This resilience is essential, particularly in mission-critical sectors where even small disruptions can have a big impact, like the industrial Internet of things and the healthcare industry. Moreover, a reinforced Opti-FOG architecture gains flexibility, scalability, and competitiveness, are making it more appropriate for the changing requirements of modern fog computing environments. Its ability to manage node failures or network disruptions strengthens its value proposition, which is important for businesses looking for solutions that can adjust to changing circumstances. Research showing how well the system tackles these issues contributes significantly to its practical relevance and usefulness in real-world deployment scenarios.

The Opti-FOG system is far more reliable and adaptive in a variety of ways when fog node assignments are dynamically reconfigured in response to node failures or shifting network conditions. Through proactive fault

tolerance methods, the framework minimizes downtime and ensures ongoing service delivery by continuously monitoring the network environment and quickly detecting node problems. Dynamic reallocation prevents individual nodes from being overloaded and maximizes system performance, especially during moments of high workload. It does this by optimizing resource consumption and task distribution simultaneously. Moreover, adaptive routing features reduce latency and improve overall system stability by ensuring dependable communication between fog nodes and edge devices. Scalability is further enhanced by this adaptability; fog node assignments are dynamically modified by the framework to meet shifting workload and network traffic demands, guaranteeing effective resource allocation and scalability. Effective redundancy management also improves system resilience and reliability by activating backup nodes or resources in the event of a failure. The Opti-FOG framework integrates these principles to provide improved adaptability and dependability in dynamic fog computing environments. This guarantees continuous service delivery and excellent performance under different circumstances.

B. SWARM INTELLIGENCE FOR COLLABORATIVE RESOURCE ALLOCATION

In fog computing, swarm intelligence algorithms like Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) provide a decentralized method for cooperative resource allocation. These algorithms use cooperative and self-organization concepts found in natural systems to maximize resource allocation choices across the board. To coordinate their activities, fog nodes share information and interact locally. They then dynamically modify resource allocations in response to environmental cues and local feedback. Swarm intelligence techniques allow fog computing systems to effectively adjust to changing network circumstances and demands by leveraging the collective intelligence of distant nodes, hence enhancing scalability and resilience.

C. OPTI-FOG IN SPECIFIC WITH AI/ML-BASED RESOURCE ALLOCATION TECHNIQUES

1) REINFORCEMENT LEARNING FOR DYNAMIC RESOURCE ALLOCATION

Regarding the dynamic resource allocation problems in fog computing, reinforcement learning presents a promising strategy. Fog nodes can interact with their surroundings to learn the best resource allocation rules by using methods like Proximal Policy Optimization (PPO) and Deep Q-Networks (DQN). Reactive learning (RL) allows fog nodes to dynamically modify their resource allocation plans in response to network feedback, guaranteeing optimal resource use even in settings with varying demand. RL-based techniques enable fog computing systems to effectively allocate resources and scale on-the-fly while preserving responsiveness and performance through ongoing learning and adaptation.

2) DEEP LEARNING FOR PREDICTIVE RESOURCE MANAGEMENT

In fog computing environments, deep learning techniques like long short-term memory networks (LSTMs) and recurrent neural networks (RNNs) provide a potent way to manage resources predictively. Deep learning models may accurately predict future resource requirements by evaluating past data on network traffic, resource utilization, and other pertinent aspects. With the help of these forecasts, fog nodes may proactively distribute resources, reducing traffic and guaranteeing that services are delivered to edge devices on time. The efficiency and dependability of fog computing systems are improved by deep learning-based predictive resource management, which learns continuously and adapts to changing network dynamics.

3) GENETIC ALGORITHMS FOR OPTIMIZATION

An efficient optimization method for resolving resource allocation issues in fog computing settings is offered by genetic algorithms. GAs iteratively produce and evolve candidate solutions to achieve nearly optimal resource allocation techniques, drawing inspiration from the concepts of natural selection and genetics. GAs can be used by fog nodes to investigate the solution space while taking into account variables like traffic patterns, edge device proximity, and node capacity. GAs evolve via successive generations until they reach a solution that balances workload across fog nodes and maximizes resource consumption. GAs provide fog computing systems with the ability to optimize performance and allocate resources in a dynamic and diverse environment by utilizing the power of evolutionary optimization.

4) FEDERATED LEARNING FOR PRIVACY-PRESERVING RESOURCE ALLOCATION

Federated learning offers a way to allocate collaborative resources in fog computing settings while protecting user privacy. Federated learning guarantees data confidentiality and privacy by allowing fog nodes to collaboratively train machine learning models using decentralized data sources. To ensure that sensitive data is kept private, fog nodes use local data samples for model training. Only model updates are communicated with a central coordinator. By means of federated learning frameworks, fog nodes can jointly optimize strategies for resource allocation while respecting organizational and regulatory privacy regulations. Federated learning improves fog computing systems' resource allocation reliability and efficiency by fusing the advantages of collaborative optimization with data privacy protection.

5) HYBRID MODELS INTEGRATING AI/ML TECHNIQUES

In fog computing, hybrid models that combine various AI/ML techniques provide a thorough method for optimizing resource allocation. Hybrid models combine deep learning for predictive analytics, reinforcement learning for dynamic decision-making, and optimization methods like genetic

algorithms or swarm intelligence to take advantage of each approach's advantages and tackle a variety of problems. Based on real-time feedback, these models adaptively modify the relative importance of various strategies, guaranteeing scalability and resilience in fog computing scenarios. Hybrid models enable fog computing systems to improve performance and responsiveness by combining complementary techniques to optimize resource allocation dynamically and mitigate the effects of capacity restrictions and fluctuating demand.

VI. CONCLUSION

The proposed Opti-FOG algorithm emphasizing weighted fog node values and proximity considerations were exploration of fog node selection strategies, underscores the complexity and significance of optimizing data storage in fog computing environments. The dual evaluation process highlights the necessity of balancing performance metrics with proximity for optimal node selection. To accomplish the task of selecting the optimal neighbor node in the fog networking, three key stages must be addressed: calculating the total energy consumption of each neighboring network node using different parameters, arranging the energy consumption of each neighbor node's data sets in chronological order using sorting techniques, and implementing an ant colony optimization technique to determine the path of least resistance based on chronological data sets. Algorithm refinement offers potential enhancements to Opti-FOG by incorporating factors like network traffic patterns and dynamic workload distribution. Validation studies done in the simulation are imperative to assess fog node selection algorithms' performance in practical deployment scenarios accurately. Moreover, delving into the applicability of Opti-FOG in diverse scenarios and evaluating its performance under varying conditions would provide valuable insights. In conclusion, this study sets the stage for exciting advancements in optimizing data storage and transmission in fog computing environments, with the Opti-FOG algorithm poised as a promising avenue for future research and innovation.

REFERENCES

[1] R. Karthikeyan, B. Monisha, A. Nagul, C. Naveen, and A. Sakthinithesh, "Multi input renewable port selection with energy management system using Internet of Things," in *Proc. 7th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2023, pp. 427–432.

[2] Y. Xu and M. Khalilzadeh, "An approach for managing the Internet of Things' resources to optimize the energy consumption using a nature-inspired optimization algorithm and Markov model," *Sustain. Comput., Informat. Syst.*, vol. 36, Dec. 2022, Art. no. 100817.

[3] J. Kalezhi, D. Ntalasha, and T. Chisanga, "Using Internet of Things to regulate energy consumption in a home environment," in *Proc. IEEE PES/IAS PowerAfrica*, Jun. 2018, pp. 551–555.

[4] H. Farman, B. Jan, H. Javed, N. Ahmad, J. Iqbal, M. Arshad, and S. Ali, "Multi-criteria based zone head selection in Internet of Things based wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 87, pp. 364–371, Oct. 2018.

[5] M. Cuka, D. Elmazi, M. Ikeda, K. Matsuo, and L. Barolli, "IoT node selection in opportunistic networks: Implementation of fuzzy-based simulation systems and testbed," *Internet Things*, vol. 8, Dec. 2019, Art. no. 100105.

[6] R. Priyadarshini and N. Malarvizhi, "Secured data transfer between fog nodes using blockchain," in *Proc. 2nd Int. Conf. Comput. Bio Eng. (CBE)*. Singapore: Springer, 2021, pp. 417–422.

[7] M. Alam, N. Ahmed, R. Matam, and F. A. Barbhuiya, "L3Fog: Fog node selection and task offloading framework for mobile IoT," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2022, pp. 1–6.

[8] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.

[9] A. Malhotra, S. K. Dhurandher, M. Gupta, and B. Kumar, "Best fit power weighted difference method for fog node selection in smart cities," *IET Commun.*, vol. 14, no. 19, pp. 3448–3457, Dec. 2020.

[10] K. Tewani, "Ant colony optimization algorithm: Advantages, applications and challenges," *Comput. Model. New Technol.*, vol. 21, no. 2, pp. 69–70, 2017.

[11] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020.

[12] B. K. Alotaibi and U. Baroudi, "Offload and schedule tasks in health environment using ant colony optimization at fog master," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2022, pp. 469–474.

[13] G. Baranwal and D. P. Vidyarthi, "FONS: A fog orchestrator node selection model to improve application placement in fog computing," *J. Supercomput.*, vol. 77, no. 9, pp. 10562–10589, Sep. 2021.

[14] A. S. Alavizadeh, S. H. Erfani, M. Mirabi, and A. Sahafi, "An efficient distributed and secure algorithm for transaction confirmation in IOTA using cloud computing," *J. Supercomput.*, vol. 80, no. 2, pp. 1491–1521, Jan. 2024.

[15] P. Yadav and S. Kar, "Efficient content distribution in fog-based CDN: A joint optimization algorithm for fog-node placement and content delivery," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16578–16590, 2024.

[16] A. Naouri, N. A. Nouri, A. Khelloufi, A. B. Sada, H. Ning, and S. Dhelim, "Efficient fog node placement using nature-inspired metaheuristic for IoT applications," *Cluster Comput.*, pp. 1–17, Apr. 2024, doi: 10.1007/s10586-024-04409-3.

[17] J. L. Sarkar, S. K. Cowlessur, V. Ramasamy, B. Pati, T. M. Selvi, C. R. Panigrahi, B. Majumder, R. K. Verma, and N. M. F. Qureshi, "FogCom: SDN-enabled fog node selection for early detection of communicable diseases," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101432.

[18] A. A. Bukhari and F. K. Hussain, "Fuzzy logic trust-based fog node selection," *SSRN J.*, Jan. 2023, doi: 10.2139/ssrn.4573967.

[19] T. Qayyum, Z. Trabelsi, A. W. Malik, and K. Hayawi, "Mobility-aware hierarchical fog computing framework for Industrial Internet of Things (IIoT)," *J. Cloud Comput.*, vol. 11, no. 1, p. 72, Oct. 2022.

[20] P. K. Deb, S. Misra, and A. Mukherjee, "Latency-aware horizontal computation offloading for parallel processing in fog-enabled IoT," *IEEE Syst. J.*, vol. 16, no. 2, pp. 2537–2544, Jun. 2022.

[21] R. Priyadarshini and N. Malarvizhi, "Bi-directional trust management system in fog computing using logistic regression," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 29, no. 2, p. 808, Feb. 2023.

[22] M. Hajvali, S. Adabi, A. Rezaee, and M. Hosseinzadeh, "Decentralized and scalable hybrid scheduling-clustering method for real-time applications in volatile and dynamic fog-cloud environments," *J. Cloud Comput.*, vol. 12, no. 1, p. 66, Apr. 2023.

[23] H. Chen, W.-Y. Chang, T.-L. Chiu, M.-C. Chiang, and C.-W. Tsai, "SEFSD: An effective deployment algorithm for fog computing systems," *J. Cloud Comput.*, vol. 12, no. 1, p. 105, Jul. 2023.

[24] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 122–134, Dec. 2020.

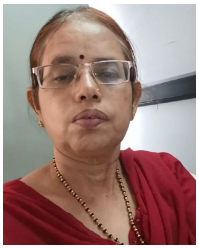
[25] A. Liutkevičius, N. Morkevičius, A. Venčkauskas, and J. Toldinas, "Distributed agent-based orchestrator model for fog computing," *Sensors*, vol. 22, no. 15, p. 5894, Aug. 2022.

[26] A. Bukhari, F. K. Hussain, and O. K. Hussain, "Fog node discovery and selection: A systematic literature review," *Future Gener. Comput. Syst.*, vol. 135, pp. 114–128, Oct. 2022.

[27] S. O. Ogundoyin and I. A. Kamil, "Optimal fog node selection based on hybrid particle swarm optimization and firefly algorithm in dynamic fog computing services," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105998.

[28] M. Rashid, L. Ardito, and M. Torchiano, "Energy consumption analysis of algorithms implementations," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Oct. 2015, pp. 1–4.
[29] T. O. John, H. C. Ukwuoma, S. Danjuma, and M. Ibrahim, "Energy consumption in wireless sensor network," *Energy*, vol. 7, no. 8, pp. 63–67, 2016.
[30] I. Khan and D. Singh, "Energy-balance node-selection algorithm for heterogeneous wireless sensor networks," *ETRI J.*, vol. 40, no. 5, pp. 604–612, Oct. 2018.
[31] R. Ullah, M. Yahya, L. Mostarda, A. Alshammari, A. I. Alutaibi, N. Sarwar, F. Ullah, and S. Ullah, "Intelligent decision making for energy efficient fog nodes selection and smart switching in the IoT: A machine learning approach," *PeerJ Comput. Sci.*, vol. 10, p. e1833, Mar. 2024.
[32] H. Li, X. Zhang, H. Li, X. Duan, and C. Xu, "SLA-based task offloading for energy consumption constrained workflows in fog computing," *Future Gener. Comput. Syst.*, vol. 156, pp. 64–76, Jul. 2024.
[33] S. R. Hassan, A. U. Rehman, N. Alsharabi, S. Arain, A. Quddus, and H. Hamam, "Design of load-aware resource allocation for heterogeneous fog computing systems," *PeerJ Comput. Sci.*, vol. 10, p. e1986, Apr. 2024.
[34] J. Du, B. Jiang, C. Jiang, Y. Shi, and Z. Han, "Gradient and channel aware dynamic scheduling for over-the-air computation in federated edge learning systems," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1035–1050, Apr. 2023.
[35] R. Priyadarshini and N. Malarvizhi, "Preputation based trust management system for malicious fog node detection," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 24, pp. 7311–7319, 2022.
[36] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Comput. Informat. J.*, vol. 3, no. 2, pp. 231–246, Dec. 2018.
[37] D. Li, C. Yu, Y. Tan, and J. Liu, "Optimization method of fog computing high offloading service based on frame of reference," *Mathematics*, vol. 12, no. 5, p. 621, Feb. 2024.
[38] H. Fahmi, M. Zarlis, E. B. Nababan, and P. Sihombing, "Ant colony optimization (ACO) algorithm for determining the nearest route search in distribution of light food production," *J. Phys., Conf. Ser.*, vol. 1566, no. 1, Jun. 2020, Art. no. 012045.
[39] H. Cheng, Z. Su, D. Zhang, J. Lloret, and Z. Yu, "Energy-efficient node selection algorithms with correlation optimization in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, Mar. 2014, Art. no. 576573.
[40] M. Salimian, M. Ghobaei-Arani, and A. Shahidinejad, "An evolutionary multi-objective optimization technique to deploy the IoT services in fog-enabled networks: An autonomous approach," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2008149.
[41] A. Kishor and C. Chakarbarty, "Task offloading in fog computing for using smart ant colony optimization," *Wireless Pers. Commun.*, vol. 127, no. 2, pp. 1683–1704, Nov. 2022.
[42] Z. Yin, F. Xu, Y. Li, C. Fan, F. Zhang, G. Han, and Y. Bi, "A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing," *Sensors*, vol. 22, no. 4, p. 1555, Feb. 2022.

P. KARTHIKEYAN received the Master of Engineering degree in computer science engineering from Anna University, Chennai, India, in 2012. He has more than 12 years of experience in teaching. Currently, he is a Research Scholar with Vellore Institute of Technology, Vellore, India. He has published papers in peer-reviewed international journals. His research interests include cloud computing, fog computing, and networking and data security. He has attended several National and International Conferences and workshops.



K. BRINDHA received the Ph.D. degree from VIT and the M.E. degree in computer science and engineering from Sathyabama University. She is currently an Associate Professor Senior with the School of Computer Science Engineering and Information Systems, VIT University, India. She has more than 25 years of experience in teaching as well expertise in cryptography, machine learning, deep learning, network security, cloud computing, and block chain domain. She published around 40 international journal articles, many international conference papers and ten books. She received the Active Researcher Award from VIT for the past 13 years. She is one of the reviewers in *Journal of Visual Languages and Computing* (JVLC) Elsevier journal and *PeerJ* journal. She is a Life Member of Computer Society of India.